UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
DEPARTAMENTO DE COMPUTAÇÃO

SISTEMA BASEADO EM REDES NEURAIS PARA COMPOSIÇÃO MUSICAL ASSISTIDA POR COMPUTADOR

DÉBORA CRISTINA CORRÊA

SÃO CARLOS
MAIO/2008

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária da UFSCar

C824sb

Corrêa, Débora Cristina.
Sistema baseado em redes neurais para composição musical assistida por computador / Débora Cristina Corrêa. -- São Carlos : UFSCar, 2008.
163 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2008.

1. Inteligência artificial. 2. Redes neurais. 3. Composição (Música). I. Título.

CDD: 006.3 (20ª)

Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia

Programa de Pós-Graduação em Ciência da Computação

"Sistema Baseado em Redes Neurais para Composição Musical Assistida por Computador"

DÉBORA CRISTINA CORRÊA

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Membros da Banca:

Prof. Dr. José Hiroki Saito
(Orientador – DC/UFSCar)

Prof. Dr. Jander Moreira
(DC/UFSCar)

Prof. Dr. Hani Camille Yehia
(DEE/UFMG)

São Carlos
Maio/2008

AGRADECIMENTOS

A Deus por estar comigo em todos os momentos.

A toda minha família, em especial aos meus pais e irmãos, pelo amor e apoio incondicionais, incentivos para que eu pudesse alcançar mais este objetivo e, principalmente, pela companhia ao longo de todos esses anos de vida. Minha família é minha razão de viver.

Ao Prof. Dr. José Hiroki Saito, meu orientador, pela confiança e amizade, pelos ensinamentos, opiniões, paciência, correções, pelos conhecimentos compartilhados, pela dedicação, e por tornar possível a concretização dessa etapa da minha vida.

A todo pessoal do GAPIS, pela amizade, consideração e discussões sobre os mais variados temas e assuntos. Em especial, aos meus amigos Alexandre Levada, Michelle Horta e Denis Salvadeo, pelo apoio incondicional nesses anos.

Ao meu amigo César Sangaletti pelos ensinamentos.

Ao meu amigo e companheiro Cristiano Sangaletti pelas horas de estudo, pelo apoio, incentivo, compreensão e companhia.

A todos meus amigos, pela paciência, pelo apoio e motivação nesses anos.

A todos do DC e PPG-CC, pelo apoio e aprendizado através das aulas e dos professores durante o mestrado.

A CAPES pelo apoio financeiro na realização do projeto.

Em suma, a todas aquelas pessoas que direta ou indiretamente contribuíram de alguma maneira para que tudo isso se tornasse realidade.

“Music is a moral law. It gives soul to the universe, wings to the mind, flight to the imagination, and charm and gaiety to life and to everything.” (Plato)

RESUMO

Várias pesquisas têm sido realizadas tendo em vista um sistema computacional de composição musical que buscasse, da melhor maneira possível, capturar as habilidades e criatividade da mente humana. Mais recentemente, as redes neurais artificiais (RNAs), ou modelos conexionistas, também passaram a ser utilizadas como modelos auxiliares para a composição musical. No caso da computação musical, os modelos conexionistas são capazes de aprender padrões e características presentes nas melodias do conjunto de treinamentos e obter generalizações dessas características para a composição de novas melodias. Portanto, as redes neurais artificiais passaram a ser utilizadas como modelos para a aprendizagem e composição musicais. O objetivo central desta dissertação de mestrado é propor um sistema de composição musical assistida por computador baseado em redes neurais. Esse sistema pode ser dividido em quatro principais etapas: treinamento, composição, avaliação e otimização. O trabalho de dissertação também propõe complementar as fases de treinamento e composição com um tipo de inspiração, proveniente da Natureza, com a utilização de contornos de relevos geográficos como informação adicional para a rede. As redes neurais usadas para o sistema são: BPTT (*Back-Propagation Through Time*) e LSTM (*Long-Short Term Memory*). Ambas as redes são comparadas quanto ao resultado obtido, sendo que a rede LSTM apresenta melhor desempenho. É também proposto para a rede LSTM um procedimento de otimização dos pesos iniciais e do número de neurônios da camada escondida, o que contribui para o desempenho obtido.

ABSTRACT

Several research studies have been realized in order to achieve a musical composition computational system that could, as much as possible, catch the human mind, skills, and creativity. More recently, artificial neural networks (ANNs), also have been deployed as auxiliary models for musical compositions. For musical computation, connectionist systems, as well as other systems that involve machine learning, are able to learn patterns and features available in the melodies of the training set and to generalize them to compose new melodies. Therefore, the use of neural networks in music learning and composition has attracted researchers and many approaches have been developed. The aim of this study is the proposal of a neural network based system for computer-aided musical composition. This system can be divided into four main processes: training, composition, evaluation and optimization. It is also proposed to complement the training and composition processes with a kind of inspiration, from Nature, using landscapes contours as additional information to the network. The neural networks used in the system are: BPTT (Back-Propagation Through Time) and LSTM (Long-Short Term Memory) networks. The results obtained are compared from both networks and it is observed that the LSTM network performs better. It is also proposed an approach that consists of optimizing the weight initialization process of the LSTM network in addition to an estimative of the ideal configuration of the hidden layer, that contributes to the obtained results.

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

INTRODUÇÃO

Os seres humanos são capazes de criar e apreciar a organização dos sons no espaço e no tempo. Música é a arte de combinar sons no espaço e no tempo, formando sentido musical. Nesse contexto, o espaço está relacionado com a harmonia, ou seja, relação simultânea entre os sons. O tempo está associado com a melodia, ou seja, relação seqüencial entre os sons. A habilidade do ser humano em criar e apreciar a organização dos sons no espaço e no tempo favorece o entendimento de que uma composição musical é formada por estruturas abstratas [MIRANDA, 2002].

A capacidade humana de entender e processar música é formada por dois domínios aparentemente distintos um do outro: o domínio da subjetividade abstrata que abrange composição musical e imaginação artística; e o domínio da objetividade abstrata, que abrange operações lógicas e raciocínio matemático. Não há dúvidas de que o computador é uma excelente ferramenta para o último domínio descrito, porém, é necessário explorar o potencial do computador para o domínio com características subjetivas.

Como a composição musical lida com estruturas abstratas, torna-se necessário definir três níveis de abstração:

1) Nível microscópico: Nesse nível o compositor trabalha com características microscópicas do som, geralmente relacionados com atributos físicos, como por exemplo, freqüências, amplitudes, espectro. Nesse caso, é mais provável que um pedaço musical seja representado por listas de valores numéricos ao invés de notas para performance de instrumentos acústicos.

2) Nível de nota: Nesse nível, o compositor trabalha com a unidade elementar da música: a **nota musical,** que é um simples evento sonoro caracterizado pelos atributos físicos (altura, intensidade, duração e timbre). Esse trabalho está mais concentrado nesse nível de abstração.

3) Nível de bloco: Nesse nível o compositor trabalha com grandes unidades musicais, como padrões de ritmo, temas melódicos e amostras de seqüências sonoras.

Segundo Miranda [2002], essa definição de níveis é importante para compositores que trabalham com computadores, uma vez que ela determina como será feita a construção dos componentes que formam a estrutura musical de uma composição.

Muitos pesquisadores têm usado computadores para aperfeiçoar a aplicação de composição musical. Recentemente, as redes neurais artificiais passaram a ser utilizadas como modelos para a aprendizagem de processos musicais.

As redes neurais artificiais, os sistemas conexionistas, representam formas de computação não algorítmica, inspiradas no modelo biológico de processamento de informações, e eliminam tanto a necessidade da separação processador / memória quanto o conceito de conjunto de instruções simbólicas presente nos modelos convencionais. Elas prevêem uma rede de "neurônios" na qual diferentes padrões de excitação são observados como uma função de interconexões entre os neurônios. Processamento e memória são distribuídos uniformemente pela rede de forma a fornecer processamento paralelo e rápido [DOLSON, 1989]. Existem diversos modelos propostos de redes neurais artificiais, e normalmente, essas redes são simuladas em computadores convencionais.

Os sistemas conexionistas geralmente oferecem mecanismos de aprendizado em que a computação desejada pode ser obtida expondo a rede repetidamente a exemplos que determinam o comportamento esperado. Nesses mecanismos, as redes neurais adaptam suas interconexões até que os padrões de excitação desejados estejam perto (do que a rede é capaz de obter) do comportamento desejado. Assim, as redes neurais são capazes de simular comportamentos complexos (e talvez biológicos) dificilmente de serem gerados por conjuntos de instruções ou regras.

MOTIVAÇÃO

É conhecido que formas de computação convencionais são velozes no processamento aritmético, porém, apresentam baixo desempenho em aplicações de reconhecimento de padrões, como identificação de indivíduos numa imagem, comparados com a capacidade de reconhecimento de padrões por um ser humano. Por essa razão têm sido pesquisados e propostos diversos modelos de redes neurais artificiais inspiradas nas redes neurais biológicas. No caso da composição musical, a aplicação de redes neurais artificiais busca suprir dificuldades encontradas em abordagens tradicionais implementadas nos computadores convencionais, visto que os seres humanos conseguem realizar uma composição simples com relativa facilidade.

Portanto, os modelos conexionistas estão sendo cada vez mais utilizados dentro de domínios como psicologia e ciência cognitiva. Esses modelos são capazes de armazenar e generalizar informações, aspectos fundamentais do

aprendizado. Percepção e cognição musical requerem essas mesmas habilidades. Assim, modelos conexionistas são apropriados para capturar aspectos do comportamento musical humano [TODD e GROY, 1991]. A possibilidade de entender e simular esse comportamento favorece a composição de melodias adequadas por redes neurais artificiais.

Uma das principais qualidades desses modelos conexionistas é a capacidade de aprender padrões e características presentes nas melodias do conjunto de treinamentos e obter generalizações dessas características para a composição de novas melodias. Essa abordagem é promissora em relação a outras abordagens que, muitas vezes, exigem a especificação de regras explícitas e não incorporam aspectos cognitivos do comportamento musical humano [TODD e GROY, 1991].

O objetivo central dessa dissertação de mestrado é propor um sistema de composição musical assistido por computador baseado em redes neurais. Esse sistema pode ser dividido em quatro etapas principais (Figura 1).

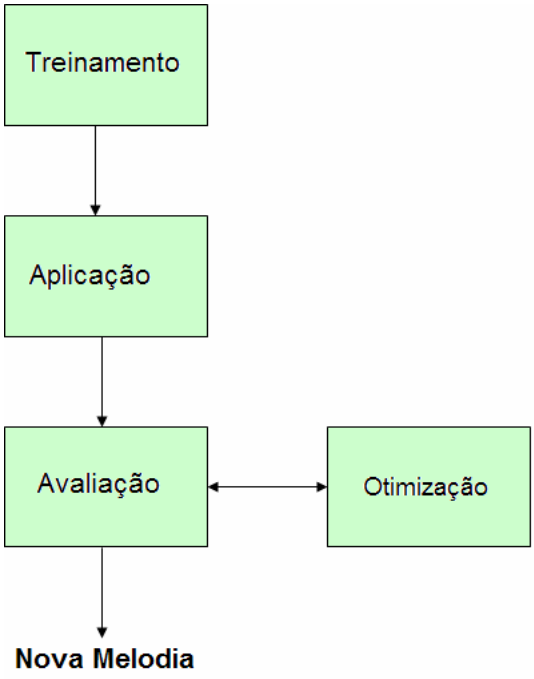


Figura 1: Sistema proposto para composição musical baseado em redes neurais

A primeira etapa consiste em treinar a rede neural adequadamente, com o conjunto de treinamento formado por melodias de um determinado estilo e com diferentes representações dos elementos musicais.

A próxima etapa é a aplicação da rede, que ocorre depois que o treinamento é concluído. Nessa etapa a rede neural é usada para a composição de novas melodias com base na etapa anterior de treinamento.

A inovação proposta nessas duas primeiras etapas é complementar as fases de treinamento e composição com um tipo de inspiração, proveniente da Natureza, com a utilização de contornos de relevos geográficos como informação adicional para a rede.

A próxima etapa consiste na avaliação das novas melodias geradas pela rede. Essa avaliação é baseada em três requisitos: notas repetidas, alternâncias abruptas de altura e notas fora da escala. Uma rede neural MLP (*Multi-Layer Perceptron*) foi utilizada para classificar as melodias em duas classes: apropriadas e inapropriadas.

Se a melodia é classificada como inapropriada, algumas correções ou otimizações são realizadas. Essas correções são baseadas em informações estatísticas coletadas do conjunto de treinamento. Essas duas últimas etapas (avaliação e correção) ainda não são muito exploradas na literatura e têm por objetivo verificar o desempenho da rede na tarefa de compor uma nova melodia.

ORGANIZAÇÃO DO TRABALHO

A descrição dessa dissertação de mestrado está estruturada nos seguintes capítulos, além do presente capítulo de introdução:

(1) **Definições sobre música**. Esse capítulo consiste na apresentação dos conceitos fundamentais do som e dos atributos das notas musicais.

(2) **Redes Neurais Artificiais**. Nesse capítulo são apresentadas as principais arquiteturas e os principais métodos de aprendizado das RNAs.

(3) **Abordagens sobre composição musical usando computadores.** Esse capítulo consiste na descrição de abordagens anteriores para composição musical por computadores, utilizando métodos convencionais e utilizando redes neurais artificiais.

(4) **Proposta de Trabalho**. Descreve a metodologia de trabalho utilizada, assim como a representação dos dados de treinamento.

(5) **Resultados Obtidos**. Apresenta os aspectos de implementação do sistema desenvolvido para composição musical baseado em redes neurais e os resultados obtidos.

(6) **Conclusões e propostas para trabalhos futuros**. Apresenta discussões sobre os resultados obtidos e apresenta propostas para trabalhos futuros.

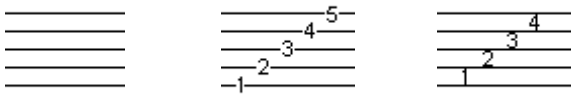
CAPÍTULO 1 – DEFINIÇÕES SOBRE MÚSICA

1.1 – CONSIDERAÇÕES INICIAIS

Para um melhor entendimento do trabalho, o capítulo apresenta as notas musicais e seus atributos, tais como a dinâmica, a altura, o timbre e a duração. O capítulo também descreve e ilustra os parâmetros da onda sonora e demonstra como esses parâmetros se relacionam entre si. Por fim, há uma breve discussão sobre o funcionamento do sistema auditivo do ser humano e de como a mente humana entende e interpreta os elementos musicais. O entendimento da música pela mente humana é uma das motivações para a criação de composições musicais por redes neurais artificiais que são inspiradas no funcionamento do cérebro humano. A organização desse capítulo é a seguinte: a sessão 1.2 apresenta os conceitos básicos sobre notações musicais; a sessão 1.3 descreve as propriedades do som e os atributos das notas musicais; a sessão 1.4 apresenta a Série de Fourier; a sessão, 1.5 discute o sistema auditivo; e a sessão 1.6 apresenta as considerações finais deste capítulo.

1.2 – NOTAÇÕES MUSICAIS

As notas musicais são escritas no pentagrama, ilustrado na Figura 1.1. O pentagrama é um conjunto de cinco linhas, paralelas e eqüidistantes que formam entre si quatro espaços.



(a) (b) (c)

Figura 1.1: (a) Pentagrama. (b) Linhas do Pentagrama. (c) Espaços do Pentagrama.

As claves são símbolos colocados no início do pentagrama e servem para dar nome às notas musicais. Existem três claves musicais: a clave de sol, de fá e de dó. As claves estão ilustradas na Figura 1.2.

(a) Clave de Sol (b) Clave de Fá (c) Clave de Dó

Figura 1.2: Claves musicais

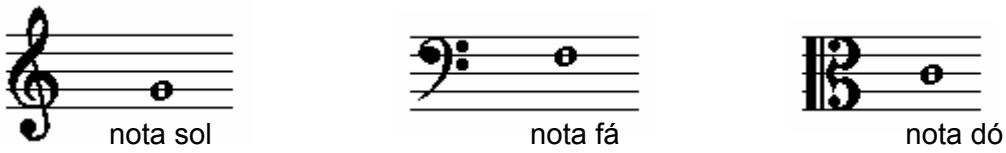


Figura 1.3: Relação das claves e suas respectivas notas

Como ilustrado na Figura 1.3, a clave de sol determina a localização da nota que receberá o nome *sol* no pentagrama; é utilizada para instrumentos musicais de sons médios e agudos. As demais notas são localizadas no pentagrama em função da diferença de freqüência em relação à nota *sol,* sendo as notas de freqüência mais baixa nas linhas ou espaço abaixo e as notas de freqüência mais alta nas linhas ou espaços acima. Da mesma forma, as claves de fá (para instrumentos de som graves) e dó (para instrumentos de som médios) determinam a localização das notas que receberão os nomes *fá* e *dó*, respectivamente, no pentagrama.

São sete as notas musicais naturais e estão ilustradas na Figura 1.4 para a clave de sol:

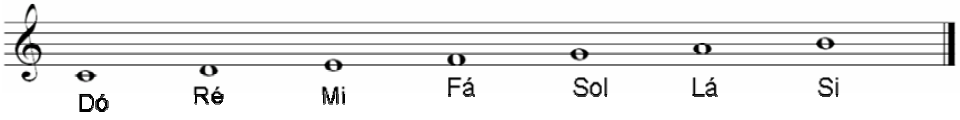


Figura 1.4: As notas musicais

Para facilitar a leitura musical e para representar todas as notas de um instrumento, como por exemplo, o piano, é necessária a utilização de duas claves, como mostra a Figura 1.5.

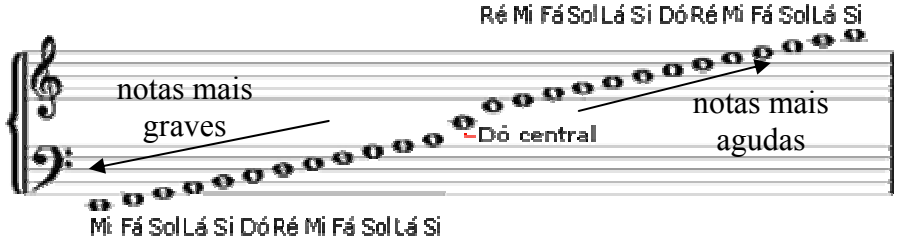


Figura 1.5: Representação das notas musicais em duas claves

As notas musicais também podem ser representadas através de números inteiros, como ilustrado na Figura 1.6:

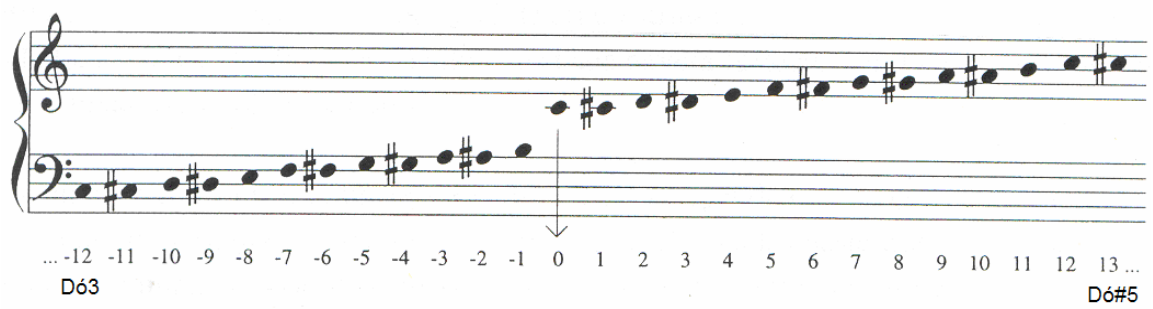


Figura 1.6: Representação das notas por valores numéricos inteiros

Caracteriza-se como intervalo a diferença de altura entre duas notas. O semitom é o menor intervalo entre duas notas. As notas musicais mostradas na Figura 1.5 não representam todas as notas com o intervalo de semitom. Para isso, foram criadas as notações que alteram as notas. Na Figura 1.6 estão representadas todas as notas com intervalo de semitom do Dó na terceira oitava (Dó3) ao Do# na quinta oitava (Dó#5). As alterações de notas são indicadas por sinais que antecedem as notas escritas no pentagrama, esses sinais são conhecidos como acidentes. Exemplos desses acidentes são o sustenido (#) que aumenta a altura da nota em um semitom e o bemol (b) que abaixa a altura da nota em um semitom.

O intervalo entre duas notas s e t pode ser determinado da seguinte forma:

$$intervalo(s,t) = t - s \tag{1.1}$$

s e t são os valores inteiros que representam as notas musicais. A Figura 1.7 ilustra alguns exemplos de intervalos musicais, escritos na clave de Fá e na clave de Sol.

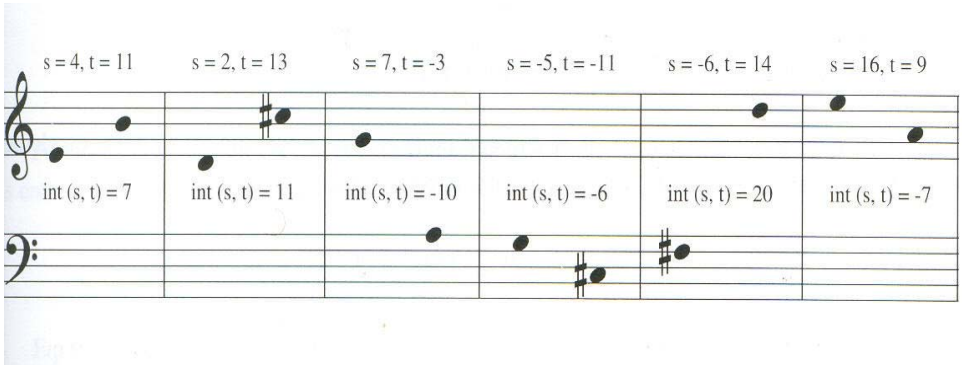


Figura 1.7: Exemplos de intervalos musicais

Os intervalos musicais podem ser classificados da seguinte forma:

0 = uníssono

1 = 2.ª menor ascendente	-1 = 2.ª menor descendente
2 = 2.ª maior ascendente	-2 = 2.ª maior descendente
3 = 3.ª menor ascendente	-3 = 3.ª menor descendente
4 = 3.ª maior ascendente	-4 = 3.ª maior descendente
5 = 4.ª perfeita ascendente	-5 = 4.ª perfeita descendente
6 = 4.ª aumentada ascendente	-6 = 4.ª aumentada descendente
7 = 5.ª perfeita ascendente	-7 = 5.ª perfeita descendente
8 = 6.ª menor ascendente	-8 = 6.ª menor descendente
9 = 6.ª maior ascendente	-9 = 6.ª maior descendente
10 = 7.ª menor ascendente	-10 = 7.ª menor descendente
11 = 7.ª maior ascendente	-11 = 7.ª maior descendente
12 = 8.ª perfeita ascendente	-12 = 8.ª perfeita descendente
13 = 9.ª menor ascendente	-13 = 9.ª menor descendente
etc.	etc.

Na música ocidental estabelece-se uma oitava[1] como um intervalo que possui uma taxa de freqüência 2:1.

A escala musical mais conhecida é a cromática ou dodecafônica com doze semitons. A Figura 1.8 mostra a escala cromática representando todas as doze notas, de dó a si, na clave de sol.

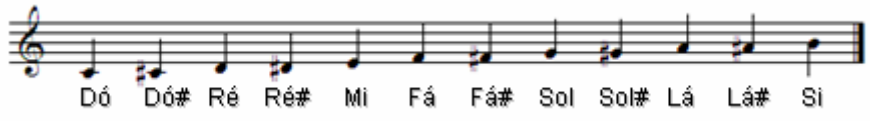


Figura 1.8: A escala cromática

Em muitos casos nesse trabalho, as notas musicais estarão representadas na forma de letras alfabéticas associadas da seguinte forma:

Letras -	A	B	C	D	E	F	G
Notas -	Lá	Si	Dó	Ré	Mi	Fá	Sol

A indicação da oitava será feita com um número na frente das letras, por exemplo, C3 representa a nota C na terceira oitava.

Cada nota musical está associada a uma freqüência. A Figura 1.9 ilustra as freqüências das notas e seus respectivos números MIDI [2], correspondentes a um teclado de piano:

[1] O nome oitava está relacionado com a seqüência de oito notas sucessivas da escala natural: Dó Ré Mi Fá Sol Lá Si Dó, o segundo Dó é dito estar uma oitava acima do primeiro.

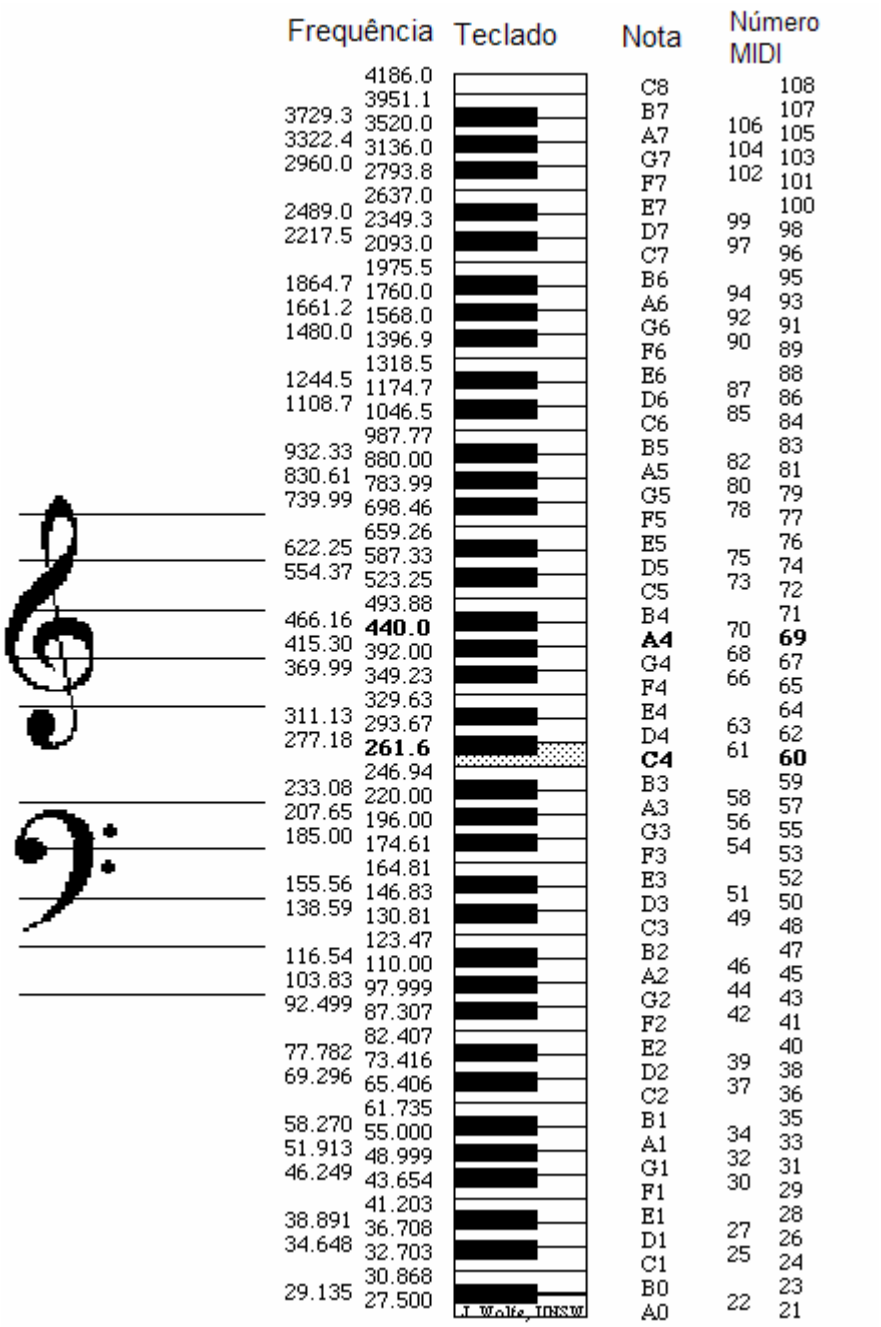


Figura 1.9: A notas musicais e suas freqüências[3]

As notas musicais atuam em ciclos proporcionais, as oitavas. Pode-se observar que, por exemplo, se a freqüência da nota A em 220 Hz é duplicada para 440 Hz, ainda é possível perceber a mesma nota A, entretanto, ela estará uma oitava acima. De maneira similar se a freqüência for diminuída pela metade, ou seja, se de 220 Hz a

[2] MIDI é a abreviatura de **M**usical **I**nstrument **D**igital **I**nterface. É um padrão utilizado para a comunicação entre instrumentos musicais e equipamentos eletrônicos, como por exemplo, teclado e computadores. Uma partitura MIDI contém instruções que determinam os instrumentos, notas, timbres, etc...Para tanto, cada nota é atribuída a um valor MIDI.

[3] Disponível em: http://www.phys.unsw.edu.au/jw/notes.html. Acesso: 06/12/2006.

freqüência da nota A for para 110 Hz, então essa nota A será percebida uma oitava abaixo.

Como complemento das relações de freqüências entre as notas, afirma-se que nosso sistema de audição trabalha de acordo com uma lei logarítmica. Assim, o fenômeno que as pessoas percebem como intervalo de altura é caracterizado por um processo logarítmico. Voltando ao exemplo anterior, a distância de 110 Hz para 440 Hz é de duas oitavas, entretanto, a razão de freqüência é quadruplicada (Figura 1.10).

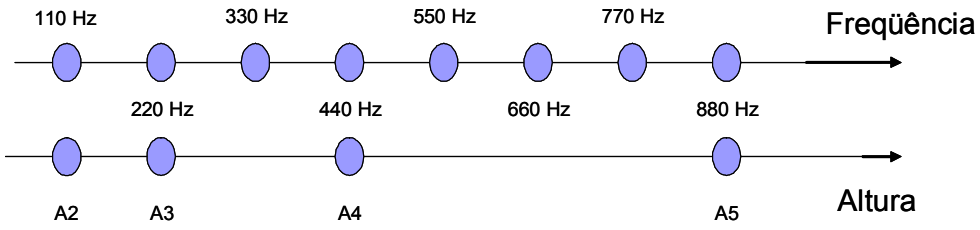


Figura 1.10: A relação das freqüências das notas musicais

Conforme anteriormente mencionado, as notas musicais são ondas sonoras complexas formadas por uma freqüência fundamental e componentes harmônicas proporcionais à freqüência fundamental [BENSON,2007]. As freqüências das componentes (fundamental e harmônicos) guardam relações matemáticas entre si, e é a freqüência fundamental que define a altura da nota. Para as notas A4 e A3, as primeiras parciais harmônicas são:

440 Hz, 880 Hz, 1320 Hz, 1760 Hz,...

220 Hz, 440 Hz, 660 Hz, 880 Hz, 1100 Hz, 1320 Hz

As oitavas são exemplos de intervalos consoantes, o que pode ser observado inclusive pelas componentes harmônicas. Por outro lado, as primeiras componentes harmônicas das notas A3 (220 Hz) e A#3 (233 Hz) são:

233 Hz, 466 Hz, 699 Hz, 932 Hz, 1165 Hz, ...

220 Hz, 440 Hz, 660 Hz, 880 Hz, 1100 Hz ...

A presença das componentes 233 Hz e 220 Hz, 466 Hz e 440 Hz etc, causa uma sensação de desconforto que é interpretado pelo ouvido como dissonância.

O intervalo musical de uma quinta justa corresponde a uma taxa de freqüência de 3:2, em que o terceiro harmônico da nota mais grave irá coincidir com o segundo harmônico da nota mais aguda, e as duas notas terão vários harmônicos em comum. Portanto, intervalos musicais em que as taxa de freqüências são números pequenos são ditos serem mais consoantes em relação aos outros intervalos[4]. O

[4] Benson [2007] afirma que essa relação apenas funciona para notas as quais os harmônicos possuem freqüências múltiplas da freqüência fundamental.

gráfico da Figura 1.11 apresenta um estudo mostrado por Benson [2007], relacionando as taxas de freqüência e os níveis de dissonância, para uma nota com sua fundamental e seis componentes harmônicas. Nota-se que o gráfico apresenta grandes picos na fundamental (1:2), no intervalo de oitava (1:2) e no intervalo de quinta justa; e apresenta picos menores nos intervalos de terça menor (5:6), terça maior (4:5), quarta justa (3:4) e sexta maior (3:5). Se mais harmônicos fossem considerados, o gráfico ganharia mais picos.

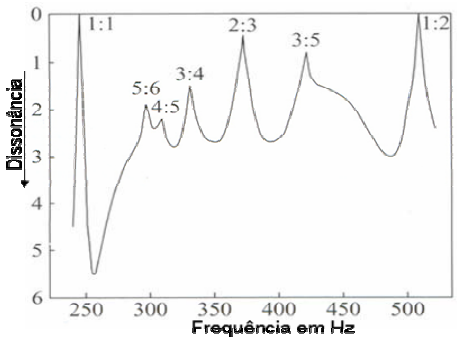


Figura 1.11: Dissonância dos intervalos musicais

1.3 – PROPRIEDADES DO SOM

Som consiste de vibrações das partículas de um meio material, que geralmente é o ar. O ar é composto por um grande número de moléculas, próximas umas das outras e que continuamente se atingem para produzir o que é percebido como pressão do ar [Benson, 2007]. Quando um objeto vibra, causa ondas de aumento e diminuição de pressão do ar. Essas ondas agitam as moléculas de ar, e se propagam, atingindo os ouvidos e produzindo a sensação sonora.

Portanto, a propagação do som se dá por meio de ondas, denominadas ondas sonoras, as quais se caracterizam por serem ondas esféricas. A onda sonora é produzida por algum objeto que produz vibrações, por exemplo, as cordas de um violão, o badalar de um sino, etc. Quando as vibrações sonoras apresentam valores estáveis de pressão o ouvido humano não consegue percebê-las, e o mesmo acontece com vibrações que contém parâmetros físicos do som e que estão fora da faixa de percepção humana. Esses limites são importantes para o processo de digitalização do som.

O som pode ser representado através de uma onda senoidal, e um ponto interessante é que complexas vibrações sonoras podem ser entendidas e sintetizadas através da combinação de ondas senoidais. Afirma-se com isso, que o sinal sonoro mais simples é a senóide.

Tal como descreve Paula Filho [2000], Smith [1997], Joaquim e Sartori [2003], a onda sonora possui os seguintes parâmetros:

- período: corresponde ao tempo necessário para se completar um ciclo, ou seja, para o padrão da onda se repetir [MIRANDA, 2001]. É representado pela letra *T*;

- ciclo: intervalo entre dois pontos de máximo ou dois pontos de mínimo no movimento ondulatório;

- freqüência: corresponde ao número de ciclos por unidade de tempo, é representada pela letra *f*. A freqüência pode ser medida em ciclos por segundo (cps), porém, a unidade de freqüência mais usada é o Hertz (Hz), que equivale a um ciclo por segundo. [MIRANDA, 2001]; e

- comprimento de onda: corresponde á distância de um ponto qualquer de um ciclo onda até o ponto correspondente do ciclo adjacente. É representado pela letra grega λ.

A velocidade de propagação da onda sonora corresponde à velocidade com que o som se propaga. Essa velocidade não depende das características da onda sonora, porém, depende da pressão e densidade do ar que são influenciados pela temperatura e altitude. Em um ambiente com a altitude próxima ao nível do mar e temperatura ambiente (25º), o som se propaga a 340 m/s, aproximadamente. Freqüência e período guardam entre si a relação:

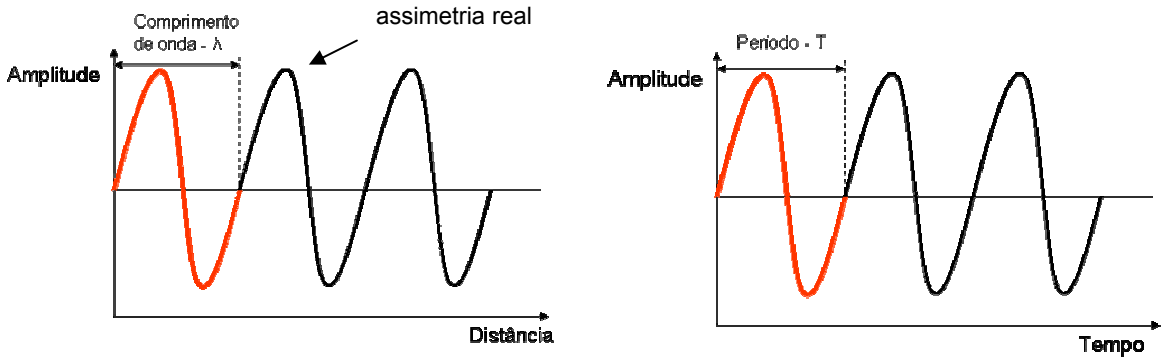
$$f = 1/T \tag{1.2}$$

No Sistema Internacional (SI), *f* é medida em Hertz e *T* em segundos, e o comprimento de onda é dado por

$$\lambda = v \cdot T \tag{1.3}$$

em que v é a velocidade de propagação.

A Figura 1.12 ilustra os parâmetros descritos acima, sendo Figura 1.12 (a) diagrama no espaço; e Figura 1.12 (b) diagrama no tempo. As distorções assimétricas são típicas de ondas acústicas reais.



(a) (b)

Figura 1.12: Onda sonora (a) no espaço (b) no tempo

Muitos objetos em movimento periódico oscilam, como por exemplo, a corda do violão, o balanço, o pêndulo do relógio. Um objeto também pode oscilar em um movimento circular, como por exemplo, um parafuso em um volante giratório interpretando um movimento uniforme e anti-horário (Figura 1.13). A esse Movimento Circular Uniforme (MCU) estará associada a seguinte velocidade angular ω (Ômega) constante:

$$\omega = \frac{2\pi}{T} = 2\pi f \tag{1.4}$$

sendo que T é o período e f a freqüência.

Figura 1.13: Volante ou Manivela [BASILIO JOAQUIM,SARTORI,2003,p.4]

Denomina-se Movimento Harmônico Simples (MHS) o movimento oscilatório unidimensional, de período igual a T e freqüência igual a f, realizado pela sombra do parafuso em relação a um plano base [JOAQUIM E SARTORI, 2007] [ROEDERER,1998]. A função horária que caracteriza esse movimento pode ser observada na Figura 1.14.

(a) (b)

Figura 1.14: Círculo Trigonométrico (a) senóide (b) fase inicial [BASILIO JOAQUIM,SARTORI,2003,p.4]

Uma vez que o ponto P da Figura 1.14 realiza um MCU, o ângulo θ varia com o tempo segundo a função:

$$\theta = \omega t + \theta_0 \quad (1.5)$$

O ângulo θ_0 representa o ângulo ou fase inicial, ou seja, o valor do ângulo θ no tempo t=0, e ω é a velocidade angular do movimento circular uniforme em rad/s (radianos por segundo), dada pela equação 1.4.

O MHS também pode ser visualizado através do movimento que a projeção do ponto P realiza em relação ao diâmetro AB, enquanto P desempenha seu movimento circular uniforme. Assim, se x representa a distância entre o centro da circunferência e o ponto Q, o uso da trigonometria permite que:

$$\cos(\theta) = x/a \text{ ou } x = a\cos(\theta) \quad (1.6)$$

Uma vez conhecido que $\theta = \omega t + \theta_0$, a distância x pode ser expressa da seguinte forma:

$$x = a\cos(\omega t + \theta_0) = a\cos(2\pi f t + \theta_0) \quad (1.7)$$

O movimento em torno do eixo x, que determina o MHS, pode, então, ser representado por uma cossenóide, tal que:

a = amplitude;

ω = velocidade angular (ou freqüência angular);

θ_0 = fase inicial.

Esse movimento seria representado por uma senóide se o mesmo fosse realizado em torno do eixo vertical y.

As ondas sonoras possuem quatro principais atributos que afetam a maneira de como elas são percebidas [BENSON,2007]. Esses atributos estão listados na Tabela 1.1 e estão descritos nas sessões 1.3.1, 1.3.2, 1.3.3 e 1.3.4. O primeiro deles é a amplitude, o qual determina a potência da vibração, e é percebido como intensidade. O segundo atributo é a altura que corresponde à freqüência da vibração. O terceiro atributo é o timbre, o qual corresponde ao formato da onda sonora. O quarto é a duração, que significa o intervalo de tempo em que uma nota é soada.

Segundo Benson [2007], essas noções dos atributos sonoros precisam ser alteradas por várias razões. Uma delas é o fato de que a maioria das vibrações sonoras não possui uma única freqüência. Além disso, Benson [2007] relata que esses atributos deveriam ser definidos em termos de percepção sonora, e não em termos da onda sonora propriamente dita. Por exemplo, a percepção de altura de um som pode representar uma freqüência não necessariamente presente na onda sonora,

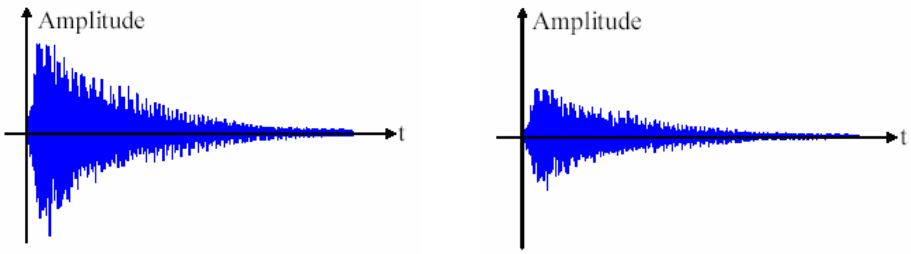
caracterizando um fenômeno conhecido como “ausência da fundamental[5]” e é parte de um tema chamado, em inglês, de *psychoacoustics*.

Tabela 1.1: Atributos do som

Físicos	Perceptivos
Amplitude	Intensidade
Freqüência	Altura
Espectro	Timbre
Duração	Duração

1.3.1 – A intensidade

Uma das principais características da intensidade corresponde à distinção de sons fortes e de sons fracos. Em música, a intensidade está relacionada com a dinâmica. A dinâmica é a forma de manipular a intensidade sonora na execução musical. Essa característica está relacionada com a amplitude da vibração sonora, ou seja, a percepção de sons fortes ou fracos está diretamente relacionada com a potência acústica presente no sinal. Vale ressaltar que sons com diferentes amplitudes podem possuir a mesma freqüência, como na Figura 1.15:



(a) Amplitude de grande intensidade

(b) Amplitude de pequena intensidade

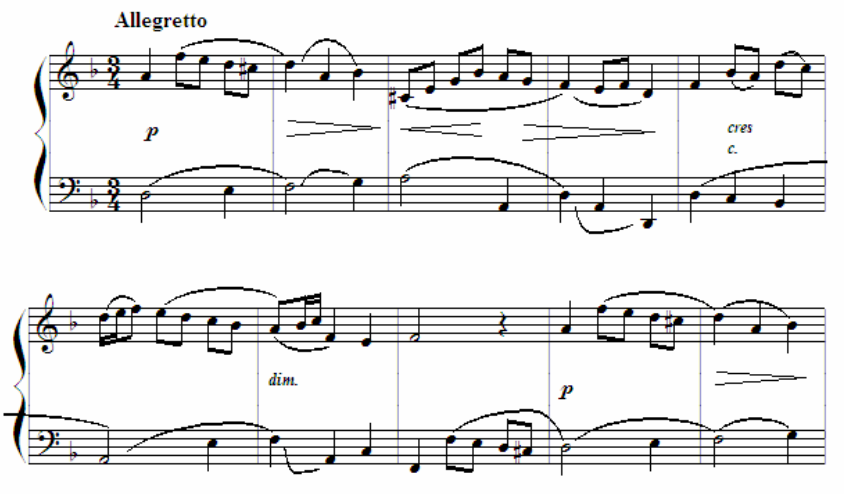
Figura 1.15: Sinais com mesma freqüência mas com amplitudes diferentes

As Figuras 1.15 (a) e 1.15 (b) podem representar o teclar forte e fraco, respectivamente, de uma nota do piano. Esses dois sons possuem a mesma freqüência e comprimento de onda. Suas amplitudes são diferentes para representar a intensidade[6] do som. O sinal representado por uma forma de onda mais intensa ou mais cheia tem

[5] Um som é dito ter ausência da fundamental (*missing fundamental*, em inglês) quando é possível determinar a altura da nota, mesmo quando a componente fundamental não está presente.

[6] A intensidade das notas que compõem uma música é componente importante da execução dessa música, pois contribui para despertar emoções e expectativas no ouvinte. A intensidade de uma nota pode ser controlada, no violão, pela força exercida sobre a corda ao tocá-la, no violino, pela força do arco sobre a corda, no piano, pela força com que a tecla é tocada, etc.

amplitude maior que a do outro sinal. Esse trabalho também irá considerar como dinâmica as variações de andamento[7] (*andante, allegro, moderado*), e as articulações[8] (*legato*, *staccato*). Algumas manifestações da dinâmica estão expressas na partitura da Figura 1.16:



Allegretto: uma das formas de andamento.

: Indicação de diminuição de velocidade.

: Indicação de aumento de velocidade.

p: Indicação de pouca intensidade (piano)

cresc.: Indicação de aumento de intensidade

dim.: Indicação de diminuição de intensidade

Figura 1.16: Exemplos de dinâmicas musicais

Em homenagem a Alexander Graham Bell a unidade fundamental de medida da intensidade do som no SI é o Bell, cujo símbolo é B. Uma potência de 10^{-12} W/m^2 representa 0B (zero Bell), e está aproximadamente dentro da região do som mais fraco que o ouvido humano consegue captar. Adicionar 1B equivale a multiplicar a potência por um fator de dez. Portanto, multiplicar a potência por um fator k equivale a adicionar $\log_{10} k$ Bell ao sinal. Na prática, é utilizado comumente o submúltiplo dB (decibel). Assim, a escala é logarítmica, e *n* decibéis representa uma potência de $10^{(n/10)-12}\ W/m^2$ [BENSON,2007].

O limite de audição indica a intensidade do som mais fraco que o ser humano consegue ouvir. O valor desse limite em decibéis varia de acordo com a freqüência do sinal. O ouvido humano é sensível a freqüências um pouco acima de 2000 Hz, quando o limite de audição de uma pessoa normal é de aproximadamente 0 dB. Aos 100 Hz o limite é de aproximadamente 50 dB, e aos 10000 Hz, de aproximadamente 30 dB. Uma conversação atinge em torno de 60 – 70 dB, e o limiar da sensação de dor é de 130 dB.

[7] O andamento corresponde à velocidade com que a música é tocada.
[8] As articulações são maneiras de como uma nota ou grupo de notas é emitido, “enfatizado”.

1.3.2 – A freqüência

Tal como descreve Paula Filho [2000], Smith [1997], Joaquim e Sartori [2003], a freqüência corresponde à percepção de sons agudos ou graves. Quando as senóides se apresentam mais comprimidas horizontalmente (menor período), a percepção é para sons mais agudos e o contrário para sons mais graves. Assim, os sons graves possuem baixas freqüências, e os sons agudos possuem altas freqüências.

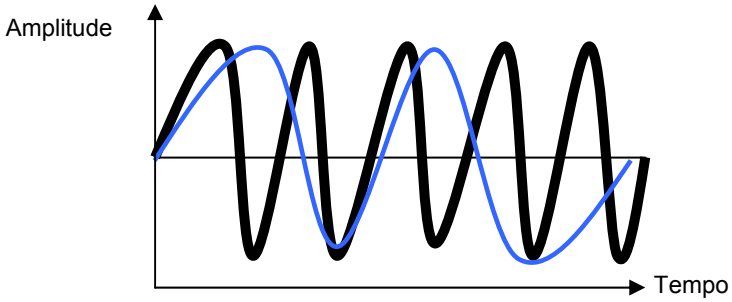


Figura 1.17: Sinais com diferentes freqüências

A Figura 1.17 mostra dois sinais com diferentes freqüências. Pode-se observar que a freqüência do sinal representado pelo traço de cor preta é maior que a do outro sinal, representado pela cor azul. Assim, esse sinal com freqüência maior representa um som mais agudo em comparação com o sinal de cor azul, o qual representa um som mais grave.

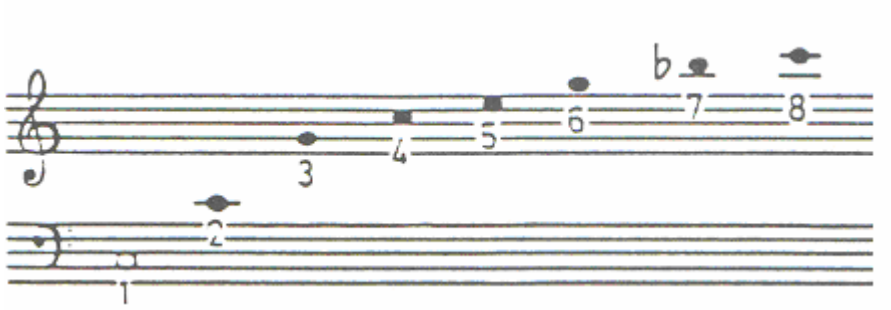


Figura 1.18: A freqüência fundamental do C3 (125 Hz) e sete dos seus harmônicos: C4 (250 Hz), G4 (375 Hz), C5 (500 Hz), E5 (625 Hz), G5 (750 Hz), Bb5 (875 Hz), C6 (1000 Hz). [SANO E JENKINS, 1989]

Quando uma nota em um instrumento de cordas ou de sopro é produzida numa certa altura, com freqüência f, o som é essencialmente periódico nessa freqüência. A teoria da Série de Fourier relata que tal som pode ser decomposto em uma soma de senos e cossenos com amplitudes adequadas (ou de senos com amplitudes e fases adequadas), cujas freqüências são múltiplos da freqüência f. A

componente desse som com freqüência f é denominada fundamental e determina a altura da nota. A componente com freqüência nf representa o n-ésimo harmônico.

A Figura 1.18 representa a série de harmônicos baseada na fundamental C3 (dó na terceira oitava), representada pelo número 1. O segundo harmônico é o C4, representado pelo número 2, e assim por diante. O sétimo harmônico é, na verdade, algo bem próximo do Bb4 (Si bemol na quarta oitava). Na moderna escala de doze temperamentos iguais[9], até mesmo o terceiro e quinto harmônico são levemente diferentes das novas G (sol) e E (mi) [BENSON,2007] [SANO E JENKINS,1989].

O ouvido humano consegue distinguir freqüências no intervalo aproximadamente entre 20 Hz e 20000 Hz (ou 20 kHz). Para freqüências fora desse intervalo, não há ressonância na membrana basilar do ouvido. Esses intervalos variam de pessoa para pessoa e também são influenciados pela idade. Contudo, raramente composições musicais utilizam freqüências maiores que 4000 Hz [MIRANDA, 2001] [BENSON, 2007].

1.3.3 – O Timbre

O timbre está relacionado com a origem do som, ou seja, a fonte sonora. É o timbre que possibilita distinguir sons com a mesma altura e a mesma intensidade e duração tocados por instrumentos diferentes. Miranda [2001] afirma que o timbre representa o domínio da percepção imediata, e que talvez essa percepção imediata esteja relacionada com o fato de que os sons podem indicar perigo, atenção, entre outros. Assim, é natural que o ser humano seja capaz de distingui-lo rapidamente e reagir ao que está causando tal som.

Conforme anteriormente mencionado, a grande maioria dos sons não possui uma freqüência pura, ou seja, o som é composto por várias freqüências. A primeira freqüência, que é a mais baixa de todas, é a fundamental (1º harmônico) e determina a altura do som. As demais freqüências são múltiplas dessa fundamental e são denominadas freqüências harmônicas ou harmônicos. Então, o som produzido não tem a forma de uma onda senoidal, pois esse som é um som composto, e sua forma de onda, apesar de continuar sendo periódica, dependerá do instrumento origem. O timbre será determinado pelo número de harmônicos presentes no som e também pela

[9] Temperamento é a divisão de uma oitava através do ajuste de intervalos entre as notas. Na escala de doze temperamentos iguais a oitava é dividida em 12 semitons iguais, ou seja, cada semitom corresponde a um intervalo de $2^{1/12}$.

amplitude de cada um desses harmônicos[10] (obtidos pela expansão em série do Fourier do sinal sonoro).

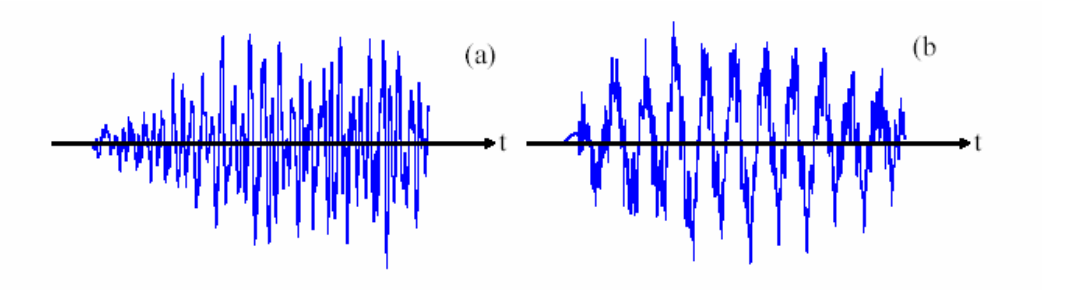


Figura 1.19: O timbre (a) Sons do piano (b) Sons do violão [BASÍLIO JOAQUIM, SARTORI, 2003]

A Figura 1.19 ilustra duas formas de onda que correspondem a dois sons que possuem a mesma amplitude e freqüência, porém tocados por instrumentos diferentes. Portanto, possuem formas de onda diferentes. Um som é rico em harmônicos quando sua forma de onda é complexa, representando assim a composição de muitas componentes harmônicas. Se a forma de onda é similar a uma senóide, então o som é pobre em harmônicos, pois isso significa que este som quase não possui componentes harmônicas.

A Figura 1.20 mostra essa comparação. Na Figura 1.20 (a) tem-se uma senóide simples, e em Figura 1.20 (b) tem-se uma onda complexa com o mesmo período, formada por senóides de deferentes freqüências.

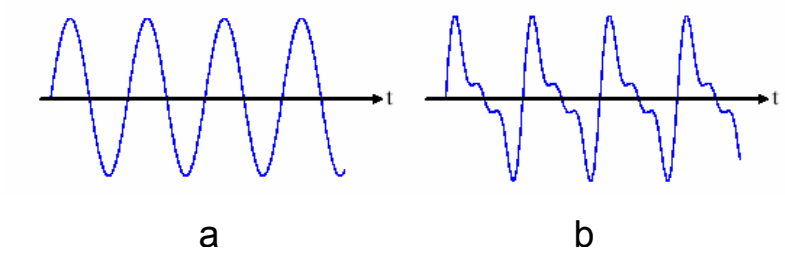


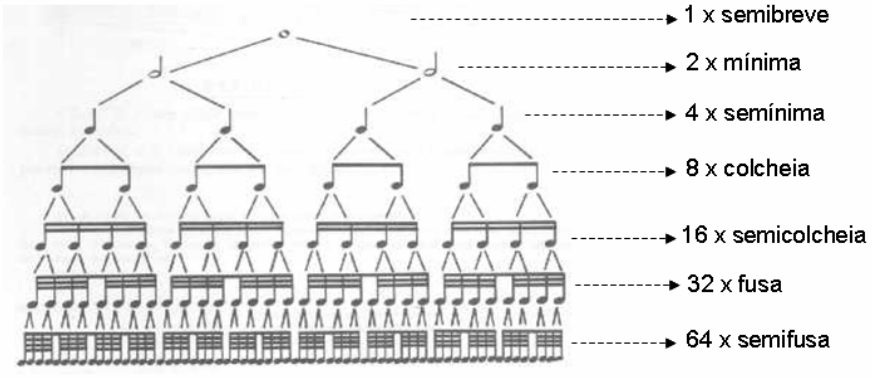
Figura 1.20: O timbre (a) Senóide (b) Onda complexa [BASILIO JOAQUIM,SARTORI,2003]

1.3.4 – A duração

Conforme Miranda [2001], em música a duração é o tempo em que uma nota é tocada ou o tempo entre duas notas (pausas). A duração é o elemento que determina o ritmo. A relação de duração entre as figuras musicais (representação gráfica das notas musicais com informação de duração) pode ser observada na Figura

[10] A distinção do timbre também é influenciada pelo envelope sonoro, que caracteriza como o som se inicia, se mantém e termina ao longo do tempo. O envelope é composto basicamente de quatro elementos: ataque, decaimento, sustentação e relaxamento [ROEDERER,1998] [BENSON, 2007].

1.21 (a), onde o número indica a quantidade de notas necessárias para a duração correspondente à primeira figura (semibreve). Cada figura musical tem uma figura de pausa correspondente como ilustrado na Figura 1.21 (b).



(a)

N°	Nome da Figura	Figura	Pausa	Duração
1	Semibreve			$\frac{1}{1}$
2	Mínima			$\frac{1}{2}$
4	Semínima			$\frac{1}{4}$
8	Colcheia			$\frac{1}{8}$
16	Semicolcheia			$\frac{1}{16}$
32	Fusa			$\frac{1}{32}$
64	Semifusa			$\frac{1}{64}$

(b)

Figura 1.21: As figuras musicais (a) Relações entre as figuras musicais (b) números, nomes, figuras musicais e pausas, e durações correspondentes

Miranda [2001] ainda menciona sobre o domínio do pulso, que está relacionado com a duração das notas. Nesse domínio tem-se a idéia de que altura e ritmo são considerados como um fenômeno contínuo do domínio do tempo. Geralmente

tem-se o auxílio de um contador de tempo, que indica a velocidade do pulso. Carpinteiro [1995] trabalha com esse domínio do pulso, como será visto mais adiante. O ser humano consegue identificar ritmos distintos em até aproximadamente 10 ciclos por segundo. Apesar desse limite, os seres humanos se sentem mais confortáveis com freqüências que estão próximas das batidas do coração, que alcançam, aproximadamente, de 30 batidas até 240 batidas por minuto [0,5 Hz (30 x 1/60), e 4 Hz (240 x 1/60)]. Portanto, os ritmos musicais geralmente estarão nesse intervalo, ou seja, de 4 pulsos por segundos (4 Hz) e um pulso a cada dois segundos (0,5 Hz).

Logo, a noção de tempo está relacionada com a quantidade de batidas. Após a invenção do metrônomo de Maezel em 1810, a indicação do tempo se tornou mais precisa e pode estar indicada, numa partitura, em batidas por minuto pela abreviação M.M. (metrônomo de Maezel), a figura de referência para a batida e um número. Por exemplo, M.M. ♩ =126 indica um tempo de 126 batidas por minuto. [MIRANDA, 2001]. Um exemplo dessa indicação pode ser observado na Figura 1.22. Nessa partitura a semínima (♩) é a unidade de tempo e vale 1/126 minuto, um pouco menos que meio segundo.

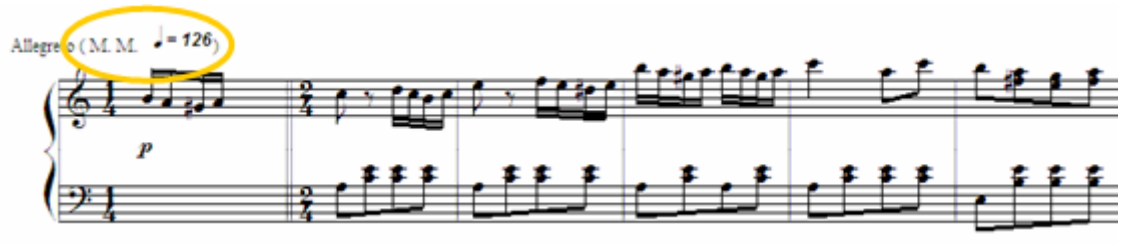
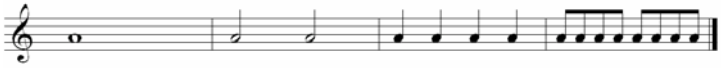


Figura 1.22: Possível indicação do tempo em uma melodia

Essas batidas são divididas em grupos, formando assim, os compassos[11]. No começo da partitura da Figura 1.23, logo após a clave, existem dois números que indicam o compasso dessa música: o numerador indica a quantidade de tempos dentro de um compasso (unidade de compasso) e o denominador indica a unidade de referência para a batida (unidade de tempo), ou seja, a figura de som que representa uma unidade de tempo. Um exemplo das relações entre as figuras musicais pode ser observado na Figura 1.23:

[11] Compasso é a divisão da melodia em tempos iguais. Cada compasso possui a mesma quantidade de tempos. E a duração de todas as notas dentro de um compasso deve somar essa quantidade. Exemplo:



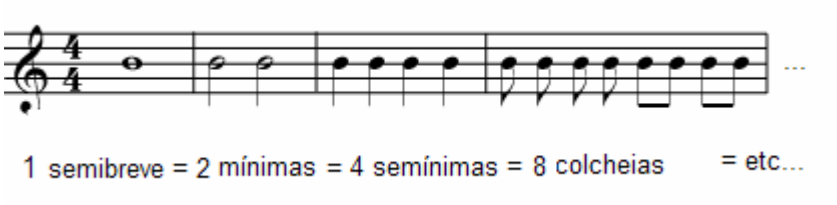


Figura 1.23: Exemplos da relação entre as figuras musicais

1.4 – SÉRIE DE FOURIER

Segundo Joaquim e Sartori [2003], em sua grande maioria, os sinais elétricos são representados no domínio do tempo. Porém, em algumas áreas relacionadas com processamento de sinais, a análise dos sinais se torna mais fácil quando esses passam a ser representados no domínio da freqüência. A análise do domínio da freqüência é mais simples e direta e traz informações sobre as componentes senoidais envolvidas em um sinal, e através dessa análise, é possível observar as propriedades e parâmetros dos sinais, os quais dificilmente serão possíveis de observar no domínio do tempo. Para a análise no domínio da freqüência, a análise e a transformada de Fourier são essenciais.

Jean Baptiste Joseph Fourier (1768-1830) contribuiu para uma das mais valiosas descobertas quando se trata de sinais: qualquer função periódica pode ser decomposta em componentes de ondas senoidais que possuem amplitudes variadas e freqüências que são múltiplas da freqüência da fundamental. Isso constitui a série de Fourier.

Para uma função periódica, $x(t)$, com período T, a série de Fourier consiste em:

$$x(t) = \frac{a_0}{2} + a_1 \cos(2\pi f_0 t) + a_2 \cos(2\pi 2 f_0 t) + ... + b_1 \text{sen}(2\pi f_0 t) + b_2 \text{sen}(2\pi 2 f_0 t) + ... \quad (1.7)$$

$a_0 / 2$ = valor médio da função;

α_1, b_1 = amplitudes do primeiro harmônico ou a componente fundamental;

α_2, b_2 = amplitude do segundo harmônico;

α_3, b_3 = amplitude do terceiro harmônico;

f_0 = 1/T, freqüência fundamental do sinal;

A equação 1.8 também pode ser escrita da seguinte forma compacta:

$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[a_n \cos(2\pi n f_0 t) + b_n \text{sen}(2\pi n f_0 t) \right] \quad (1.8)$$

É imprescindível que o sinal seja periódico. Diz-se que uma função $x(t)$ é periódica se existe um número T tal que $x(t) = x(t+T)$, para todo t. Como a série de Fourier pode ser definida dentro de um intervalo possível, não há necessidade da verificação da periodicidade o tempo todo.

Outros pontos devem ser observados e estão ilustrados na Figura 1.24:

- se há descontinuidades, a função deve ter um número finito delas dentro de um período.

- a função deve ter um número finito de máximos e mínimos dentro de um período.

- possibilidade da integração de uma função em um período, tal que

$$\int_{t}^{t+T} |x(t')|\, dt' < \infty \tag{1.9}$$

em que $x(t)$ descreve a função.

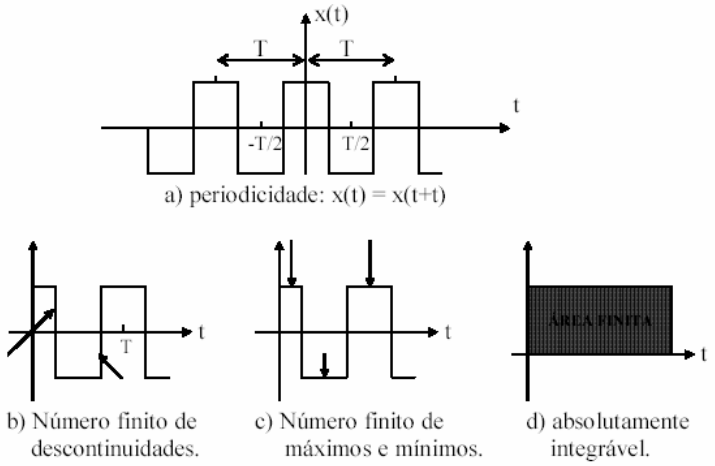


Figura 1.24: Requisitos para a Série de Fourier [BASÍLIO JOAQUIM,SARTORI,2003,p.13]

Uma simplificação da série de Fourier pode ser descrita por:

$$x(t) = E_0 + \sum_{n=1}^{\infty} E_n \cos(2\pi n f_0 t + \theta_n) \tag{1.10}$$

E_0 = α_2 /2;

E_n = amplitude do n-ésimo harmônico;

θ_n = fases do n-ésimo harmônico.

As informações de amplitude e fase estão presentes em cada um dos termos.

É possível utilizar a Série de Fourier somente com os senos ou somente com os cossenos. Para isso, é necessário utilizar a fase. A fase nada mais é que um deslocamento da onda sonora. Isso parte da seguinte relação trigonométrica para a soma de arcos:

$$\text{sen}(a+b) = \text{sen}(a)\cdot\cos(b) + \text{sen}(b)\cdot\cos(a) \tag{1.11}$$

Para obtenção dos coeficientes da série mostrada na equação 1.9, vale ressaltar que os termos $\cos(2\pi nf_0 t)$ e $\text{sen}(2\pi nf_0 t)$ formam uma base ortogonal completa, isto é:

$$\int_{-T/2}^{T/2} \cos(2\pi nf_0 t)\cos(2\pi mf_0 t)dt = \begin{cases} 0, & se\ \ m \neq n \\ T/2, & se\ \ m = n \neq 0 \end{cases} \tag{1.12}$$

$$\int_{-T/2}^{T/2} \text{sen}(2\pi nf_0 t)\,\text{sen}(2\pi mf_0 t)dt = \begin{cases} 0, & se\ \ m \neq n \\ T/2 & se\ \ m = n \neq 0 \end{cases} \tag{1.13}$$

$$\int_{-T/2}^{T/2} \text{sen}(2\pi nf_0 t)\cos(2\pi mf_0 t)dt = 0 \tag{1.14}$$

Para n = 0,1,2,3,4,5..., os coeficientes a_n da Série são obtidos da seguinte forma:

$$a_n = \frac{2}{T}\int_{-T/2}^{T/2} x(t)\cos(2\pi nf_0 t)dt \tag{1.15}$$

E para o a_0, que equivale ao termo médio da função $x(t)$, a expressão é a mesma, e o parâmetro n terá valor 0.

Para $n = 1,2,3,4...$, os coeficientes b_n da Série são obtidos da seguinte forma:

$$b_n = \frac{2}{T}\int_{-T/2}^{T/2} x(t)\,\text{sen}(2\pi nf_0 t)dt \tag{1.16}$$

E os coeficientes E_n, e θ_n da seguinte forma:

$$E_n = \sqrt{a_n^2 + b_n^2} \quad \text{e} \quad \theta_n = arctg\frac{b_n}{a_n} \tag{1.17}$$

Tomando o instante zero como referência, é necessária uma informação adicional da posição da senóide, em relação a esse instante. Essa informação é obtida através da fase. A fase indica a posição inicial da harmônica no instante inicial considerado. Por convenção a fase será negativa quando o pico positivo mais próximo acontecer depois do instante zero, ou seja, a senóide está atrasada. Será positiva quando o pico positivo mais próximo acontecer antes do instante zero, ou seja, a senóide está avançada. Tanto a fase e a amplitude de uma componente são representadas, no domínio da freqüência, por um impulso com amplitude igual à da senóide, e a abscissa desse impulso será a freqüência correspondente da senóide. A Figura 1.25 ilustra alguns exemplos de fases.

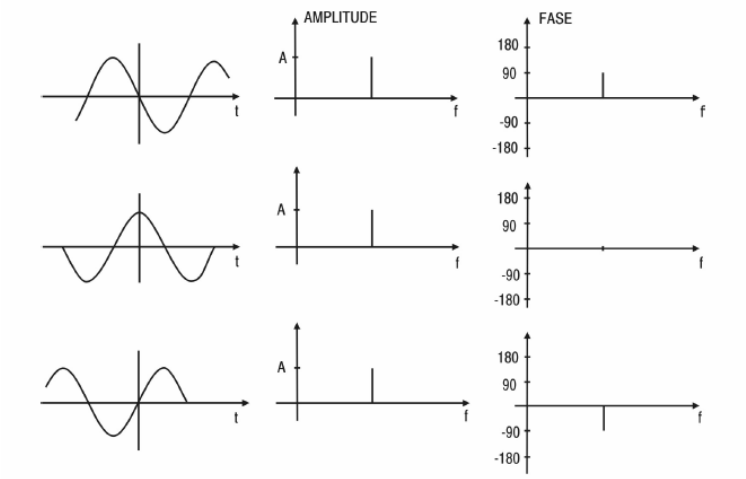


Figura 1.25: Exemplos de fases [BASILIO JOAQUIM,SARTORI,2003,p.3]

Um sinal não senoidal (exemplificado na Figura 1.26) é composto por uma série de componentes senoidais com amplitudes, freqüências e fases determinadas. Esse sinal passa a ter cristas e vales. Se um sinal elétrico é dito periódico, então ele pode ser representado.

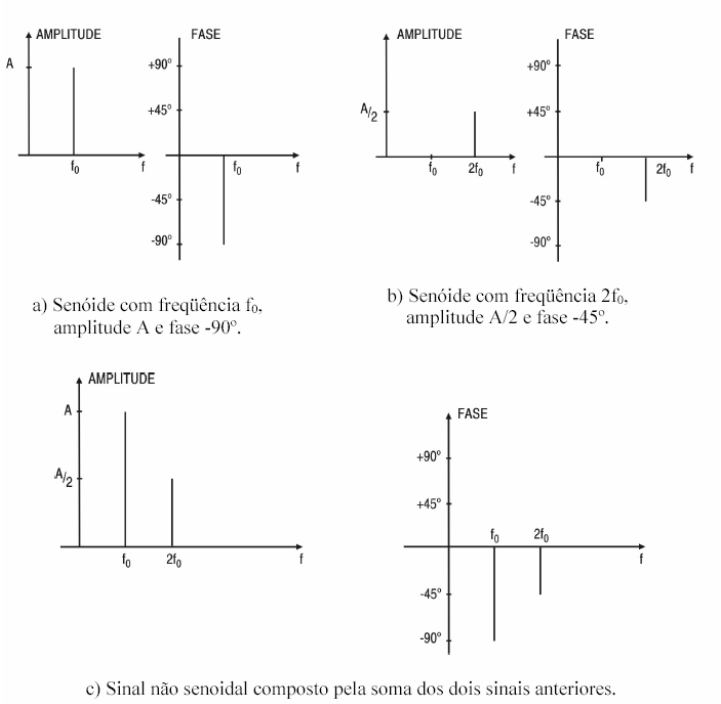


Figura 1.26: Composição de dois sinais senoidais [BASILIO JOAQUIM,SARTORI,2003,p.4]

A figura 1.27 a seguir ilustra o sinal resultante da figura anterior no domínio do tempo.

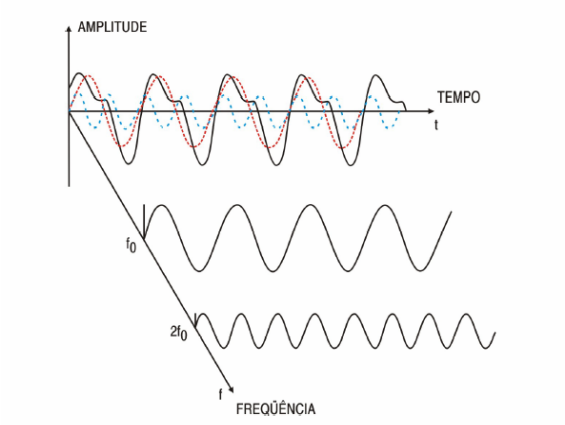


Figura 1.27: Sinal resultante não senoidal [JOAQUIM;SARTORI,2003,p.5]

A esse processo é possível adicionar muitas senóides, e os resultados serão sempre diferentes, ou se simplesmente houver uma alteração na freqüência, amplitude ou fase das componentes senoidais, essas alterações também implicam em uma onda resultante diferente (Figura 1.28).

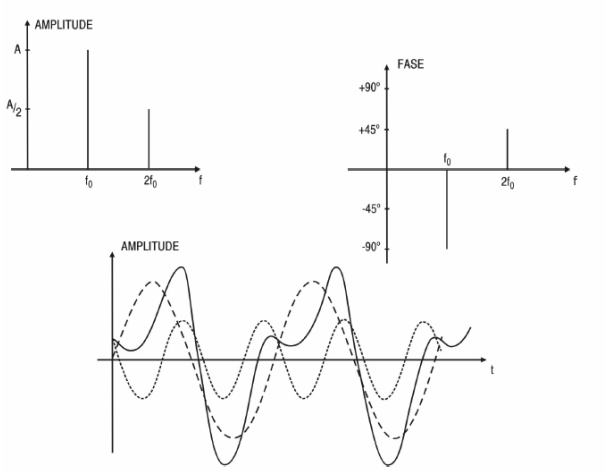


Figura 1.28: Soma de dois sinais senoidais com variação na fase de uma das componentes [JOAQUIM,SARTORI,2003,p.6]

1.4.1 - Propriedades da série de Fourier

Ainda segundo Joaquim e Sartori [2003], algumas propriedades em relação à Série de Fourier devem ser consideradas, entre elas:

1 - Para calcular os coeficientes da série de Fourier de uma função periódica é importante que a integração possa ser efetuada dentro de um intervalo de tempo que corresponde a um período do sinal. Assim, sinais periódicos determinam que,

$$\int_0^T x(t)\cos(2\pi n f_0 t)\,dt = \int_{t_0}^{t_0+T} x(t)\cos(2\pi n f_0 t)\,dt \quad \forall t_0 \tag{1.18}$$

$$\int_0^T x(t)\,\mathrm{sen}(2\pi n f_0 t)\,dt = \int_{t_0}^{t_0+T} x(t)\,\mathrm{sen}(2\pi n f_0 t)\,dt \quad \forall t_0 \tag{1.19}$$

2 – Se uma função é par, todos os coeficientes b_n da Série de Fourier serão nulos. Sendo assim, haverá somente os termos a_n dos cossenos. Uma função é par se: $x(-t) = x(t), \forall t$.

3 - Se uma função é ímpar, todos os coeficientes a_n da Série de Fourier serão nulos, sendo assim, haverá somente os termos b_n dos senos. Uma função é ímpar se, $x(-t) = -x(t), \forall t$.

1.5 – O SISTEMA AUDITIVO HUMANO

Uma faculdade comum, porém muito interessante que está presente no ser humano é a habilidade de reconhecer, reproduzir e analisar melodias ou pedaços de músicas. Longuet-Higgins [1979] e Temperley [2001] mencionam que essa habilidade abrange tanto a capacidade de identificar ritmos, relação entre as tonalidades, alturas, harmonia, quanto julgar se uma nota ou frase foi tocada fora do tempo ou está fora de tonalidade.

Goga e Goga [2004] dizem que a música é sentida pela porção do cérebro responsável pelos sentimentos, e não simplesmente visualizada pela porção cerebral responsável pela razão e inteligência. Para esses autores, os compositores podem influenciar o ser humano, no sentido de trazer alegrias, paz, tristeza, melancolia, entre outros. Os autores ainda afirmam que a circulação sangüínea e respiração podem sofrer influências dependentes do tipo de música escolhida pelo ouvinte. Por isso também é importante que as composições musicais sejam bem planejadas e estudadas.

O ouvido é dividido basicamente em três partes (Figura 1.29): ouvido externo, ouvido médio ou tímpano e ouvido interno, ou labirinto [LENT,2002] [BENSON,2007]. O ouvido externo representa a parte visível, composto pelo pavilhão auricular, pela concha e pelo meato auditivo externo, onde as ondas sonoras são concentradas, amplificadas e transmitidas para os receptores. O meato auditivo externo possui aproximadamente 2,7 cm e termina no tímpano ou membrana timpânica que vibra ao ser incidida por um estímulo sonoro. O tímpano separa o ouvido externo do ouvido médio. O ouvido médio representa uma cavidade cheia de ar na qual estão localizados ossículos articulados entre si (martelo, bigorna e estribo), responsáveis por transmitir as vibrações do tímpano para outra membrana que veda um orifício denominado janela oval. Essa membrana da janela oval separa o ouvido médio do ouvido interno, representado pela cóclea. A cóclea é uma cavidade óssea em forma de caracol onde estão os receptores auditivos. Um de seus propósitos é separar o som em várias componentes de freqüências antes de transformá-lo em impulso nervoso.

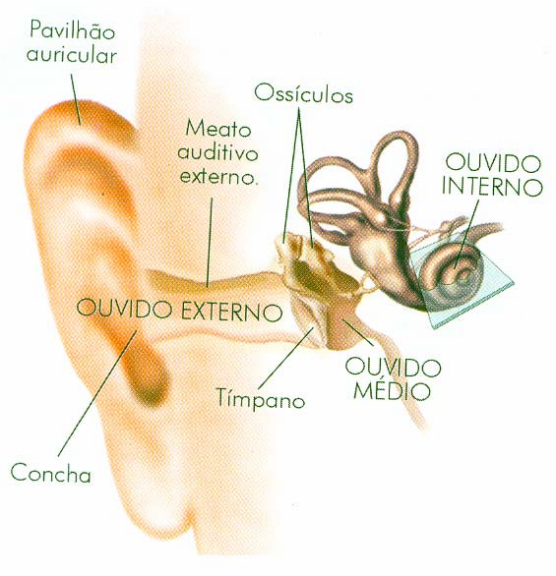


Figura 1.29: O ouvido [LENT,2002,pp.190]

Portanto, a detecção de freqüência é desempenhada por vibrações da membrana basilar na cóclea do ouvido. A membrana basilar é mais estreita e rígida na base do que no ápice da cóclea. Na Figura 1.30 (a) está ilustrada a cóclea, que é o órgão receptor do sistema auditivo. Na Figura 1.30 (b) está ilustrado o corte de uma volta da cóclea, em que se pode observar a membrana basilar. As freqüências mais baixas, os sons mais graves, fazem vibrar regiões da membrana basilar próximas ao ápice da cóclea e não conseguem mover com facilidade as regiões próximas à base, conforme observado na Figura 1.31 (a). Ao contrário, as freqüências mais altas, os sons mais agudos, fazem vibrar regiões da membrana basilar perto da base e menos a região perto do ápice, conforme observado na Figura 1.31 (b) [LENT, 2002].

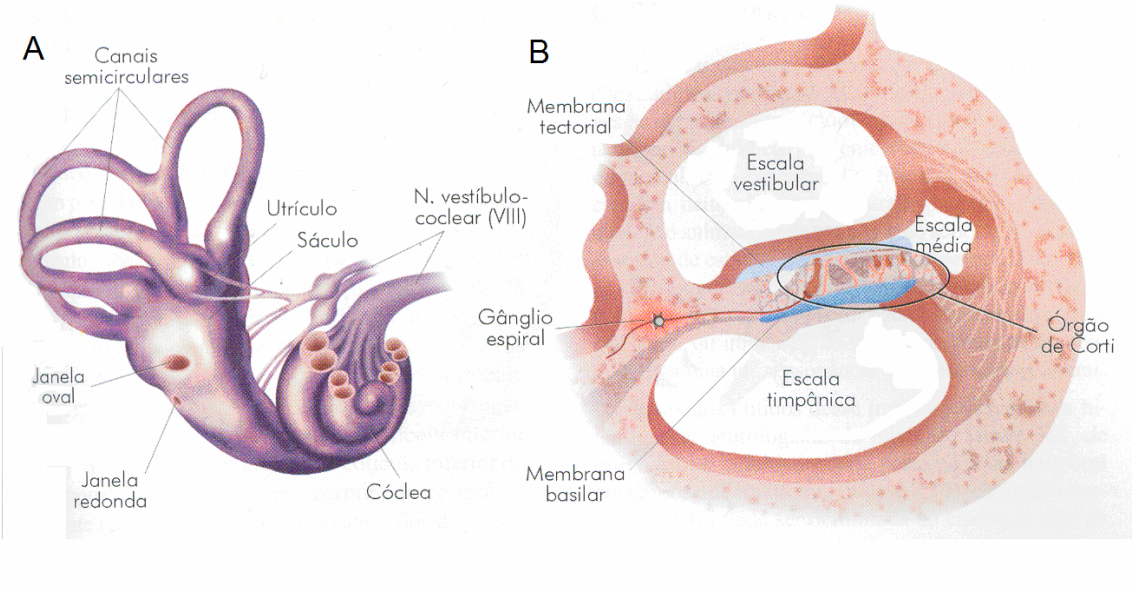


Figura 1.30: Parte do sistema auditivo humano. (A) A cóclea e (B) Mostra de um corte transversal da cóclea. [LENT, 2002]

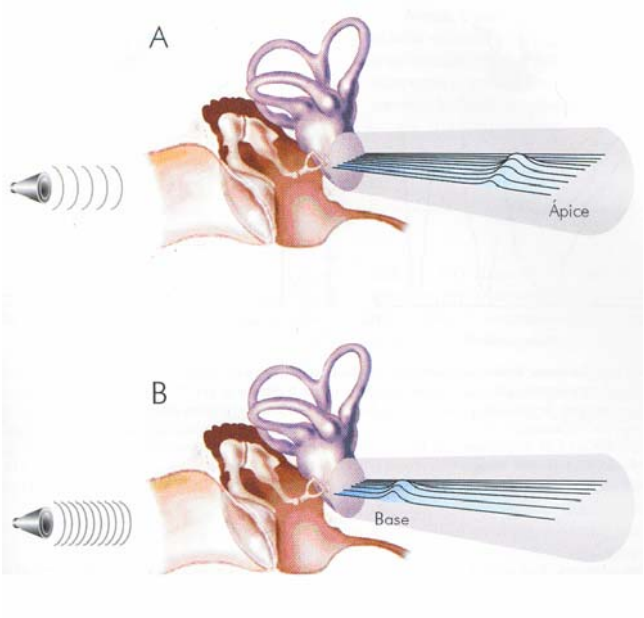


Figura 1.31: A tonotopia representa uma especialização da membrana basilar: os sons mais graves fazem vibrar o ápice (A), e os mais agudos movimentam a base (B). [LENT, 2002]

Ainda segundo Longuet-Higgins [1979], algumas teorias para o entendimento da percepção musical podem ser consideradas. Sobre a percepção da tonalidade, considera-se a Figura 1.32 como ilustração. Nessa figura são mostradas as notas musicais nas linhas numeradas de y=-2 a y=3 e nas colunas numeradas de x =- 3 a x=4, onde um intervalo numa linha corresponde a uma quinta (7 semitons) e um intervalo numa coluna corresponde a uma terça maior (4 semitons). A principal associação que o ouvinte interpreta ao ouvir cada nota está associada com a tonalidade estendida das primeiras duas notas, porém se as notas se moverem drasticamente no espaço harmônico, o ouvinte será forçado a buscar uma nova tonalidade de acordo com os intervalos menos remotos. Por exemplo, a tonalidade estendida de C[12], abrange as notas referentes à escala de C maior e notas referentes à escala de C menor mais duas notas extras, como ilustrado do retângulo da Figura 1.32. Portanto, o ouvinte começa a ouvir uma seqüência de notas e assume que a primeira nota seja a tônica[13] da tonalidade e atribui à essa tônica a tonalidade estendida. Se a segunda nota for coerente com a hipótese levantada, por exemplo, se representar a nota mais a direita da tônica ou a imediatamente acima dela, então a hipótese é mantida. Caso contrário, a

[12] Notas da escala de C maior: C-D-E-F-G-A-B. Notas da escala de C menor: C-D-Eb-F-G-Ab-Bb. As duas notas extras (Db e F#) do exemplo da Figura 1.32 não pertencem a nenhuma dessas duas escalas .
[13] A escala musical é uma seqüência de 7 notas sucessivas. A tônica é o primeiro grau da escala musical e determina o tom ou a tonalidade da música.

primeira nota estará mais propícia para ser a dominante, então a tonalidade estendida será movida uma nota para a esquerda no espaço harmônico.

$y = 3$	D♯	A♯	E♯	B♯	F×	C×	G×	D×
2	B	F♯	C♯	G♯	D♯	A♯	E♯	B♯
1	G	D	A	E	B	F♯	C♯	G♯
0	E♭	B♭	F	C	G	D	A	E
−1	C♭	G♭	D♭	A♭	E♭	B♭	F	C
−2	A♭♭	E♭♭	B♭♭	F♭	C♭	G♭	D♭	A♭
$x =$	−3	−2	−1	0	1	2	3	4

Figura 1.32: Espaço harmônico sugerido por Longuet-Higgins [1979]

Assim, tanto a percepção de tonalidade quanto a percepção de ritmo[14] envolve a interação entre o que o ouvinte está ouvindo e ponto de referência e conhecimento criado por ele. Para a percepção de ritmo, esse ponto de referência é o tempo e o metrônomo e na percepção de tonalidade o ponto de referência é a escala formada pela tonalidade estendida LONGUET-HIGGINS [1979].

Griffith e Todd [2001] estudaram a percepção da altura ou tonalidade relacionando-as com o reconhecimento do instrumento musical. Os autores também mencionam sobre a capacidade dos seres humanos em reconhecer um centro tonal e a função de cada altura dentro desse centro, de memorizar melodias e reproduzi-las corretamente e dos diferentes parâmetros que os humanos utilizam para indicar se uma melodia oferece boa ou má qualidade. Sobre todos esses aspectos e todas as habilidades de compositores musicais humanos, Griffith e Todd [2001] discutem as dificuldades e pesquisas realizadas para sistemas de composição artificial.

Sano e Jenkins [1989] propuseram um modelo de rede neural para examinar os estímulos sensitivos na percepção da altura com ênfase na sua representação neural. A tarefa da rede neural proposta é determinar qual a altura e oitava dessa altura através de simulações dos estímulos da membrana basilar.

Scarborough, Miller e Jones [1989] propuseram uma rede neural para análise de tonalidade em uma melodia. A melodia geralmente é escrita em uma tonalidade, e essa tonalidade define uma relação entre as notas e acordes presentes nela. Os humanos conseguem identificar a tonalidade de uma melodia porque

[14] A percepção de ritmo estudada por Longuet-Higgins [1979] é semelhante a percepção da tonalidade. Para identificar um ritmo de uma seqüência musical, o ouvinte precisa conhecer a batida dessa peça. Se o ouvinte pode identificar duas notas que ocorrem em sucessivas batidas, será possível descobrir quando a próxima batida será realizada. Caso contrário, se outra nota é soada naquela batida, será necessário identificar o tempo dessa nota no começo da próxima batida e atualizar, se necessário, sua estimativa de tempo.

conseguem capturar essa relação. Portanto, a rede neural busca identificar a tonalidade ao obter a relação entre notas e acordes de uma tonalidade, através da ocorrência, disposição e duração dessas notas na melodia.

Bharucha e Todd [1989] sugerem que pessoas que não possuem conhecimento sobre a estrutura de uma melodia são capazes de criar expectativas e ter intuições sobre melodias de sua cultura quando são expostas a vários exemplos dessas melodias. Os autores propuseram um modelo de rede neural que buscasse imitar essas expectativas através das tonalidades de uma cultura e que realizasse predições de acordes dessa tonalidade durante uma melodia, como acontece com humanos.

1.6 – CONSIDERACÕES FINAIS

Esse capítulo apresentou as propriedades do som e o movimento harmônico simples. Também foi apresentada uma breve discussão sobre a Série de Fourier para sons periódicos e sobre o sistema auditivo do ser humano.

Verificou-se que a nota musical possui quatro principais atributos: freqüência, duração, dinâmica e timbre (ou seja, a fonte sonora). Esses atributos estão relacionados com os parâmetros perceptivos da audição. Portanto, tem-se que uma música abrange uma estrutura de notas caracterizada por um cuidadoso controle de seus atributos. Nesse sentido, composição musical é desenhada em uma partitura por símbolos que representam arranjos de notas e os músicos aprendem a interpretar essa partitura relacionando esses arranjos. Miranda [2001] em seu ponto de vista diz que essa representação simbólica na partitura não constitui a música em si, mas sim oferece instruções para os músicos realizarem as ações necessárias para que esses símbolos sejam transformados em música.

O trabalho propõe uma abordagem para composições musicais realizadas por computadores, com o auxílio de redes neurais artificiais. Para tanto, no próximo capítulo serão apresentados conceitos iniciais de redes neurais artificiais e algumas abordagens já existentes para a aplicação de computação musical.

CAPÍTULO 2 - REDES NEURAIS ARTIFICIAIS

2.1 – CONSIDERAÇÕES INICIAIS

As redes neurais artificiais (RNAs), também conhecidas como sistemas conexionistas, constituem uma forma de computação não-algorítmica inspirada na estrutura e processamento do cérebro humano. Por não serem baseadas em regras ou programas, as redes neurais oferecem uma alternativa à computação algorítmica convencional.

As RNAs são sistemas de processamento paralelo e distribuído compostos por unidades de processamento simples (neurônios) que calculam determinadas funções matemáticas e são capazes de armazenar o conhecimento adquirido e torná-lo disponível para uso. Essas unidades de processamento são dispostas em uma ou mais camadas e interligadas por conexões sinápticas associadas a pesos que são utilizados para ponderar a entrada recebida de cada neurônio da rede e armazenar o conhecimento adquirido. [HAYKIN, 2001] [BRAGA, LUDEMIR, CARVALHO, 2000]

Para que um problema seja resolvido, as RNAs passam por um processo de aprendizagem que geralmente consiste em apresentar à rede um conjunto de exemplos para que ela consiga extrair desses exemplos características necessárias para representar a solução desejada. Geralmente, cada exemplo consiste em uma entrada para a rede e uma correspondente resposta desejada. No aprendizado, os pesos são ajustados adequadamente para representar essa solução. Uma vez treinada, a rede passa para a fase de aplicação propriamente dita, na função para a qual ela foi destinada, como classificação de padrões, imagens, etc.

As RNAs, portanto, são capazes de aprender através de exemplos e generalizar o conhecimento adquirido. A generalização ocorre quando a rede consegue produzir saídas adequadas para entradas que não pertençam ao conjunto de treinamento.

A organização desse capítulo é a seguinte: a sessão 2.2 apresenta a base biológica para o entendimento das redes neurais; a sessão 2.3 apresenta as principais arquiteturas de redes; a sessão 2.4 apresenta os principais métodos de aprendizado; a sessão 2.5 apresenta a rede neural LSTM; e a sessão 2.6 apresenta as considerações finais desse capítulo.

2.2 – BASE BIOLÓGICA

O cérebro humano possui grande habilidade em manipular problemas sem a necessidade que regras sejam explicitamente formuladas, mas sim através de exemplos. O cérebro consegue reconhecer padrões e relacioná-los, armazenar o conhecimento adquirido e utilizá-lo quando necessário, desenvolver a percepção, entre outros [HAYKIN, 2001] [BRAGA, LUDEMIR, CARVALHO, 2000].

O processamento da informação é realizado através de unidades de processamento, os neurônios. Cada neurônio recebe sinais de vários outros neurônios através das conexões sinápticas, combina essas entradas e envia outros sinais a vários outros neurônios. Portanto, a capacidade das sinapses serem moduladas é a principal base para todos os processos cognitivos, como percepção, raciocínio e memória.

A estrutura individual de cada neurônio, a topologia das conexões sinápticas e o comportamento conjunto desses neurônios formam a base de estudos em Redes Neurais Artificiais.

Uma representação básica de um neurônio biológico está ilustrada na Figura 2.1. Um neurônio é dividido praticamente em corpo da célula, dendritos e axônio. Os dendritos possuem a função de receber as informações, os estímulos nervosos, transmitidos por outros neurônios e conduzi-los até o corpo celular. O corpo celular, também conhecido como soma, coleta a informação recebida dos dendritos e gera novos impulsos. Estes impulsos são transmitidos para outros neurônios através do axônio.

O ponto de contato entre a terminação axônica de um neurônio e o dendrito de outro é conhecido como sinapse. As sinapses unem os neurônios, formando redes neurais. As sinapses são capazes de controlar a transmissão de impulsos entre os neurônios da rede. Os impulsos que chegam ao neurônio através dos dendritos são “somados” no corpo celular, e caso a soma seja maior do que um determinado valor limiar, o neurônio é ativado e dispara um impulso que caminha pelo axônio até a sinapse, para transmitir o sinal a outro neurônio.

O neurônio que envia um impulso recebe o nome de neurônio pré-sináptico e o neurônio receptor do impulso recebe o nome de neurônio pós-sináptico.

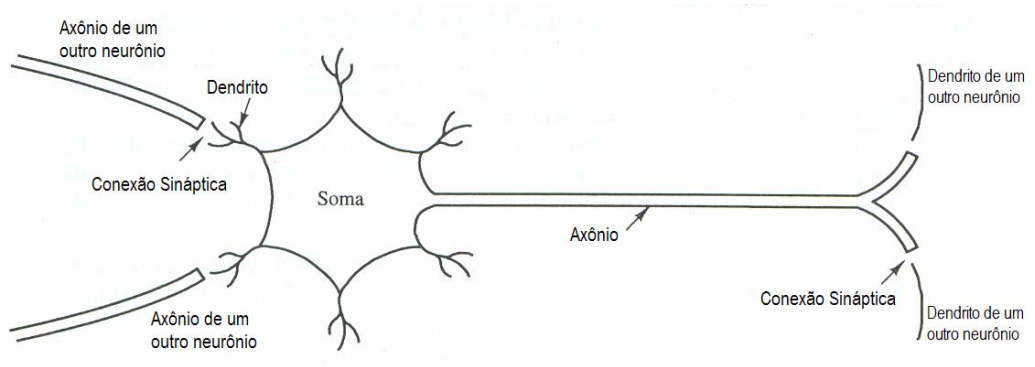


Figura 2.1: Partes simplificadas de um neurônio biológico[15]

O modelo de neurônio MCP proposto por McCulloch e Pitts [1943] é uma simplificação do neurônio biológico e está ilustrado na Figura 2.2. Os dendritos são representados pelos *n* terminais de entrada $x_1, x_2, ..., x_n$ e o axônio é representado pela saída y. A dinâmica das sinapses é simulada pelos pesos $w_1, w_2, ..., w_n$ associados aos terminais de entrada. Os pesos podem ser positivos (excitatórios) ou negativos (inibitórios). O efeito de uma sinapse particular i no neurônio pós-sináptico é dado por $x_i w_i$. O neurônio biológico dispara quando a soma dos impulsos recebidos pelo corpo celular ultrapassa o seu limiar de decisão (*threshold*). O corpo celular no MCP é simulado pela soma ponderada dos valores $x_i w_i$ recebidos e decide se o neurônio deve ou não disparar comparando a soma resultante com o limiar do neurônio.

No neurônio MCP, a ativação é obtida através de uma função de ativação degrau. Dependendo do valor resultante da soma ponderada das entradas do neurônio, essa função é responsável por ativar ou não a saída com valor 1, uma vez que o modelo MCP manipula apenas valores binários. No modelo original MCP a condição de ativação é dada pela seguinte função linear:

$$\sum_{i=1}^{n} x_i w_i \geq \theta \qquad (2.1)$$

em que n é o número de entradas do neurônio, w_i é o peso associado à entrada x_i e θ é o limiar (*threshold*) do neurônio.

[15] Disponível em: http://www.icmc.usp.br/~andre/research/neural. Acesso: 12/02/2007.

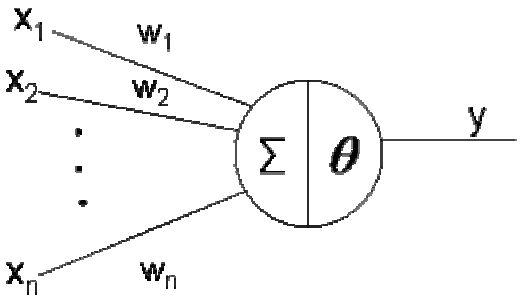


Figura 2.2: Neurônio de McCulloch e Pitts [BRAGA, LUDEMIR, CARVALHO, 2000, p.9]

A Figura 2.3 ilustra graficamente algumas das funções de ativação mais utilizadas pelas redes neurais [TANG, TAN, YI, 2007].

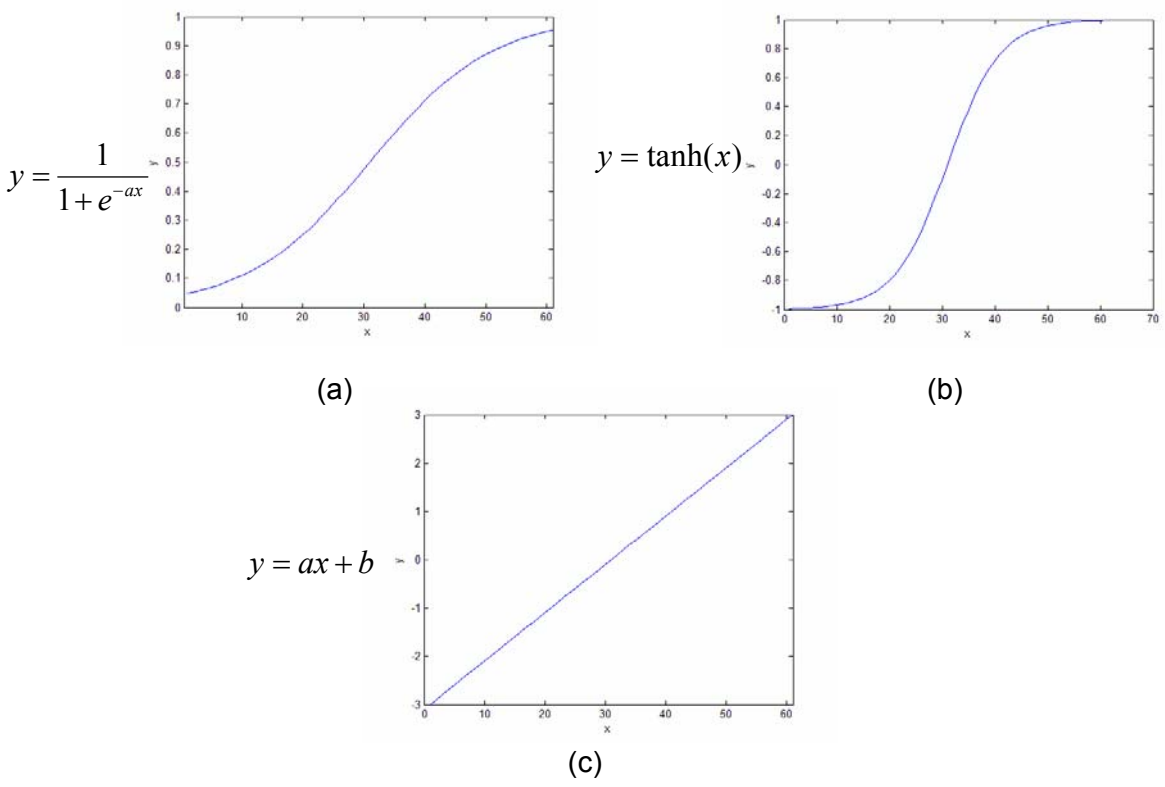


Figura 2.3: Exemplos de funções de ativação. (a) função logística (b) função tangente hiperbólica (c) função linear

Na Figura 2.3 (a) está ilustrada a função de ativação logística, que em sua forma geral, é definida por:

$$y = \varphi(x) = \frac{1}{1+\exp(-ax)} \qquad a > 0, x \in ¡ \tag{2.2}$$

O valor de saída fica no intervalo $0 \leq y \leq 1$. Sua derivada é computada como se segue:

$$\varphi'(x) = ay(1-y) \tag{2.3}$$

Na Figura 2.3 (b) tem-se a função de ativação tangente hiperbólica, definida como:

$$y = \varphi(x) = \frac{1-\exp(-2x)}{1+\exp(-2x)} \tag{2.4}$$

O valor de saída fica no intervalo $-1 \leq y \leq 1$. Sua derivada é definida como:

$$\varphi'(x) = (1+y)(1-y) \tag{2.5}$$

Por fim, na Figura 2.3 (c) tem a função de ativação linear, simplesmente definida como:

$$y = \varphi(x) = x \text{ e sua derivada é } \varphi'(x) = 1 \tag{2.6}$$

2.3 – ARQUITETURA DE REDES NEURAIS

A definição da arquitetura de uma rede neural artificial é importante, uma vez que determina qual tipo de problema a rede é capaz de resolver. Redes com uma única camada de neurônios MCP são somente capazes de resolver problemas linearmente separáveis. As redes neurais recorrentes são mais indicadas para a resolução de problemas que envolvem processamento temporal. A organização dos neurônios de uma rede neural também estabelece o algoritmo de aprendizado adequado para treiná-la [HAYKIN, 2001] [BRAGA, LUDEMIR, CARVALHO, 2000].

A arquitetura de uma rede neural pode ser definida através dos seguintes parâmetros: números de camadas, tipo de conexão entre os neurônios e topologia da rede.

Uma rede é dita ser fortemente conectada quando todos os neurônios de uma camada estão conectados com todos os neurônios da camada seguinte. Em contraste, uma rede é dita ser fracamente ou parcialmente conectada quando um neurônio de uma camada está conectado apenas aos neurônios adjacentes da camada seguinte.

Em geral, as redes neurais artificiais se classificam em três arquiteturas: (1) redes acíclicas com uma única camada; (2) redes acíclicas com múltiplas camadas; e (3) redes recorrentes ou cíclicas.

2.3.1 – Redes Acíclicas com uma Camada Única

Nessa arquitetura há uma camada de entrada de nós de alimentação e apenas uma camada de saída de neurônios computacionais. Os nós da camada de entrada correspondem aos neurônios sensoriais que possibilitam a entrada de sinais na rede (não fazem processamento). O processamento é realizado pelos neurônios MCP da camada de saída. Essa rede está ilustrada na Figura 2.4.

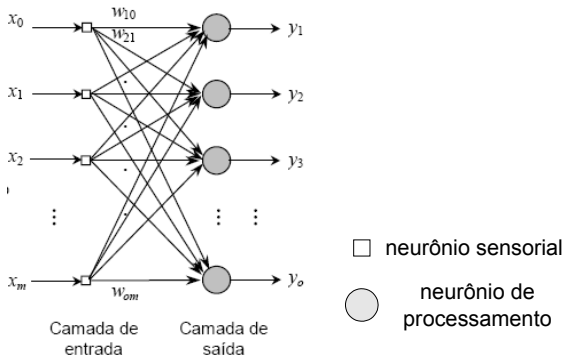


Figura 2.4: Rede acíclica com uma camada de neurônios [HAYKIN, 2001]

O **perceptron,** representado na Figura 2.5, pertence a essa categoria de rede e foi primeiramente proposto por Frank Rosenblatt, em 1957. O perceptron consiste em uma única camada de neurônios com pesos sinápticos e *bias* ajustáveis. Se os padrões de entrada forem linearmente separáveis, o algoritmo de treinamento possui convergência garantida, isto é, tem capacidade para encontrar um conjunto de pesos que classifica corretamente os dados.

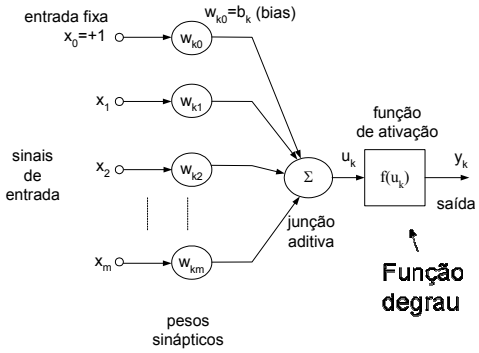


Figura 2.5: O perceptron [HAYKIN, 2001]

O perceptron da Figura 2.5 inclui um *bias* (b_k) aplicado externamente. Este *bias* tem o efeito de aumentar ou diminuir a entrada da função de ativação, quando é positivo ou negativo, respectivamente. Considere as seguintes equações, representando um determinado neurônio k:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{2.7}$$

$$y_k = \varphi(u_k + b_k) \tag{2.8}$$

em que $x_1, x_2, ..., x_m$ representam os sinais de entrada; $w_{k1}, w_{k2}, ..., w_{km}$ são os pesos sinápticos deste neurônio; u_k representa a soma ponderada dos sinais de entrada e dos pesos sinápticos; b_k é o *bias*; $\varphi(.)$ é a função de ativação; e y_k representa a saída do neurônio. O *bias* b_k permite um deslocamento à saída u_k, definida como:

$$v_k = u_k + b_k \tag{2.9}$$

Dependendo se o *bias* b_k é positivo ou negativo, a relação entre o potencial de ação v_k do neurônio k e a saída da soma ponderada u_k é modificada conforme ilustrado na Figura 2.6.

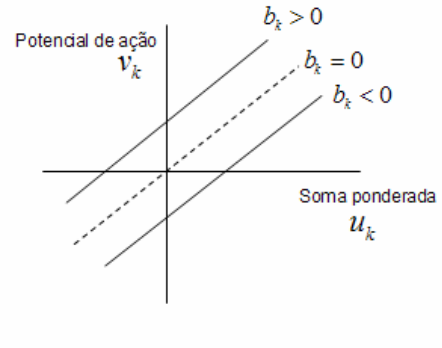


Figura 2.6: Deslocamento produzido pela presença de um *bias* [HAYKIN,2001]

2.3.2 – Redes Acíclicas com Múltiplas Camadas

Foi visto que os perceptrons de uma única camada são capazes de resolver apenas problemas linearmente separáveis. A solução de problemas não linearmente separáveis é obtida por perceptrons com uma ou mais camadas intermediárias. Camadas de neurônios que não pertencem nem a camada de entrada nem a camada de saída são camadas de neurônios internos à rede e são denominadas camadas intermediárias, ou camadas escondidas ("*hidden layers*") [HAYKIN, 2001] [BRAGA, LUDEMIR, CARVALHO, 2000].

A arquitetura de rede acíclica com múltiplas camadas se caracteriza pela presença de uma ou várias camadas escondidas. Os neurônios dessa camada são chamados de neurônios escondidos. A função dos neurônios escondidos é intervir entre a entrada externa da rede e a camada de saída. Geralmente, os neurônios de cada camada escondida da rede recebem como entradas apenas os sinais da camada precedente.

O perceptron de múltiplas camadas (***Multi-Layer Perceptron* – MLP**) é uma rede do tipo perceptron com pelo menos uma camada intermediária, onde o sinal de entrada se propaga para frente através da rede, camada por camada. Essa rede é treinada com o algoritmo de retropropagação de erro (*error back-propagation*)[16], que é baseado na regra de aprendizado por correção de erro (visto na seção 2.4).

A Figura 2.7 apresenta uma rede neural MLP totalmente conectada com duas camadas escondidas.

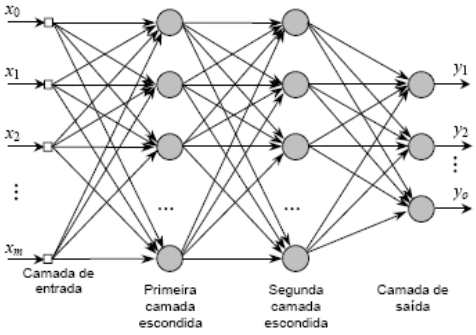


Figura 2.7: Rede Neural MLP com duas camadas escondidas [HAYKIN,2001,p.186]

Segundo Haykin [2001] a aprendizagem de uma rede MLP pode ser dividida em duas fases:

- de passos para frente, a propagação: nessa fase o padrão de atividade (vetor de entrada) é aplicado aos nós sensoriais da rede e seu efeito se propaga por toda a rede, camada por camada. Consequentemente, um conjunto de saídas é produzido como resposta da rede. Durante essa fase, os pesos sinápticos são fixos.

- de passo para trás, a retropropagação: nessa fase os pesos sinápticos são ajustados de acordo com uma regra de correção de erros. A resposta atual da rede é então comparada com a resposta desejada e a diferença dessas duas respostas produz o sinal de erro. Esse sinal é então propagado para trás através da rede, contra a direção das conexões sinápticas. Os pesos sinápticos são ajustados

[16] Também conhecido como retropropagação (*back-propagation*).

para que a resposta da rede esteja cada vez mais perto da resposta desejada. Como a propagação do erro é calculada no sentido inverso do sinal, o algoritmo é denominado de retropropagação do erro. Uma ilustração das direções dos sinais dessas duas fases pode ser observada na Figura 2.8.

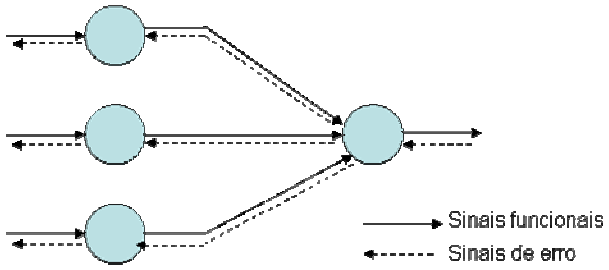


Figura 2.8: Ilustração das direções dos sinais do algoritmo de retropropagação: a propagação de sinais funcionais e a retropropagação de sinais de erro [HAYKIN, 2001, p.186]

No passo de propagação do sinal os pesos sinápticos mantêm-se inalterados, e cada sinal funcional é calculado individualmente, neurônio por neurônio. O sinal funcional da saída do neurônio *j* é dado por:

$$y_j(n) = \varphi(v_j(n)) \tag{2.10}$$

em que $v_j(n)$ é o campo de ativação ou potencial de ação do neurônio *j*, dado por:

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n) \tag{2.11}$$

em que *m* é o número total de entradas, exceto o *bias*, que são aplicadas ao neurônio *j*, e $w_{ji}(n)$ é o peso sináptico que conecta o neurônio *i* ao neurônio *j*, e $y_i(n)$ é o sinal de entrada do neurônio *j* ou o sinal funcional correspondente à saída do neurônio *i.* Se o neurônio *j* estiver situado na primeira camada oculta da rede, $m = m_0$ e o índice *i* se referir ao *i*-ésimo terminal de entrada da rede, então:

$$y_i(n) = x_i(n) \tag{2.12}$$

em que $x_i(n)$ representa o *i*-ésimo elemento do vetor de entrada. Se o neurônio *j* estiver na camada de saída da rede, e o índice *j* se referir ao *j*-ésimo terminal de saída da rede, então:

$$y_j(n) = o_j(n) \tag{2.13}$$

em que $o_j(n)$ é o *j*-ésimo elemento do vetor de saída.

A saída é comparada com a resposta desejada $d_j(n)$, e o sinal de erro $e_j(n)$ para o *j*-ésimo neurônio de saída é obtido. Dessa forma, a fase de propagação

começa na primeira camada oculta da rede, com a utilização do vetor de entrada e termina na camada de saída, em que se calcula o sinal de erro de cada neurônio dessa camada.

Ao contrário, a fase de retropropagação do erro começa na camada de saída da rede e os sinais de erro são passados para a esquerda no decorrer da rede, a cada camada, e recursivamente calculando-se o δ (gradiente local) de cada neurônio.

Se o neurônio *j* está localizado na camada de saída da rede, o sinal de erro $e_j(n)$ associado a esse neurônio pode ser calculado da seguinte forma:

$$e_j(n) = d_j(n) - y_j(n) \tag{2.14}$$

Depois de calculado $e_j(n)$, o calculo do gradiente local $\delta_j(n)$ é efetuado da seguinte forma:

$$\delta_j(n) = e_j(n)\varphi_j'(v_j(n)) \tag{2.15}$$

em que $\varphi_j'(v_j(n))$ é a derivada da função de ativação associada.

Se o neurônio *j* está localizado em uma camada oculta da rede, não é possível determinar uma resposta desejada para esse neurônio e o gradiente local $\delta_j(n)$ para o neurônio *j* é dado por:

$$\delta_j(n) = \varphi_j'(v_j(n))\sum_k \delta_k(n)w_{kj}(n) \tag{2.16}$$

A Figura 2.9 representa graficamente o fluxo do sinal de retropropagação do erro.

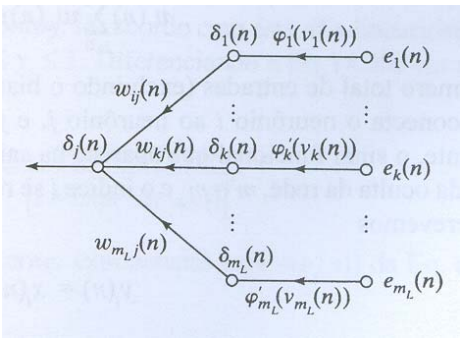


Figura 2.9: Sinal de retropropagação do erro [HAYKIN,2001,p.193]

Os pesos sinápticos são então alterados da seguinte forma:

$$\Delta w_{ji}(n) = \eta\delta_j(n)y_i(n) \tag{2.17}$$

em que η representa a taxa de aprendizagem da rede.

O aprendizado é resultado da apresentação repetitiva das amostras do conjunto de treinamento em que cada apresentação de todo o conjunto de treinamento é denominada época. O processo de aprendizagem é repetido por várias épocas e termina quando um critério de parada seja satisfeito.

2.3.3 – Redes Recorrentes ou Cíclicas

São redes que possuem pelo menos um laço de realimentação.

Uma das redes neurais recorrentes mais utilizadas é a rede de Hopfield. (Figura 2.10) na qual a resposta de rede depende sempre de seu estado no intervalo de tempo anterior [BRAGA, LUDEMIR E CARVALHO, 2000].

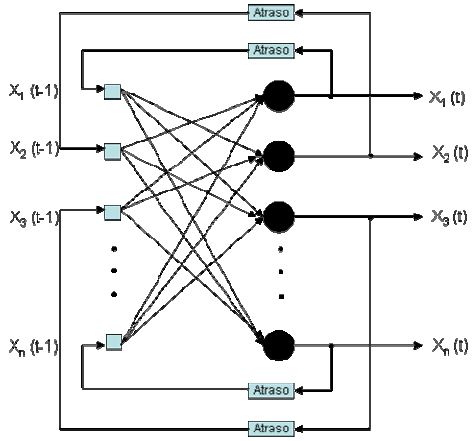


Figura 2.10: Diagrama de uma rede Hopfield [BRAGA, LUDEMIR E CARVALHO, 2000, p. 89]

O algoritmo *Back Propagation Through Time* é uma extensão do algoritmo *Back Propagation*. É utilizado para o treinamento de redes recorrentes e efetua a operação temporal de uma MLP onde a topologia da rede é acrescida de uma camada a cada instante de tempo. Na figura 2.11 está representada uma rede MLP estendida para três tempos.

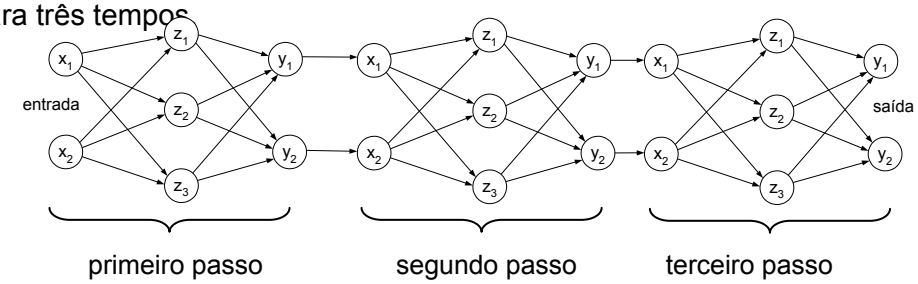


Figura 2.11: Exemplo de rede BPTT com extensão de três tempos

A Figura 2.12 ilustra graficamente uma rede BPTT para a resolução da função do senóide amortecido (Figura 2.13).

$$f(t) = \frac{\sin(\varpi t)}{\varpi t} \tag{2.18}$$

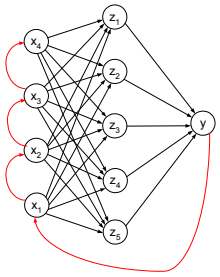


Figura 2.12: Exemplo de rede BPTT para a função do senóide amortecido [FAUSETT, 1994]

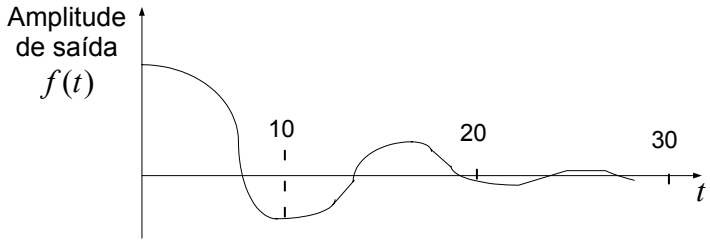


Figura 2.13: Função senóide amortecida [FAUSETT, 1994]

A entrada da rede representa valores da função em vários tempos anteriores, e a saída é o valor atual da função. No exemplo da Figura 2.13 têm-se quatro unidades de entrada e cinco unidades escondidas. O número de unidades escondidas depende da freqüência da oscilação. Para $\varpi = \pi$, sete unidades de entrada são suficientes. Para $\varpi = 0,5$, a rede pode ter de dez unidades de entrada e dez unidades escondidas. No tempo *t*, x_1 recebe o valor computado $f(t-1)$ de y; x_2 recebe o valor $f(t-2)$ de x_1; x_3 recebe $f(t-3)$ de x_2; e x_4 recebe $f(t-4)$ de x_3.

A Figura 2.14 representa o algoritmo da senóide amortecida.

- Passo 0 – inicializar os pesos (pequenos valores aleatórios)
- Passo 1 – até a condição de parada do treinamento, realizar os passos 2 a 9.
- Passo 2 – inicializar as ativações (para pequenos valores aleatórios)
- Passo 3 - apresentar o valor inicial da função, f(0) para a unidade de entrada x_1.
- Passo 4 – até a condição de parada da época, executar os passos 5 a 8.
- Passo 5 – calcular a resposta da rede y = f(1)
- Passo 6 – calcular o erro corrente. Calcular os ajustes por retropropagação, porém, não realizar os ajustes.
- Passo 7 – atualizar as ativações:
  $x_4 = x_3$
  $x_3 = x_2$
  $x_2 = x_1$
  $x_1 = y$.
- Passo 8 – testar pela condição de parada da época.
  Se y > max, ou se o número de passos > 30,
  aplicar os ajustes de pesos e continuar com o passo 9;
  senão, continuar com o passo 4.
- Passo 9 – testar a condição de parada para o treinamento.
  se (erro < tolerância) ou (número total de épocas > limite)
  parar
  senão
  continuar com o passo 1.

Figura 2.14: Algoritmo do Senóide Amortecido [FAUSETT, 1994]

2.4 – APRENDIZADO

O processo de aprendizagem de uma rede neural consiste em estimular a rede para um determinado ambiente[17], e ajustar iterativamente seus parâmetros como resultado dessa estimulação. Então a rede responde de uma maneira diferente ao ambiente, devido às mudanças ocorridas na sua estrutura interna [HAYKIN, 2001]. O aprendizado da rede pode ser supervisionado, não-supervisionado ou por reforço.

2.4.1 – Aprendizado Supervisionado

No aprendizado supervisionado, a entrada e as saídas desejadas da rede são conhecidas e o objetivo é ajustar os parâmetros da rede para aprender a relação entre os pares de entrada e saída fornecidos. O aprendizado supervisionado é também conhecido como aprendizado com professor, que conhece o ambiente e fornece o conjunto de exemplos entrada-saída desejada. O aprendizado é feito utilizando a regra de aprendizagem por correção de erro. Uma ilustração gráfica do

[17] É o ambiente que define o uso da rede. Por exemplo, se o objetivo for reconhecer caracteres, o ambiente representará todos os caracteres que podem ser apresentados à rede. [AZEVEDO,BRASIL,OLIVEIRA,2000]

aprendizado supervisionado pode ser observada na Figura 2.15 [HAYKIN, 2001] [BRAGA, LUDEMIR, CARVALHO, 2000].

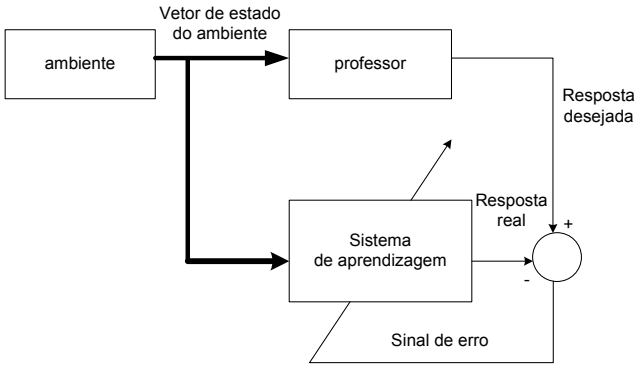


Figura 2.15: Diagrama em blocos do aprendizado supervisionado [HAYKIN, 2001, p. 88]

2.4.1.1 – Regra de Aprendizado por Correção de Erros

A regra de aprendizado por correção de erros procura minimizar a diferença entre a soma ponderada das entradas pelos pesos (saída calculada pela rede) e a saída desejada. Uma ilustração dessa aprendizagem está representada na Figura 2.16 [HAYKIN, 2001] [BRAGA, LUDEMIR, CARVALHO, 2000].

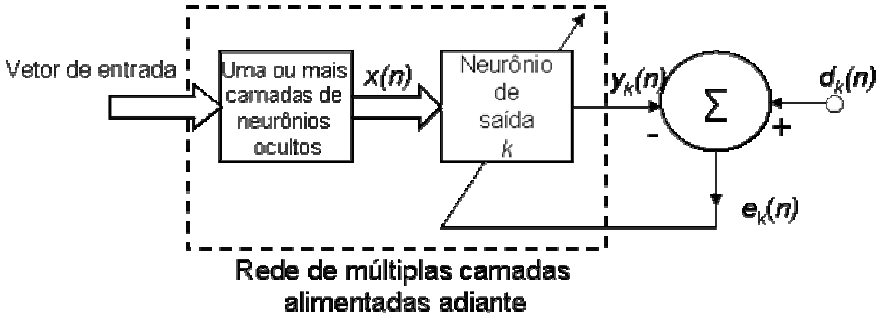


Figura 2.16: Aprendizagem por correção de erros [HAYKIN, 2001, p. 77]

Um vetor de entrada é aplicado aos nós de fonte da camada de entrada da rede neural que, por sua vez, acionam uma ou mais camadas de neurônios ocultos. As camadas de neurônios ocultos acionam o neurônio da camada de saída k através de um vetor de sinal $x(n)$. O argumento n representa o instante de tempo. O sinal de saída do neurônio k é representado por $y_k(n)$. Esse sinal de saída, que representa a

única saída da rede neural, é comparado com a saída desejada, representada por $d_k(n)$. O sinal de erro é representado por $e_k(n)$, de tal forma que:

$$e_k(n) = d_k(n) - y_k(n) \tag{2.19}$$

O sinal de erro $e_k(n)$ aciona um mecanismo de controle para ajustar os pesos sinápticos do neurônio k. Esses ajustes contribuem para que, passo a passo, o sinal de saída $y_k(n)$ esteja próximo da saída desejada $d_k(n)$. Isso é possível através da minimização de uma função de custo ou índice de desempenho, $\varepsilon(n)$, tal que:

$$\varepsilon(n) = \frac{1}{2} e_k^{\,2}(n) \tag{2.20}$$

Assim, $\varepsilon(n)$ é o valor instantâneo da energia do erro. Os ajustes dos pesos sinápticos continuam até o sistema atingir um estado estável. A minimização da função de custo $\varepsilon(n)$ resulta na regra de aprendizagem geralmente conhecida como *regra delta* ou *regra de Widrow-Hoff*.

Tendo $w_{kj}(n)$ como o valor do peso sináptico w_{kj} do neurônio k excitado por $x_j(n)$ do vetor de sinal $x(n)$ no passo de tempo n, o ajuste $\Delta w_{kj}(n)$ aplicado ao peso sináptico w_{kj} no passo de tempo n é dado por:

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n) \tag{2.21}$$

em que η é uma constante que representa a taxa de aprendizado.

Portanto, o valor atualizado do peso sináptico w_{kj} é determinado por:

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \tag{2.22}$$

2.4.2 – Aprendizado Não-Supervisionado

No aprendizado não-supervisionado não há um professor, ou seja, não há exemplos rotulados da função a ser aprendida pela rede. Nesse modelo, também conhecido como auto-organizado, são dadas as condições para realizar uma medida da representação que a rede deve aprender, e os parâmetros livres da rede são otimizados em relação a essa medida [HAYKIN, 2001] [BRAGA, LUDEMIR, CARVALHO, 2000].

Durante o treinamento, a RNA recebe diferentes padrões de entrada e os organiza em categorias. Ao realizar sua tarefa, a RNA fornece uma resposta

indicando em qual classe a entrada pertence. Se uma determinada classe não puder ser encontrada para aquele padrão de entrada, uma nova classe é gerada [AZEVEDO, BRASIL, OLIVEIRA, 2000]. O aprendizado não-supervisionado pode ser realizado utilizando a regra de aprendizagem competitiva, ou a aprendizagem Hebbiana, descritas nas subseções seguintes. Uma ilustração gráfica do aprendizado não-supervisionado pode ser observada na Figura 2.17.

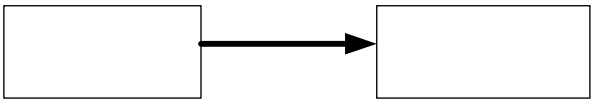


Figura 2.17: Diagrama em blocos do aprendizado não-supervisionado [HAYKIN, 2001, p. 91]

2.4.2.1 – Regra de Aprendizagem Hebbiana

Em termos matemáticos a regra de Hebb (ou aprendizagem Hebbiana) consiste na seguinte equação:

$$\Delta w_{ij}(n) = \eta y_k(n) x_j(n) \tag{2.23}$$

que significa que a mudança do peso sináptico $\Delta w_{ij}(n)$ é proporcional ao valor do neurônio pós-sináptico $y_k(n)$ e ao valor do neurônio pré-sináptico $x_j(n)$ multiplicado pelo fator de aprendizado positivo η. Isso significa que existe reforço quando há coincidência na ativação de ambos os neurônios.

2.4.2.2 – Regra de Aprendizagem Competitiva

A idéia dessa aprendizagem é, dado um vetor de entrada, fazer com que os neurônios de saída disputem entre si para serem ativados. Somente um único neurônio de saída fica ativo num determinado instante. O neurônio que vence a competição é denominado neurônio vencedor (*winner-takes-all*, em inglês).

2.4.3 – Aprendizado por Esforço

O aprendizado por reforço é considerado um caso particular de aprendizado supervisionado. A principal diferença entre o aprendizado supervisionado

clássico e o aprendizado por esforço é a medida de desempenho utilizada por cada uma das redes. No aprendizado supervisionado clássico, a medida de desempenho é baseada no conjunto de respostas desejadas de acordo com algum critério conhecido; no aprendizado por reforço o desempenho é baseado em qualquer medida que possa ser fornecida ao sistema. Assim, a única informação fornecida para a rede é se uma determinada saída está correta ou não, ou seja, não é fornecida para a rede a resposta correta para o padrão de entrada [HAYKIN, 2001] [BRAGA, LUDEMIR, CARVALHO, 2000]. Uma ilustração gráfica do aprendizado por esforço pode ser observada na Figura 2.18.

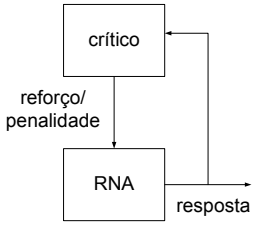


Figura 2.18: Diagrama em blocos do aprendizado por esforço [HAYKIN, 2001]

2.5 – REDE NEURAL LSTM

A rede neural *Long-Short Term Memory* (LSTM) é uma abordagem nova de redes neurais, primeiramente proposta por [HOCHREITER, SCHMIDHUBER,1997] e que desempenha um algoritmo apropriado de aprendizado baseado no gradiente [GERS, SCHMIDHUBER,2000] [GAVES, SCHMIDHUBER,2005] [PÉREZ, GERS, SCHMIDHUBER, 2003] [SCHMIDHUBER, WIERSTRA, GAGLIOLO, 2007]. Ela foi projetada para minimizar o problema do gradiente que desaparece (*vanishing gradient,* em inglês) comum nas redes recorrentes padrões [WILLIAMS, ZIPSER, 1992] [HOCHREITER, BENGIO, FRASCONI, SCHMIDHUBER, 2001]. Os primeiros métodos de aprendizado baseado no gradiente, como o BPTT, por exemplo, dividem um problema: conforme o tempo de aprendizado, como a informação do gradiente é retropropagada para atualizar os pesos que influenciarão as próximas saídas, o gradiente é continuamente diminuído pelos valores escalares das atualizações dos pesos. Portanto, em cada época de treinamento, os sinais de erros que são retropropagados dependem da magnitude dos pesos. Por essa razão, primeiras abordagens de redes neurais recorrentes falham na aprendizagem de seqüências longas de padrões de entrada e valores desejados de saída. A rede neural LSTM minimiza esse problema ao forçar um fluxo de erro constante através dos CECs (*Constant Error Carrousels*)

dentro das células de memória, permitindo com que o erro não decresça quando retropropagado. Isso melhora a capacidade de aprendizado da rede.

O bloco de memória é a unidade básica na camada escondida de uma rede neural LSTM e substitui o neurônio escondido de uma rede neural recorrente padrão (Figura 2.19).

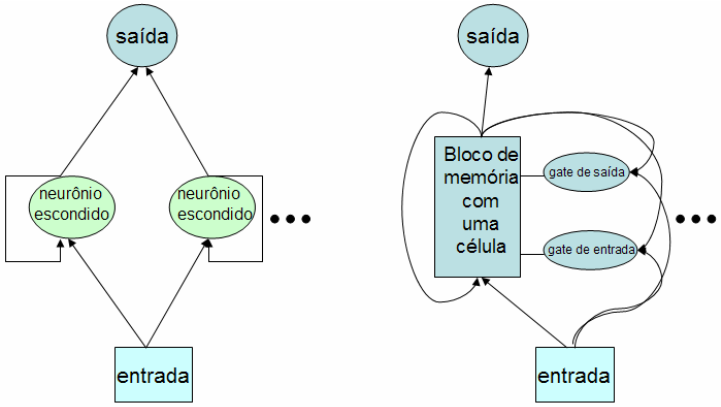


Figura 2.19: (a) Rede neural recorrente com uma camada escondida (b) Rede LSTM com blocos de memória na camada escondida [GERS, 2001, pp.11]

Um bloco de memória é formado por uma ou mais células de memória e por um par de *gates* multiplicativos, os quais computam entrada e saída para todas as células no bloco. Todas as células no bloco compartilham os mesmos *gates*. Figura 2.20 ilustra um detalhado bloco de memória com uma célula de memória.

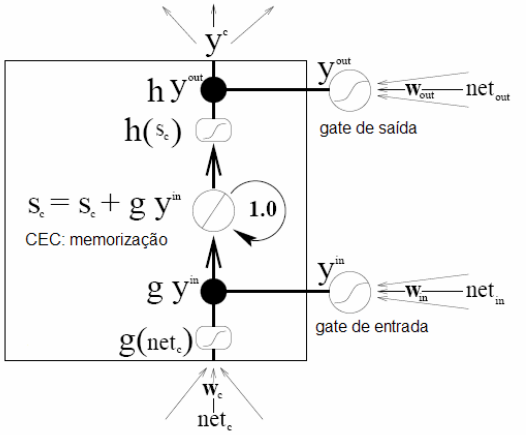


Figura 2.20: Um bloco de memória com uma única célula de memória [GERS, 2001, pp.12]

2.5.1 – Passo de propagação

Essa sessão descreve o passo de propagação da rede neural LSTM e está baseada em [GERS,2001] [HOCHREITER, SCHMIDHUBER,1997].

A atualização do estado da célula s_c é baseada em seu atual estado e nas três conexões: net_c, que representa as conexões provenientes dos padrões de entrada; net_{in} que representa as conexões do *gate* de entrada; e net_{out} que representa as conexões do *gate* de saída.

Foram considerados passos de treinamento discretos ($t = 1, 2, ...$). Um passo de treinamento envolve computar os valores de cada neurônio (passo de propagação) e calcular o sinal de erro para atualização dos pesos (retropropagação). Ainda, j indexa os blocos de memória, v indexa a célula de memória no bloco j, de tal forma que c_j^v representa a v-ésima célula do j-ésimo bloco de memória; w_{lm} representa o peso na conexão do neurônio m para o neurônio l.

A ativação dos *gates* de entrada $y^{in_j}(t)$ e saída $y^{out_j}(t)$ são funções sigmóides sobre a soma ponderada das entradas $net_{in_j}(t)$ e $net_{out_j}(t)$, respectivamente, que são recebidas das entradas recorrentes do bloco de memória e das entradas externas da rede, como se segue:

$$\begin{aligned} net_{in_j}(t) &= \sum_m w_{in_{jm}} y^m(t-1) + b_{in_j}; \\ y^{in_j}(t) &= f_{in_j}(net_{in_j}(t)), \end{aligned} \tag{2.24}$$

$$\begin{aligned} net_{out_j}(t) &= \sum_m w_{out_{jm}} y^m(t-1) + b_{out_j}; \\ y^{out_j}(t) &= f_{out_j}(net_{out_j}(t)), \end{aligned} \tag{2.25}$$

onde b_{in_j} e b_{out_j} representam os *bias* dos *gates*.

Os *gates* usam função sigmóide logística f (no intervalo [0,1]), tal que:

$$f(x) = \frac{1}{1+e^{-x}} \tag{2.26}$$

Para as células c_j^v, as entradas são multiplicadas por pesos $w_{c_j^v m}(t)$ (da entrada m para a célula) c_j^v como se segue:

$$net_{c_j^v}(t) = \sum_m w_{c_j^v m} y^m(t-1), \tag{2.27}$$

$net_{c_j^v}(t)$ é aplicada a uma função sigmóide g, com intervalo [-2,2]:

$$g(x) = \frac{4}{1+e^{-x}} - 2 \tag{2.28}$$

O estado interno da célula de memória c_j^v é:

$$
\begin{aligned}
& s_{c_j^v}(0) = 0; \\
& s_{c_j^v}(t) = s_{c_j^v}(t-1) + y^{in_j}(t) g(net_{c_j^v}(t))
\end{aligned}
\tag{2.29}
$$

para $t > 0$. A saída da célula $y^{c_j^v}$ é então:

$$
y^{c_j^v}(t) = y^{out_j}(t) h(s_{c_j^v}(t)),
\tag{2.30}
$$

em que h é uma função sigmóide com intervalo [-1,1]:

$$
h(x) = \frac{2}{1+e^{-x}} - 1
\tag{2.31}
$$

Para concluir o passo de propagação, considera-se uma rede neural com uma camada de entrada padrão, uma camada escondida consistindo de blocos de memória, e uma camada de saída padrão. A saída da rede $y^k(t)$ é a soma ponderada $net_k(t)$ passada por uma sigmóide f como segue:

$$
\begin{aligned}
& net_k(t) = \sum_m w_{km} y^m(t-1) \\
& y^k(t) = f_k(net_k(t)),
\end{aligned}
\tag{2.32}
$$

em que m representa todos os neurônios que alimentam os neurônios de saída (geralmente todas as células dos neurônios escondidos e os neurônios de entrada).

2.5.2 – Passo de retropropagação

O passo de retropropagação é iniciado com a definição de uma função objetiva, no caso da rede LSTM, o erro quadrático:

$$
E(t) = \frac{1}{2} \sum_k e_k(t)^2
\tag{2.33}
$$

em que $e_k(t) = t^k(t) - y^k(t)$ denota a diferença entre a saída obtida pela rede e a saída desejada. E é minimizado via gradiente descendente adicionando alterações Δw_{lm} aos pesos w_{lm} (do neurônio l para o neurônio m) usando taxa de aprendizado α e o delta de Kronecker δ_{ij}:

$$
\Delta w_{lm}(t) = \alpha \delta_l(t) y^m(t-1)
\tag{2.34}
$$

Para $l = k$, obtêm-se:

$$\delta_k(t) = f_k^{'}\left(net_k(t)\right)e_k(t) \tag{2.35}$$

Similarmente, as mudanças para os pesos dos *gates* de saída $\Delta w_{out_{jm}}$ são obtidos da seguinte forma:

$$\delta_{out_j}(t) = f_{out_j}^{'}\left(net_{out_j}(t)\right)\left(\sum_{v=1}^{S_j} h\left(s_{c_j}^{v}(t)\right)\sum_k w_{kc_j^v}\delta_k(t)\right) \tag{2.36}$$

Para os pesos que alimentam as células de memória, a equação para atualização dos pesos é:

$$\Delta w_{c_j^v m}(t) = \alpha e_{s_{c_j^v}}(t)\frac{\partial s_{c_j^v}(t)}{\partial w_{c_j^v m}} \tag{2.37}$$

em que $e_{s_{c_j^v}}$ é definido como erro interno, tal que:

$$e_{s_{c_j^v}}(t) = y^{out_j}(t)h'\left(s_{c_j^v}(t)\right)\left(\sum_k w_{kc_j^v}\delta_k(t)\right) \tag{2.38}$$

Para ($l = c_j^v$) e ($l = in$) tem-se:

$$\frac{\partial s_{c_j^v}(t)}{\partial w_{c_j^v m}} = \frac{\partial s_{c_j^v}(t-1)}{\partial w_{c_j^v m}} + g'\left(net_{c_j^v}(t)\right)y^{in_j}(t)y^m(t-1) \tag{2.39}$$

$$\frac{\partial s_{c_j^v}(t)}{\partial w_{in_j m}} = \frac{\partial s_{c_j^v}(t-1)}{\partial w_{in_j m}} + g\left(net_{c_j^v}(t)\right)f_{in_j}^{'}\left(net_{in_j}\right)y^m(t-1) \tag{2.40}$$

O estado inicial da rede não depende dos pesos, portanto:

$$\frac{\partial s_{c_j^v}(t=0)}{\partial w_{lm}} = 0, \qquad \text{para} \quad l \in \left\{in, c_j^v\right\} \tag{2.41}$$

Finalmente, para atualizar os pesos do *gate* de entrada, é necessário somar as contribuições de todas as células no bloco:

$$\Delta w_{lm}(t) = \alpha\sum_{v=1}^{S_j} e_{s_{c_j^v}}(t)\frac{\partial s_{c_j^v}(t)}{\partial w_{lm}}, \quad \text{para} \quad l = in \tag{2.42}$$

2.6 – CONSIDERACÕES FINAIS

O estudo das Redes Neurais Artificiais é motivado pelo funcionamento dos neurônios biológicos em sistemas nervosos. Apesar de atualmente as RNAs estarem distantes das Redes Neurais Naturais (RNNs), elas oferecem soluções para problemas que a forma de computação algorítmica possui dificuldade para resolver. A topologia da rede e o algoritmo de aprendizado utilizado determinam como a rede neural irá obter a solução desejada para o problema.

As redes LSTM e BPTT são utilizadas nesse trabalho para a aplicação de composições musicais.

No próximo capítulo, serão discutidas as abordagens encontradas na literatura sobre composição musical usando computadores, incluindo redes neurais.

CAPÍTULO 3 – ABORDAGENS SOBRE COMPOSIÇÃO MUSICAL USANDO COMPUTADORES

3.1 – CONSIDERAÇÕES INICIAIS

A computação musical, incluindo reprodução e composição, tem atraído pesquisadores há muito tempo. Composições musicais por computadores, mais especificamente, datam da década de 50, quando as cadeias de Markov foram utilizadas para compor melodias. Uma vez que estudantes de música geralmente aprendem a compor através de exemplos, abordagens iniciais foram motivadas e baseadas em análises de padrões nas melodias [TODD e LOY,1991]. Essas abordagens incluem cadeias de Markov, gramáticas e autômatas. Mais recentemente, as redes neurais artificiais foram desenvolvidas para a aprendizagem de processos musicais.

O desenvolvimento de algoritmos de aprendizado por redes neurais trouxe uma nova possibilidade para composição musical. Redes *feedforward* e redes neurais recorrentes podem ser treinadas para produzir notas sucessivas ou compassos de melodias em um conjunto de treinamento, uma vez dados antecipadamente notas ou compassos como entrada. Uma vez elas terem aprendido a reproduzir as melodias do conjunto de treinamento, essas redes neurais podem ser induzidas para compor novas melodias baseadas nos padrões que elas aprenderam. As redes neurais podem incorporar estrutura musical partindo do conjunto de treinamento. Elas permitem também a construção de estruturas futuras, como restrições motivadas psicologicamente na representação das notas com os respectivos tempos. Isso contribui para que suas saídas apresentem músicas mais apropriadas [TODD e WERNER, 1998].

Nesse capítulo algumas abordagens utilizadas e estudadas são apresentadas. Essas abordagens utilizam técnicas como probabilidades, gramáticas, redes neurais artificiais, entre outros. A organização do capítulo é a seguinte: a sessão 3.2 apresenta exemplos de abordagens para composições musicais envolvendo técnicas tradicionais; a sessão 3.3 apresenta exemplos de abordagens para composições musicais usando redes neurais; e a sessão 3.4 apresenta as considerações finais desse capítulo.

3.2 – EXEMPLOS DE ABORDAGENS PARA COMPOSIÇÕES MUSICAIS USANDO TÉCNICAS TRADICIONAIS

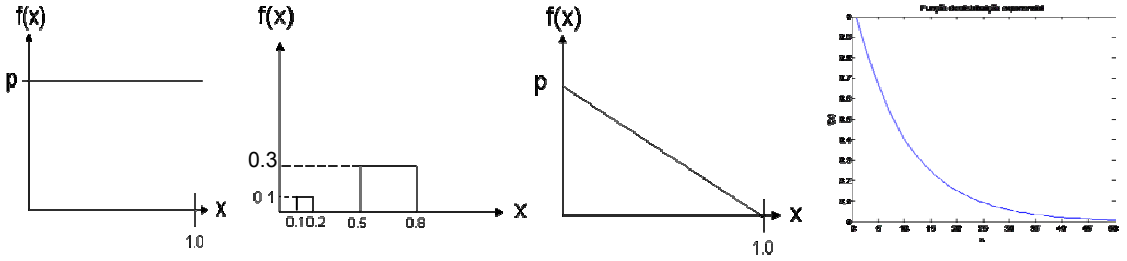
A idéia de compor música utilizando formas aleatórias se mostra atrativa para os compositores, principalmente após a invenção do computador. Porém, simplesmente utilizar processos aleatórios pode não gerar composições interessantes. Uma alternativa é adicionar a esses processos restrições de composição. Uma das formas para obter isso é através das probabilidades ou dos processos iterativos. Outras abordagens constituem a utilização de gramáticas formais ou de autômatos finitos, que possuem um conjunto de regras a serem seguidas durante o processo de composição. A seguir serão apresentadas abordagens propostas por Miranda [2001] que envolvem probabilidades, gramáticas, autômatos finitos e processos iterativos.

3.2.1 – Probabilidades

Em música, probabilidades são normalmente usadas para gerar seqüências musicais selecionando elementos de um conjunto. Para gerar uma seqüência musical a partir de um conjunto de notas, o computador pode ser programado para selecionar aleatoriamente uma nota por vez e tocá-la através de um sintetizador. Essa seleção de notas pode ser através da probabilidade justa ou da probabilidade condicional. A probabilidade justa ocorre quando as chances de uma possível escolha são iguais para todas as escolhas do conjunto, ou seja, não há favoritismo e nem informação sobre o passado das escolhas anteriores. A probabilidade condicional ocorre quando a chance de uma possível escolha depende das informações do passado das escolhas anteriores. Se não há notas repetidas no conjunto de notas, então as escolhas serão justas. Se tiver uma ou mais notas repetidas no conjunto, a chance dessa nota em particular ser selecionada será maior e aumenta proporcionalmente com o número de repetições dessa nota no conjunto. As funções de distribuição são utilizadas como ferramentas de seleção baseadas em probabilidade. Nesse contexto, um gerador estocástico representa um sistema que gera elementos musicais selecionando-os de um dado conjunto de acordo com alguma função de distribuição. Basicamente, há quatro classes de funções de distribuição: uniforme, linear, exponencial, côncava e convexa.

A função de distribuição uniforme é a mais simples e está ilustrada na Figura 3.1. A probabilidade de uma escolha é igual para todas as escolhas (*fair trial* – *triagem justa*). No gráfico da Figura 3.1 (a) todas as escolhas estarão em zero e um, com igual probabilidade entre elas. A probabilidade também é expressa pela

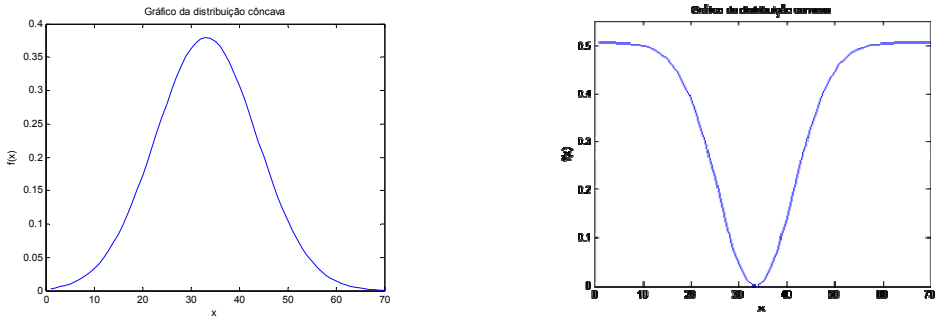
probabilidade de uma escolha cair em uma região de possíveis escolhas e é dada pela área delimitada pelo valor da linha horizontal, conhecida como curva. Na Figura 3.1 (b) a probabilidade de uma escolha estar em 0,1 e 0,2 é 10% e a probabilidade de uma escolha estar entre 0,5 e 0,8 é 30%. Na função de distribuição linear (Figura 3.1 (c)), as chances de uma escolha são maiores para valores menores. A função de distribuição exponencial (Figura 3.1 (d)) também favorece os valores menores. A diferença é a presença de um parâmetro λ que define a curva desse favoritismo. Quanto maior o valor de λ, maior será a probabilidade dos valores menores.



(a) (b) (c) (d)

Figura 3.1: (a) distribuição constante (b) distribuição constante por intervalo (c) distribuição linear decrescente (d) distribuição exponencial

Na função de distribuição côncava os elementos com maiores probabilidades de escolha estão no centro na função, ou seja, a distribuição representa uma função exponencial bilateral (Figura 3.2 (a)). O parâmetro λ nesse caso determina a largura da curva. As distribuições côncavas podem ser simétricas ou assimétricas. Na função de distribuição convexa, ao contrário da distribuição côncava, os maiores valores possuem maiores probabilidades de escolha. Também pode ser simétrica ou assimétrica. Está graficamente representada na Figura 3.2 (b).



(a) (b)

Figura 3.2: (a) Função de distribuição côncava (b) Função de distribuição convexa

Uma função de distribuição pode ser implementada como uma tabela de probabilidade que estabelece valores que representam a probabilidade de ocorrência de um ou mais eventos. Em um sistema de composição musical essas

tabelas de probabilidades podem contribuir nas rotinas de tomada de decisão. Por exemplo, a distribuição uniforme pode ser utilizada para escolher um dos *x* eventos distribuídos uniformemente ao selecionar um número entre *0* e *x-1* que será utilizado para chamar a rotina. Um exemplo dessa abordagem está ilustrado na Figura 3.3, em que $V[n]$ representa um vetor de notas.

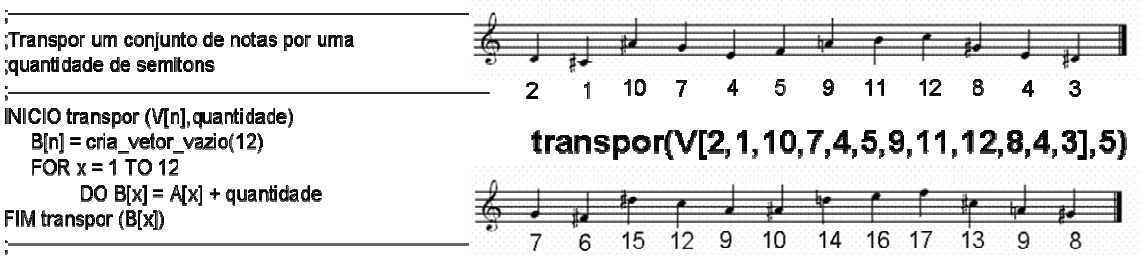
A rotina da Figura 3.3 recebe como entrada um vetor de notas, $V[n]$. Então, aleatoriamente seleciona uma das quatro operações a serem realizadas nessas notas e retorna o resultado no vetor de notas $B[n]$. Outras funções de distribuição podem ser utilizadas simplesmente alterando a forma com que os números são gerados.

As rotinas presentes no exemplo da Figura 3.3, tais como transpor e retroceder estão ilustradas nas Figuras 3.4 e 3.5. Na Figura 3.4 (a) está representada uma rotina que transpõe uma seqüência de notas, $V[n]$, para uma quantidade de semitons, como no exemplo da Figura 3.4 (b). Na Figura 3.5 (a) a seqüência de notas do vetor $V[n]$ é retrocedida, como no exemplo da Figura 3.5 (b).

```
;_____________________________________________
; Seleciona um das quatro operações e ; aplica a um dado
conjunto de notas
;_____________________________________________
INICIO tabela_uniforme(V[n])
    B[n] = cria_vetor_vazio(12)
    rand = random(2)
    CASO
          rand = 0 ENTAO B[n] = transpor (V[n],5)
          rand = 1 ENTAO B[n] = transpor (V[n],7)
          rand = 2 ENTAO B[n] = retroceder (V[n])

FIM tabela_uniforme (B[n])
;
_____________________________________________
```

Figura 3.3: Exemplo de tabela de probabilidade [Miranda, 2001]



(a) (b)

Figura 3.4: Transposição de notas. (a) Exemplo de rotina de transposição (b) Exemplo de transposição. [Miranda, 2001]

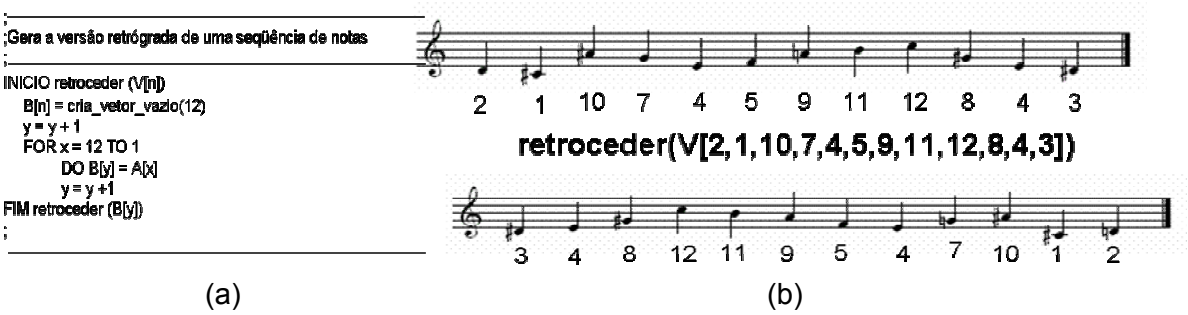


Figura 3.5: Retroação de notas. (a) Exemplo de rotina de retroação (b) Exemplo de retroação. [Miranda, 2001]

3.2.2 – Cadeias de Markov

Mozer [1994] comenta sobre a possibilidade de criar uma composição musical a partir de notas selecionadas seqüencialmente, de acordo com alguma tabela de transição que determina a probabilidade da próxima nota em função da ocorrência da nota anterior, ou até mesmo da anterior da anterior.

Cadeias de Markov são sistemas de probabilidades condicionais em que a probabilidade da ocorrência de eventos futuros depende de um ou mais eventos passados. O número de eventos passados (no caso, notas geradas anteriormente) que são levados em consideração constitui a ordem da tabela. Uma tabela de transição que leva em consideração *n* notas passadas pode ser representada como uma matriz de *n* + 1 dimensões. A matriz de transição de estados fornece a probabilidade de ocorrência de um evento dados os n estados anteriores. Para ilustrar a geração de seqüência de notas utilizando cadeias de Markov, serão utilizadas as notas da seqüência musical escrita na clave de sol da Figura 3.6, que representa a escala de C maior (Dó maior) na quarta oitava.

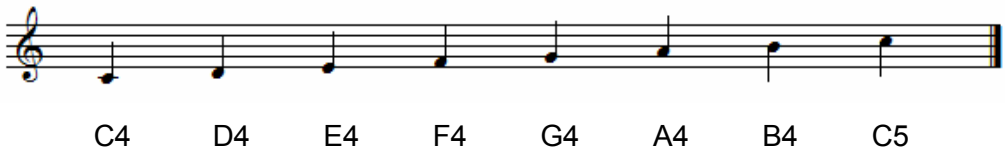


Figura 3.6: Escala de Dó Maior na quarta oitava

Considerando também as seguintes regras para determinar quais notas podem suceder uma dada nota:

- Se C4, então C4, D4, E4, G4 ou C5;
- Se D4, então C4, E4 ou G4.

- Se E4, então D4 ou F4.
- Se F4, então C4, E4 ou G4.
- Se G4, então C4, F4, G4 ou A4.
- Se A4, então B4.
- Se B4, então C5.
- Se C5, então A4 ou B4.

Depois de ocorrer C4 cada uma das cinco notas C4, D4, E4, G4 ou C5 podem ocorrer com 20% de chance cada, ou seja, cada uma dessas cinco notas possui probabilidade *p* = 0,2. A probabilidade não precisa ser uniformemente distribuída. Por exemplo, depois de ocorrer D4, a nota C4 pode ter probabilidade *p* = 0,2 enquanto as notas E4 e G4 podem ter probabilidade *p* = 0,4. Essas probabilidades podem ser expressas em uma matriz de transição de estados de primeira ordem (Figura 3.7).

próximos eventos

eventos passados

	C4	D4	E4	F4	G4	A4	B4	C5
C4	0.2	0.2	0.2	0.0	0.2	0.0	0.0	0.2
D4	0.2	0.0	0.4	0.0	0.4	0.0	0.0	0.0
E4	0.0	0.5	0.0	0.5	0.0	0.0	0.0	0.0
F4	0.2	0.0	0.4	0.0	0.4	0.0	0.0	0.0
G4	0.25	0.0	0.0	0.25	0.25	0.25	0.0	0.0
A4	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
B4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
C5	0.0	0.0	0.0	0.0	0.0	0.5	0.5	0.0

Figura 3.7: Matriz de transição de estados para a escala de Dó Maior [Miranda, 2001]

Matrizes de alta ordem funcionam similarmente. Uma tabela de transição de segunda ordem deve possuir três dimensões: uma para a nota atual, uma para a nota anterior e outra para a segunda nota anterior.

Papadopoulos e Wiggins [1999] mencionam que algoritmos de composição musical que utilizam métodos de cadeias de Markov são muito utilizadas por serem simples de serem implementadas e são boas alternativas para aplicações de tempo real.

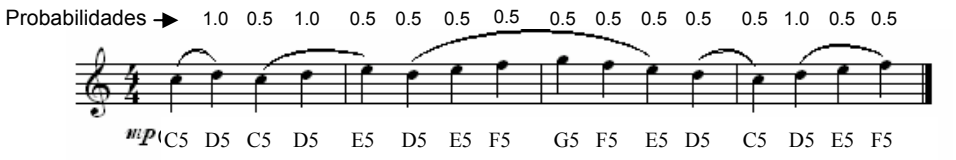
A Figura 3.8 ilustra outra tabela de transição de primeira ordem para as notas da escala C maior na quinta oitava. Nesse exemplo, a próxima nota será um passo acima ou abaixo da nota atual. [MOZER, 1994]

Nesse método se a representação em matriz consiste em entradas não-zeros imediatamente em um dos lados da diagonal principal e zeros em qualquer outro lugar, têm-se um processo de caminhada aleatória. Tabelas de transição podem ser construídas de acordo com um determinado critério, como na Figura 3.8 ou podem representar estilos musicais específicos. Nesse último caso, informações estatísticas são coletadas de um conjunto de exemplos (o conjunto de treinamento) e assim, as entradas da tabela de transição representarão a probabilidade de transição nesses exemplos.

próxima nota

nota anterior	C5	D5	E5	F5	G5	A5	B5	C6
C5	0.0	**1.0**	0.0	0.0	0.0	0.0	0.0	0.0
D5	**0.5**	0.0	**0.5**	0.0	0.0	0.0	0.0	0.0
E5	0.0	**0.5**	0.0	**0.5**	0.0	0.0	0.0	0.0
F5	0.0	0.0	**0.5**	0.0	**0.5**	0.0	0.0	0.0
G5	0.0	0.0	0.0	**0.5**	0.0	**0.5**	0.0	0.0
A5	0.0	0.0	0.0	0.0	**0.5**	0.0	**0.5**	0.0
B5	0.0	0.0	0.0	0.0	0.0	**0.5**	0.0	**0.5**
C6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**1.0**

(a)



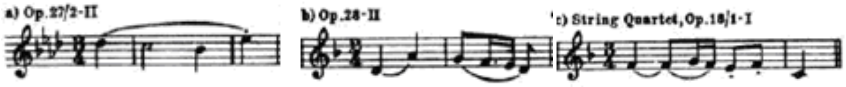
(b)

Figura 3.8: (a) Tabela de transição (b) Exemplo de seqüência musical resultante partindo da nota C5 [MIRANDA, 2001, p. 72]

3.2.3 – Gramáticas

Um pedaço de música pode ser pensado como um conjunto de estrutura hierárquica: no nível mais baixo estão as notas, as quais formam frases[18] e melodias, temas[19], etc.

[18] A frase musical é a menor unidade estrutural musical, composta por eventos musicais que se relacionam e se completam e combinam com outras unidades estruturais. SHOENBERG [1967] Como forma de ilustração, a frase musical representa o que alguém consegue soletrar ou cantar em uma única respiração. Exemplos:



Essa abordagem da música possui uma estreita similaridade com a linguagem, uma vez que as notas musicais podem ser relacionadas com os fonemas, que se transformam em palavras, frases, e assim por diante.

Um exemplo de uma representação hierárquica para uma sonata está parcialmente ilustrado na Figura 3.9. A sonata geralmente é dividida em ABA', em que têm-se a apresentação de um tema (A), desenvolve-se esse tema (B) e depois o compositor retorna ao tema inicial com alguma possível alteração (A').

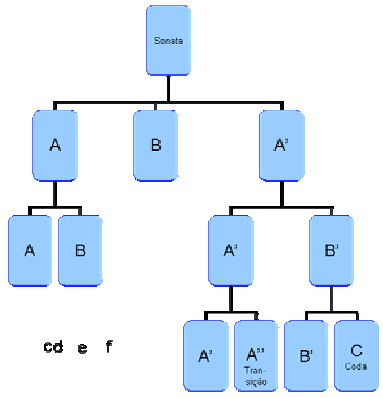


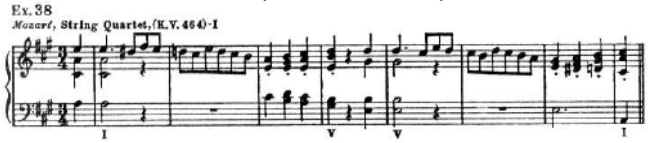
Figura 3.9: Estrutura hierárquica de uma sonata [Miranda, 2001]

As gramáticas formais foram primeiramente propostas por Noam Chomsky [1957]. Chomsky acreditava que os seres humanos conseguem se comunicar entre si através de uma linguagem, porque é possível coordenar a gramática dessa linguagem. Chomsky também acreditava que poderia existir uma gramática universal, para todas as linguagens. Compositores musicais trabalhando com computadores ficaram interessados nos trabalhos de Chomsky e passaram a tentar aplicar esses conceitos para a composição musical.

Como exemplo de gramática para composição musical, considera-se as regras da Figura 3.10 (a) e as cinco noções abaixo:

- Noção R_n para referenciar uma nota (ex: R_1 = C4).
- Noção de intervalo I_n entre duas notas (ex: I_7 = quinta perfeita).
- Noção de direção do intervalo D_n (ex: D_1 = ascendente).
- Noção de seqüência SEQ_n.

[19] O tema musical está relacionado com o período de uma seqüência. Geralmente aparecem em músicas clássicas como partes de grandes formas (por exemplo, representando o A da forma ABA de uma sonata), ou podem ser totalmente independentes. Exemplos:



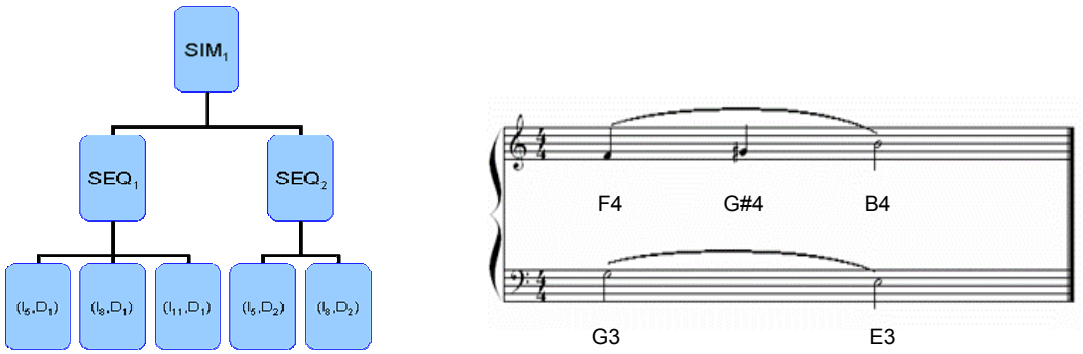
– Noção de simultaneidade SIM_n .

Portanto, têm-se os cinco vetores:

– R que representa as notas musicais da escala de C maior na quarta oitava. R = {C4, D4, E4, F4. G4. A4, B4};

– I que representa os intervalos musicais. I = {2ª menor, 2ª maior, 3ª menor, 3ª maior, 4ª perfeita, 4ª aumentada, 5ª perfeita, 6ª menor, 6ª maior, 7ª menor, 7ª maior, oitava};

– D que representa a direção dos intervalos. D = {ascendente, descendente}



(a) (b)

Figura 3.10: Exemplo de uma gramática musical. (a) Exemplo de regras. (b) Exemplo de notas geradas pelas regras. [Miranda, 2001]

O resultado da aplicação das regras da Figura 3.10 (a) pode ser observado na Figura 3.10 (b) em que uma passagem musical é composta por duas seqüências tocadas simultaneamente. A primeira seqüência é formada por três notas e a segunda seqüência é formada por duas notas. As notas da primeira seqüência são obtidas da seguinte maneira: partindo da nota de referência R_1 = C4, a primeira nota é obtida por um intervalo de quarta perfeita ascendente (F4), depois um intervalo de sexta menor ascendente (G#4), e por fim um intervalo de sétima maior ascendente (B4). As notas da segunda seqüência são obtidas através da mesma nota de referência nessa ordem: um intervalo de quarta perfeita descendente (G3) e um intervalo de sexta menor descendente (E3). A passagem musical da Figura 3.10 (b) foi editada, pois a gramática acima não lida com outros atributos das notas musicais, como por exemplo, duração. A curva acima das notas denomina-se ligadura. Todas as notas dentro da ligadura devem ser tocadas sem interrupção, ligadamente.

O exemplo da gramática acima pode ser alterado de várias maneiras e outras seqüências podem ser adicionadas para a composição de uma melodia musical.

O desafio para sistemas musicais baseados em regras é permitir com que as regras se interagem entre si para lidar com novas situações, uma vez que expectativa em novas situações são importantes para o entendimento musical [LOY,2001].

3.2.4 – Autômatos de estado finito

Os autômatos de estado finito, geralmente utilizados nas linguagens de programação, são semelhantes às gramáticas formais.

Um autômato finito contém os seguintes elementos: A = (Q, I, F, T), em que: Q é o conjunto de estados; I é um subconjunto de Q que contém os estados iniciais; F é um subconjunto de Q que contém os estados finais; e T representa as transições. Os elementos de T são formados pela combinação de dois em dois estados de Q através de um *link*. Por exemplo, (p,a,q) significa que há uma transição do estado p para o estado q através de uma ação a. As representações gráficas auxiliam no entendimento das regras. Um exemplo de autômato finito está ilustrado na Figura 3.11. É descrito por: A = ({ p, q, r }, { p }, { r }, {(p , a , p),(p , a , q), (q , b , q), (q , b , r)}).

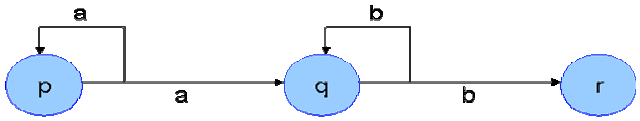


Figura 3.11: Exemplo de autômato finito com três estados [Miranda, 2001]

Para utilizar um autômato finito para composição musical, utiliza-se preencher os estados do autômato com notas ou passagens musicais curtas. A Figura 3.12 (a) apresenta um exemplo de autômato finito para composições musicais e exemplo de seqüência musical gerada por esse autômato pode ser observado na Figura 3.12 (b).

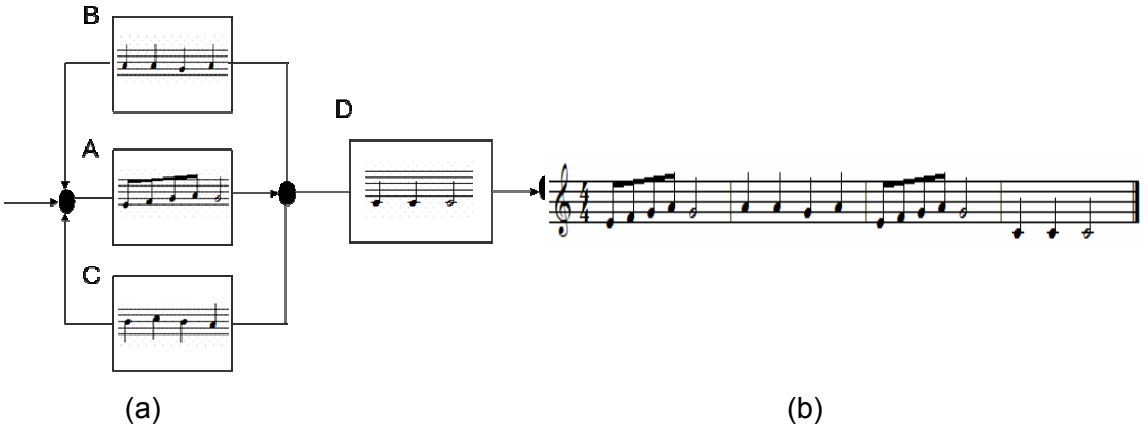


Figura 3.12: (a) Exemplo de um autômato finito para composição musical. (b) Exemplo de seqüência musical gerada pelo autômato com quatro compassos. [Miranda, 2001]

3.2.5 – Algoritmos Iterativos

Um processo iterativo é a aplicação continuada de um procedimento matemático onde cada resultado é retornado para a obtenção do próximo resultado. A Figura 3.13 ilustra graficamente os passos de um processo iterativo.

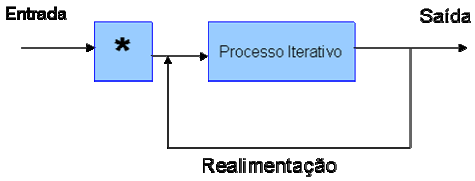


Figura 3.13: Passos de um processo Iterativo [Miranda, 2001]

Um processo interativo representa uma regra que descreve a ação a ser repetidamente aplicada a um valor inicial x_0. Os resultados de um processo iterativo constituem um conjunto, formalmente referenciado como a *órbita* do processo, e os valores pertencentes a esse conjunto são nomeados como *pontos* da órbita. A órbita O resultante da aplicação de um processo iterativo para uma regra F para um valor inicial x_0 é escrita como $O^F(x_0)$. Por exemplo, seja a regra $F: x_{n+1} = x_n + 2$. Se o valor inicial de x_0 é 0, então $O^F(0) = \{0,2,4,6,...\}$.

Um processo iterativo geralmente produz três classes de órbitas:

- Órbitas nas quais os pontos tendem a se estabilizarem em um valor fixo. Por exemplo, seja a regra $F: x_{n+1} = (x_n / 2)$.Se x_0 = 1, $O^F(1)$= {1, 0.5, 0.25, 0.125, ...}. Essa órbita tende a zero, independente do valor inicial.

- Órbitas nas quais os pontos tendem a oscilarem entre valores específicos. Por exemplo, seja a regra $F: x_{n+1} = 3.1x_n(1-x_n)$. Se x_0 = 0,5, $O^F(0.5) = \{0.5,0.775,0.540,0.770,0.549,0.768,0.553,0.766,0.765,0.557,0.765,0.557,...\}$. Depois do período inicial, a órbita cai em uma oscilação entre 0,765 e 0,557, oscila entre dois pontos.

- Órbitas nas quais não é possível distinguir explicitamente um padrão entre os pontos. Por exemplo, $F: x_{n+1} = \Delta x_n(1-x_n)$. Atribuindo Δ=4 têm-se os valores ilustrados na Figura 3.14. A Figura 3.14 (a) representa graficamente o mesmo processo iterativo para um valor inicial x_0 = 0,3. A Figura 3.14 (b) representa graficamente o mesmo processo iterativo para um valor inicial x_0 = 0,301. Observa-se que pequenas variações no valor inicial x_0 causam grandes diferenças depois de poucas iterações do processo iterativo.

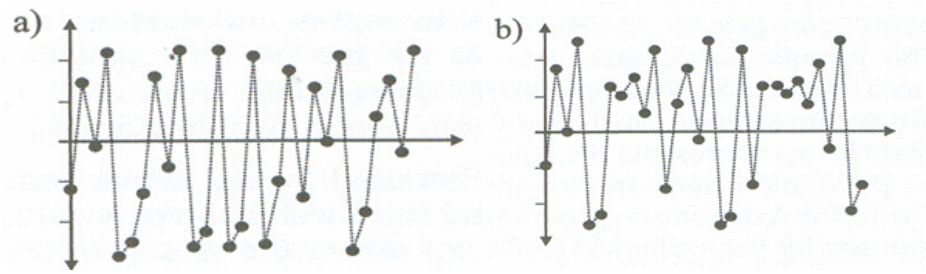


Figura 3.14: Órbita caótica. (a) Órbita gerada para o valor inicial x_0 = 0,3. (b) Órbita gerada para o valor inicial x_0 = 0,301. [Miranda, 2001]

Os seres humanos apreciam músicas que apresentam um bom balanço entre a repetição de elementos musicais e inovações dentro de uma melodia musical e da relação dessa melodia com outras melodias. Para a aplicação de composição musical, órbitas que em pouco tempo convergem para um valor estável não são apropriadas, uma vez que seu comportamento logo passa a ser estático. Órbitas oscilatórias oferecem resultados musicais interessantes, principalmente quando sua oscilação é complexa, envolvendo vários pontos, pois se a oscilação envolve poucos pontos, os elementos musicais se repetirão com grande freqüência. Órbitas caóticas são mais apropriadas para a composição de novas melodias, pois tendem a percorrer uma extensão fixa de valores, como pontos similares, mas não idênticos. Portanto, órbitas caóticas conseguem gerar materiais musicais que se relacionam entre si.

Um processo iterativo bidimensional pode permitir controle sobre dois parâmetros musicais (nota e duração, por exemplo). Uma dificuldade encontrada pelos compositores é obter um método efetivo para mapear órbitas para parâmetros

musicais, principalmente porque os processos iterativos não foram originalmente desenvolvidos com uma perspectiva musical em mente.

3.3 – EXEMPLOS DE ABORDAGENS PARA COMPOSIÇÕES MUSICAIS USANDO REDES NEURAIS

As redes neurais artificiais geralmente oferecem mecanismos de aprendizado em que o objetivo final pode ser obtido expondo a rede repetidamente a exemplos que determinam o comportamento esperado. Nesses mecanismos, as redes neurais adaptam suas interconexões até que os padrões de excitação desejados estejam pertos (do que a rede é capaz de obter) do comportamento desejado. Assim, as redes neurais são capazes de simular comportamentos complexos dificilmente de serem gerados por conjuntos de regras ou probabilidades. As redes neurais artificiais têm se mostrado apropriadas para a composição musical, pois são capazes de aprender padrões e características presentes nas melodias do conjunto de treinamento e obter generalizações dessas características para a composição de novas melodias. A seguir são apresentadas algumas abordagens que utilizam redes neurais artificiais para a composição musical.

3.3.1 – Abordagem por Todd [1989]

A abordagem estudada por Todd [1989] para uma composição algorítmica consiste na criação de uma rede neural capaz de aprender aspectos da estrutura musical através de exemplos musicais dados a ela e então ser capaz de utilizar o que aprendeu para construir novas melodias. O autor sugere que para a rede neural atingir esses requisitos ela deve ser capaz de reproduzir exatamente um conjunto de exemplos musicais, pois ser capaz de reproduzir os exemplos exige que a rede neural aprenda lidar com as estruturas musicais desses exemplos. A rede neural desenvolvida por Todd tem sido aplicada para a tarefa de composição algorítmica em que o domínio musical está restrito a melodias monofônicas.

O autor trata música como um fenômeno seqüencial em que notas ocorrem uma após a outra em seqüência. Por isso, o autor utiliza uma rede neural seqüencial que aprende a gerar uma seqüência de notas, em que a próxima nota depende da memorização de algumas notas geradas anteriormente, através de uma memória do passado provida por conexões que retornam da camada de saída para a camada de entrada. O tempo é representado pela posição da nota dentro da seqüência e a rede neural aprende a associar padrões de saída com padrões de

entradas ajustando os pesos das conexões na rede. A saída atual da rede influencia na geração da próxima saída.

A rede neural proposta por Todd [1989] está ilustrada na Figura 3.15. A rede é do tipo BPTT (*Back Propagation Through Time*). Na camada de entrada, um conjunto de neurônios adicionais, denominados neurônios de planejamento, indicam qual seqüência, entre várias possibilidades, a rede irá aprender ou produzir. Se a rede é treinada para aprender 3 melodias, os neurônios de planejamento são 001, 010 e 100 para a primeira, segunda e terceira melodia, respectivamente. Isso é feito com um conjunto fixo de ativações ligadas durante o aprendizado ou produção da seqüência.

Os neurônios de contexto formam o restante da camada de entrada. Esses neurônios mantêm uma memória da seqüência produzida anteriormente, a qual forma o contexto atual utilizado pela rede para criar a próxima nota da seqüência. Cada saída sucessiva da rede é retornada para essa memória pelas conexões que retornam da camada de saída para os neurônios de contexto. Além da memória da saída anterior, os neurônios de contexto possuem conexões para si mesmo.

Os neurônios de contexto e os neurônios de planejamento são totalmente conectados com a camada de neurônios escondidos. Os neurônios da camada escondida combinam a informação dos pesos dos neurônios de planejamento e dos neurônios de contexto e processam essa informação através de uma função logística. Essa informação processada é combinada com o conjunto final de pesos e transmitida para a camada de saída. Os neurônios de saída determinam o que a rede irá produzir como o próximo elemento da seqüência. Cada saída é então passada de volta para os neurônios de contextos através das conexões para alterar o contexto, permitindo a geração do próximo elemento da seqüência e assim por diante.

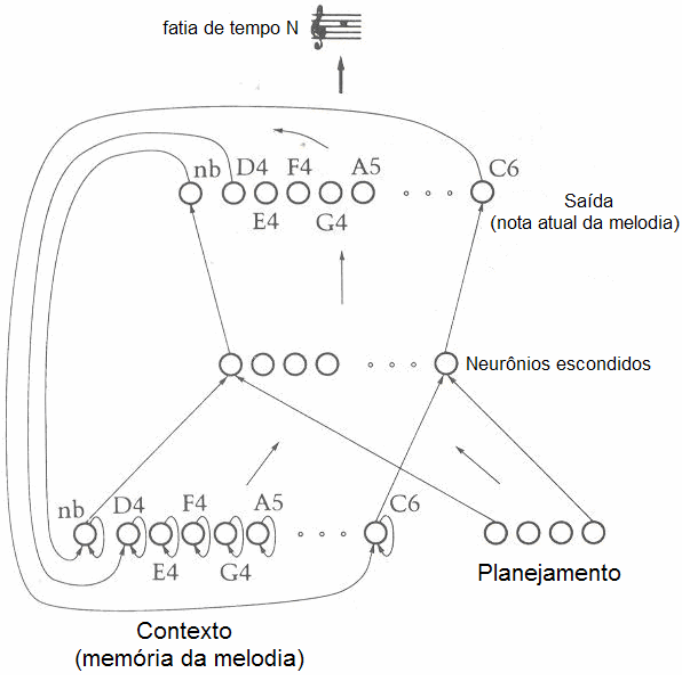


Figura 3.15: A rede seqüencial utilizada por Todd [1989]

Para simplificar, Todd [1989] codifica apenas a altura e duração de cada nota. Essa codificação pode ser feita de duas maneiras [BHARUCHA,1991] [BHARUCHA e TODD,1989]:

- O valor absoluto de cada altura pode ser especificado. Nessa abordagem são necessários neurônios de saída para cada possível altura que a rede pode gerar, tal como um neurônio para C, outro para C# e assim por diante. Essa abordagem foi utilizada por Todd [1989], visto que a rede apresentou um melhor desempenho na fase de treinamento. Ele usou 14 notas para a escala de C maior, da nota D4 até a C6. D4 é representado como 10000000000000; E4 é 01000000000000, e assim por diante.

- O intervalo entre sucessivas alturas pode ser especificado. Nessa abordagem os neurônios de saída correspondem a medidas de mudanças de alturas, os intervalos. Assim um neurônio de saída pode especificar um intervalo ascendente de um semitom, ou um intervalo descendente de três semitons e assim por diante. Por exemplo, para representar A-B-C, a saída de uma representação onde o valor de cada nota é especificado é {A, B, C}, enquanto uma representação de intervalos tem como saída algo do tipo {A, +2, +1}. Com a representação por intervalos, dado um número fixo de neurônios de saída, a representação das notas não fica limitada ao intervalo que consegue cobrir. Por exemplo, se há apenas quatro neurônios de saída, a rede é restrita a representar somente quatro notas. Porém, se esses neurônios representam intervalos entre notas, como por exemplo, + 1 semitom, -1 semitom, 0 semitom, + 2 semitons, qualquer nota pode ser alcançada repetitivamente ascendendo e

descendendo semitons. Além disso, a representação por intervalos favorece com que as músicas criadas sejam independentes de uma tonalidade específica. A não ser pela especificação da altura inicial, a saída da rede não contém indicação da tonalidade, exceto indicações de modos, como maior e menor, baseados nos intervalos utilizados. Independência de tonalidade também permite transposições de uma melodia inteira simplesmente alterando o valor da nota inicial. Em contraste, utilizando a representação de um valor absoluto da nota, ao transpor a melodia seria necessário treinar a rede novamente.

Há uma desvantagem nessa representação por intervalos de notas, conforme mencionado por Todd [1989]. Quando ocorre um intervalo que não é adequadamente gerado na criação de uma seqüência, o restante da melodia será transposto para diferente tonalidade. Essa diferença de tonalidades é bem observada quando ouvida. Portanto, um erro em uma seqüência compromete todo o restante da melodia, ao passo que na representação do valor absoluto da nota esse erro é local, e somente a nota errada é alterada e, portanto, todas as outras notas da melodia têm seus valores inalterados. Essa desvantagem pode ser minimizada especificando a primeira nota de cada seqüência, assim, se a rede cometer um erro, na próxima seqüência volta a produzir as saídas corretas.

Da mesma forma, a duração das notas pode ser representada de duas maneiras:

- A duração pode ser representada por neurônios de saída e de contexto adicionais. Os neurônios podem representar a duração de uma nota localmente, um neurônio representando uma semicolcheia, outro neurônio representando uma colcheia e assim por diante.

- Uma alternativa consiste em dividir a melodia em fatias de tempo iguais e cada saída em uma determinada seqüência corresponde a uma nota que vale uma fatia de tempo. A duração é determinada pelo número de saídas sucessivas e pelo número de fatias de tempo de uma particular nota. O tamanho específico da fatia de tempo pode ser determinado por um fator comum das durações de todas as notas presentes no conjunto de treinamento. Isso garante que a duração de cada nota seja adequadamente representada. Por exemplo, se a melodia é designada para aprender a melodia A-B-C-D correspondendo às durações semínima, colcheia, semínima e semínima pontuada[20] ( ), a fatia de tempo pode ser equivalente à colcheia. A rede neural aprende então a seqüência {A,A,B,C,C,D,D,D}. Uma informação adicional é necessária para esse tipo de representação. É uma indicação da fatia de tempo em

[20] O ponto de aumento serve para aumentar metade do valor da figura que acompanha. Se a semínima vale uma unidade de tempo, com o ponto de aumento ela valerá uma unidade e meia de tempo.

que cada nota começa (*nb* na Figura 3.15). Sem essa indicação não seria possível determinar quando a saída da rede dada por {A,A} indicaria duas notas A, cada uma tendo como duração uma fatia de tempo; ou uma nota A que possui como duração duas fatias de tempo. Essa indicação é feita pelo neurônio "começo de nota" (*note begin*) nas camadas de saída e contexto.

O primeiro passo do treinamento proposto por Todd [1989] consiste em iniciar os pesos com pequenos valores aleatórios. O segundo passo consiste em determinar o planejamento correto para a primeira melodia do conjunto de treinamento. As ativações dos neurônios de contexto são iniciadas com zero, de modo que a rede comece uma seqüência com um contexto vazio. As ativações são passadas pela rede, dos neurônios de planejamento e contexto até atingir os neurônios de saída. A saída obtida pela rede é comparada com a saída desejada e o erro entre as duas saídas é utilizado para ajustar os pesos das conexões pelo método de repropagação do erro. Os valores de saída são passados pelas conexões para serem adicionados ao contexto atual e as ativações são novamente passadas pela rede e os valores de saída e desejados são novamente comparados. Esse ciclo se repete com as conexões que ligam os neurônios de saída com os neurônios de contexto da rede, determinando as próximas saídas e erros e ajustando os pesos para cada fatia de tempo para a primeira melodia. Então, os neurônios de contexto são atribuídos zero novamente, os neurônios de planejamento são arranjados adequadamente para a segunda melodia e todo o processo se repete. Todo esse processo pode ser efetuado para cada melodia do conjunto de treinamento até que o erro total produzido pela rede para esse conjunto de treinamento atinja um determinado *threshold*, ou seja, até que a rede seja capaz de produzir as melodias do conjunto de treinamento de maneira adequada.

Depois que a rede neural seqüencial é treinada para produzir as melodias do conjunto de treinamento, é utilizada para produzir novas melodias com base no aprendizado. As variações das melodias podem ser influenciadas pelo tamanho do conjunto de treinamento utilizado, pela determinação dos neurônios de planejamento, com a utilização de diferentes notas no início da composição, entre outras possibilidades. TODD [1989]

3.3.2 – Abordagem por Laden e Keef [1989]

Para Laden e Keefe [1989] uma preocupação importante na elaboração de redes neurais para aplicações musicais é a representação da entrada para o

sistema. Essa forma de representação pode ser influenciada pelo ponto de vista teórico do pesquisador, pela principal função da rede neural e pelos recursos computacionais disponíveis.

Os autores exploram alternativas na representação da nota para uma rede neural que possui a tarefa de classificar acordes, como maior, menor ou diminuto[21]. A rede é capaz de classificar os acordes através de apresentações dadas a ela de um mesmo acorde transposto ascendentemente ou descendentemente em intervalos.

São utilizados dois tipos de arquiteturas: arquitetura totalmente conectada e arquitetura de camada adjacente. Na primeira categoria cada neurônio possui conexões com todos os neurônios das camadas superiores. Na segunda abordagem cada neurônio tem conexões com cada neurônio somente da camada adjacente. A rede neural possui três camadas: camada de entrada, camada escondida, e camada de saída. O número de neurônios da camada de entrada depende da representação da nota escolhida e para o treinamento é utilizado o algoritmo de aprendizado de retropropagação do erro. O conjunto de treinamento consiste em 12 acordes maiores, 12 acordes menores e 12 acordes diminutos.

Os autores inicialmente escolheram a forma mais simples de representar as alturas, utilizando as doze notas existentes na escala cromática. Um acorde é especificado por três alturas. A Figura 3.16 ilustra uma arquitetura de camadas adjacentes com 12 neurônios de entradas e três neurônios de saída para representar o tipo de acorde e três neurônios na camada escondida utilizada no trabalho desenvolvido por Laden e Keefe [1989].

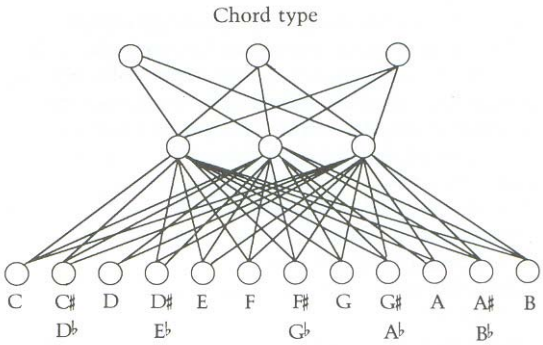


Figura 3.16: RNA proposta para classificar acordes musicais por Laden e Keefe [1989]

Laden e Keefe [1989] também utilizam uma abordagem em que uma nota é representada por suas componentes harmônicas. Essa abordagem é motivada

[21] Acordes são formados por, no mínimo, três notas tocadas simultaneamente, denominadas fundamental, terça e quinta. O acorde diminuto possui um intervalo de terça menor entre a primeira nota (fundamental) e a segunda nota (terça) e um intervalo de quinta diminuta entre a fundamental e a terceira nota (quinta).

pela estrutura espectral de um som musical, pela estrutura psicológica dos padrões de ativações neuronais no sistema auditivo e trazem informações sobre inversões de acordes. O tipo de representação pode influenciar a habilidade da rede de aprender uma tarefa e o número de épocas necessárias para seu treinamento. Redes neurais que utilizam a representação local aprendem mais rápido, enquanto redes com representações complexas das componentes harmônicas classificam os acordes com um melhor desempenho.

3.3.3 – Abordagem por Lewis [1991]

Lewis [1991] desenvolveu um paradigma que utiliza redes neurais para a criação de melodias por refinamento (CBR – Creation by Refinement). CBR consiste em uma fase de aprendizado no qual um algoritmo de aprendizado utilizando gradiente descendente treina a rede neural para desempenhar uma crítica musical, julgando exemplos musicais de acordo com determinados critérios. Após o aprendizado, CBR desempenha a fase de aplicação, na qual uma melodia não adequadamente composta é refinada por um gradiente descendente até que o critério estabelecido na fase de treinamento seja alcançado.

Na fase de aprendizado do CBR, quantidades de exemplos de padrões musicais e não musicais são apresentados como entradas para a rede; e a crítica de cada padrão é apresentada para a saída desejada da rede. A diferença entre a saída desejada e a obtida pela rede é retornada para a rede como um erro de treinamento (E) e os pesos (W) são ajustados pelo gradiente descendente na direção $-\frac{\partial E}{\partial W}$. Um simples exemplo de conjunto de treinamento poderia ser várias seqüências de notas que são consideradas musicais ou não musicais, com uma codificação utilizada para o treinamento, por exemplo, 1 para boas seqüências musicais, 0 para seqüências não musicais e 0.5 razoáveis seqüências musicais.

Na fase de criação, o inverso do procedimento de treinamento é probabilisticamente explorado para gerar novas seqüências. A figura 3.17 ilustra um esquema simplificado de criação por refinamento proposto por Lewis [1991].

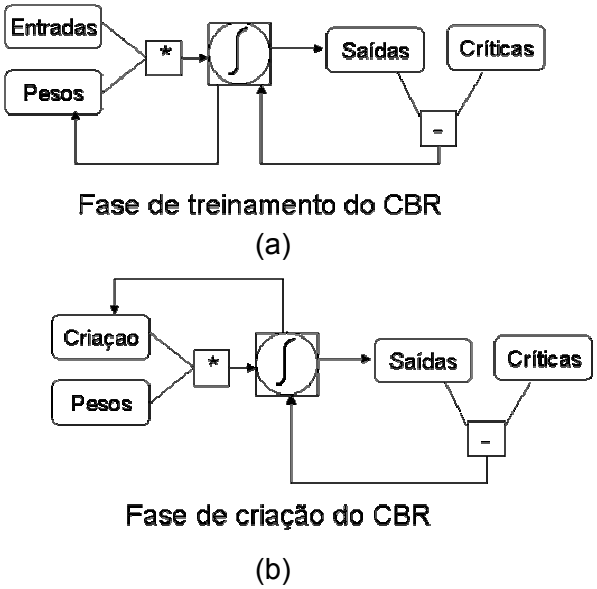


Figura 3.17: O esquema CBR [LEWIS, 1991]

3.3.4 – Abordagem por Mozer [1994]

A arquitetura proposta por Mozer [1994], CONCERT é uma rede neural recorrente (RNN), cuja topologia pode ser observada na Figura 3.18. A rede recebe cada nota da melodia em seqüência e deve produzir como resposta a próxima nota da melodia. Mozer comenta sobre as tabelas de transição de Markov e sobre o algoritmo de Kohonen [1989] [Kohonen, Laine, Tiits e Torkkola, 1991]. Essas também são técnicas de composição nota por nota, no sentido de que as notas são produzidas sequencialmente e linearmente, do começo ao fim da melodia e cada nota depende do contexto anterior.

A representação dos elementos musicais no CONCERT utiliza informações psicológicas. Ou seja, a representação procura ficar próxima ao entendimento musical das pessoas. Notas que as pessoas julgam ser similares teriam representações similares na rede neural. Por exemplo, pares de notas como C1 e C# soam mais similares que C1 e A4. Portanto, representações de notas utilizando freqüências ou formas de codificação direta não seriam capazes de abranger esse entendimento.

Shepard [1982] em sua pesquisa sobre a similaridade das notas propôs uma teoria de generalização na qual a percepção de similaridade de dois itens está diretamente ligada com a distância entre eles em um determinado espaço, seja ele de representação interna ou psicológica. CONCERT representa as notas de acordo com as idéias de Shepard [1982].

A duração da nota é baseada na divisão de cada batida em doze partes. A semínima terá uma duração equivalente a 12/12, a colcheia terá a duração de 6/12, a mínima terá a duração de 24/12 e assim por diante. A representação dos acordes foi feita baseada nos trabalhos de Laden e Keef [1989] que sugeriram uma forma de representação na qual envolvesse características psicológicas.

Com um determinado procedimento para o treinamento proposto por Mozer (1994), os pesos são ajustados tal que CONCERT tenha um bom desempenho em sua tarefa para um conjunto de exemplos de melodias, utilizadas para o treinamento. Os exemplos consistem em seqüência de notas. A nota atual na seqüência é representada pela camada de entrada e a camada de saída representa a próxima nota a ser composta. Tanto as camadas de entrada e saída conseguem representar três atributos das notas: altura, duração e acorde de acompanhamento harmônico. Como a Figura 3.18 indica a próxima nota a ser composta é codificada de duas maneiras diferentes, distribuída e local. A camada que representa a nota distribuída (ND) é a representação interna de nota do CONCERT, dividida em três grupos de neurônios, formando representação distribuída de altura, duração e acorde. A camada que representa a nota localmente (NL) contém um neurônio para cada altura, duração e acorde, permitindo mais facilmente que as saídas finais sejam tratadas como probabilidades.

Após ter sido treinado, CONCERT é executado para criar novas melodias. Para isso, ele é preenchido com pequenas seqüências de notas, provavelmente as notas iniciais do conjunto de treinamento. Após isso, cada resultado da camada de saída é realimentado para a camada de entrada. A saída não gerará como resultado uma nota com absoluta certeza, esse resultado se caracteriza pela distribuição de probabilidade sobre alguns candidatos. A nota resultante final será gerada de acordo com essa distribuição [MOZER,1994].

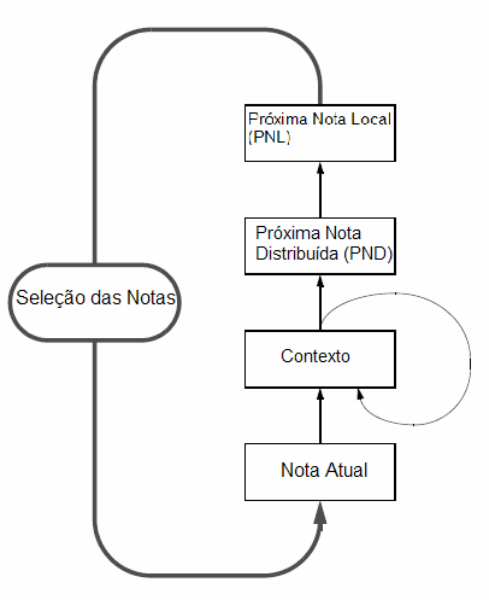


Figura 3.18: Arquitetura da Rede Neural (CONCERT) proposta por Mozer [1994]

3.3.5 – Abordagem por Carpinteiro [1995]

Carpinteiro [1995] propôs uma rede neural para compor seqüências musicais de acordo com três casos de segmentação rítmica: durações longas, pausas e quebras de similaridade. A topologia do modelo é semelhante ao modelo *NETtalk* [22] estudadas por Sejnowski e Rosenberg [1987].

Conforme as sugestões de segmentação de Carpinteiro [1995], exemplo de duração longa (Figura 3.19 (a)), pausa (Figura 3.19 (b)) e quebrada de similaridade (Figura 3.19 (d)) está ilustrado na Figura 3.19.

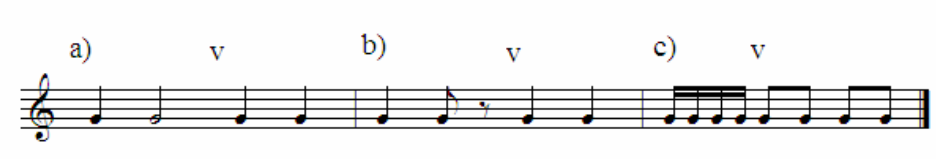


Figura 3.19: Segmentações do ritmo proposta por Carpinteiro [1995]

Se a colcheia (♪) for utilizada como unidade de tempo (UT), todas as outras figuras se tornarão múltiplas da colcheia, ou seja, uma semínima (♩) terá dois tempos, a mínima (𝅗𝅥) terá quatro tempos e assim por diante. Pode-se também definir um contador de unidade de tempo (CTU). Um CTU é uma unidade na qual a

[22] A NETtalk foi a primeira rede paralelamente distribuída a converter o idioma inglês para fonemas. Seu desempenho mostra um aprendizado semelhante ao aprendizado humano.[HAYKIN, 1999]

seqüência musical é medida. Portanto, a cada CTU ou tem-se uma nota sendo soada (quando a nota é continuada, porém sem tocar), ou uma nota é tocada, ou acontece uma pausa (silêncio). No trabalho proposto por Carpinteiro [1995], cada um desses três eventos foi representado por um par de unidades de neurônios de entradas. Os pares estão indicados como 00 quando ocorrer uma pausa, por 10 quando uma nota estiver sendo soada e 11 quando a nota for tocada (Figura 3.20).

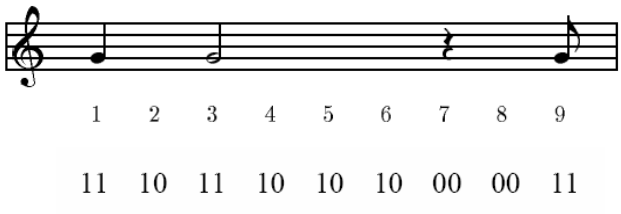


Figura 3.20: Representação do CTU (Contador de Unidade de Tempo para a colcheia como Unidade de Tempo. CARPINTEIRO [1995]

A arquitetura proposta por Carpinteiro [1995] está ilustrada na Figura 3.21. A camada de entrada contém certo número de pares de neurônios de entrada que forma uma janela. O tamanho dessa janela pode variar conforme desejado. Cada par representa um dos três eventos mencionados anteriormente (nota soada, nota tocada e pausa). Esses pares de neurônios da janela são ativados de tal forma a representar um padrão rítmico. Como se pode observar, Carpinteiro [1995] utilizou dois neurônios na camada de saída. Os resultados obtidos mostraram que a rede, que foi treinada com algumas composições de Bach, teve um bom desempenho e que, portanto, segmentação musical pode ser realizada por uma rede neural com aprendizado supervisionado.

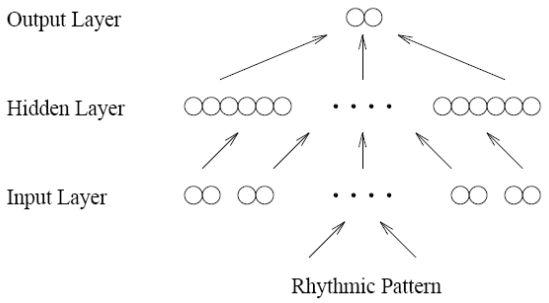


Figura 3.21: Arquitetura proposta por Carpinteiro [1995]

3.3.6 – Abordagem por Chen e Miikkulainen [2001]

Segundo Chen e Miikkulainen [2001] e Eck e Schmidhuber [2002] redes neurais *feedforward* não contém habilidade em armazenar a informação sobre o passado e, portanto, não são indicadas para o processo de geração de composições

musicais como preditor de um passo, uma vez que sempre será necessário repetir padrões de alturas e ritmos. Já as redes recorrentes podem utilizar ativações nas camadas de entrada e escondida como memórias e, portanto, possuem dinâmica temporal.

A arquitetura proposta por Chen e Miikkulainen [2001] é uma rede neural recorrente do tipo SRN (*Simple Recurrent Network*) e está representada na Figura 3.22. Essa rede neural compõe um compasso a cada tempo[23]. Os valores dos neurônios de saída no tempo *t* são copiados para os neurônios de entrada no tempo *t + 1,* e uma cópia da camada escondida é salva na camada de contexto para que a rede possa iniciar de um dado ponto de partida.

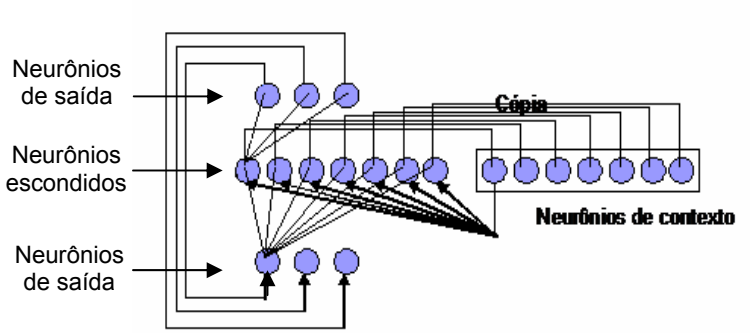


Figura 3.22: Arquitetura proposta por Chen e Miikkulainen [2001].

A rede neural de Chen e Miikkulainen [2001] é totalmente conectada na direção *forward* e seus pesos são evoluídos com a utilização de algoritmos genéticos.

Os autores optaram por representar apenas as cinco primeiras notações rítmicas: semibreve, mínima, semínima, colcheia e semicolcheia. Não são usadas pausas nem figuras pontuadas e o intervalo de notas abrange três oitavas, ou seja, do C2 ao C5. Segundo Chen e Miikkulainen [2001] é possível representar as notas de duas maneiras: nota relativa e absoluta. São raras as pessoas que conseguem identificar notas sem um contexto tonal previamente apresentado. A maioria das pessoas apresentam uma melhor performance para notas relativas por que notas conhecidas são apresentadas anteriormente como referência. Para tanto, é normal treinar uma rede neural com base em notas relativas, ainda mais quando essa rede compõe músicas com base no seu passado.

Continuando com o trabalho de Chen e Miikkulainen [2001], na camada de saída um vetor de neurônios é utilizado para representar os intervalos das notas relativas para uma determinada nota de referência. O neurônio mais a esquerda

[23] A camada de entrada representa um compasso no tempo t, e a camada de saída representa o compasso no tempo t + 1.

possui o maior valor negativo, e o neurônio mais a direita possui o maior valor positivo. O neurônio do meio corresponde a nenhum intervalo especificado, ou seja, a nota se repetirá.

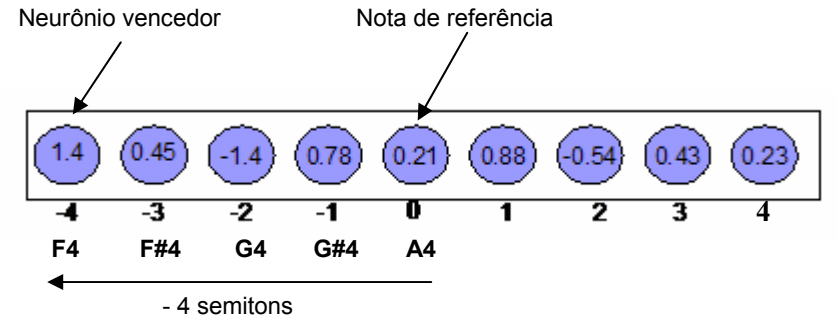


Figura 3.23: Exemplo de geração da próxima nota tendo como nota anterior A4 [Chen e Miikkulainen, 2001]

Conforme ilustrado na Figura 3.23, cada neurônio de saída corresponde a um aumento ou diminuição da altura em um semitom em relação à nota de referência. O neurônio com o maior valor será o vencedor. Na ilustração da Figura 3.23, o vencedor é o neurônio com valor -4, pois seu valor de saída é o maior de todos (1.4). Então, se a nota de referência é A4, o resultado será A4 – 4 semitons = F4.

Chen e Miikkulainen [2001] utiliza a mesma idéia para representar a duração das notas. Ou seja, um vetor de cinco neurônios é utilizado, onde cada neurônio corresponde a uma duração. Como na representação das notas, o neurônio com a maior valor de saída vence e sua duração é atribuída para a nota em questão. Caso haja empate, o vencedor será o neurônio que representa a maior duração (Figura 3.24).

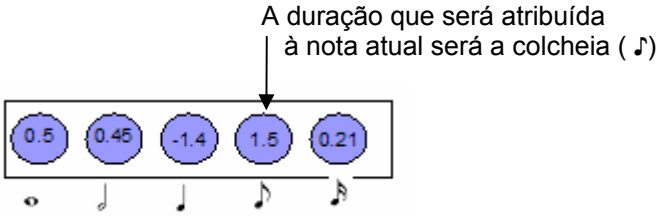


Figura 3.24: Representação da duração segundo Chen e Miikkulainen [2001]

Como citado anteriormente, os compassos são utilizados na melodia para agrupar os tempos em porções iguais, ou seja, a duração de todas as notas dentro de um compasso deve somar a mesma quantidade para todos os compassos da melodia. Nos experimentos de Chen e Miikkulainen [2001] é utilizada uma unidade de tempo como o tamanho do compasso. Assim, dentro de um compasso poderá haver uma semibreve, ou duas mínimas, ou quatro semínimas, ou uma

mínima e duas semínimas, ou qualquer combinação que preencha todo o compasso. Chen e Miikkulainen [2001] criaram o par *D-N* que concatena uma representação da duração e uma representação da nota. Portanto, uma representação de compassos abrange dezesseis pares (*D-N)* e há um algoritmo para a formação dos compassos (Figura 3.25).

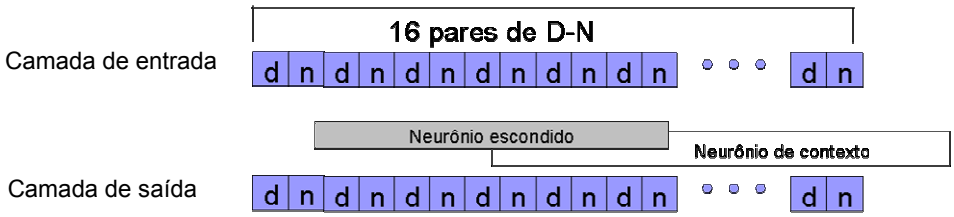


Figura 3.25: Representação dos compassos Segundo Chen e Miikkulainen [2001]

3.3.7 – Abordagem por Rowe [2001]

Rowe [2001] propôs uma rede neural para aprender a identificar a tonalidade de uma seqüência de acordes. Os acordes da seqüência representam o primeiro, quarto e quinto graus de uma escala, denominados tônica, subdominante e dominante, respectivamente. Por exemplo, se a rede recebe como entrada a seqüência `I-IV-V-I` que contém os acordes C–F–G–C, deve corretamente identificar a tonalidade de C maior. Esses graus foram escolhidos por serem os mais importantes dentro de uma escala.

Em um primeiro estudo, a rede possui uma camada de entrada, escondida e de saída com doze neurônios. No conjunto de treinamento, os acordes (`I-IV-V-I`) indicam uma determinada tonalidade maior, representada pelo primeiro grau, a tônica. Um exemplo de treinamento está representado na Figura 3.26, em que a linha de cima de doze valores de ponto flutuante é inserida nos neurônios de entradas e a linha de baixo representa a saída que a rede neural deve associar com a entrada dada.

C	C#	D	D#	E	F	F#	G	G#	A	A#	B
1.0	0.0	0.0	0.0	0.0	**1.0**	0.0	**1.0**	0.0	0.0	0.0	0.0 - entrada
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0 - saída

Figura 3.26: Exemplo de treinamento para a tonalidade de C maior

Posteriormente, Rowe [2001] inseriu como entrada da rede, as notas que representam as sete notas da escala de C maior e obteve os resultados indicados na Figura 3.27 depois de mais de duas mil épocas. Apesar de nota C ter sido indicado com a tônica para essa entrada com um valor menor, comparado com o exemplo original de treinamento da Figura 3.26 (0,37 ao invés de 1,0), nem uma outra nota foi referenciada como possível candidata com algum valor significante.

C	C#	D	D#	E	F	F#	G	G#	A	A#	B
1.0	0.0	1.0	0.0	1.0	1.0	0.0	1.0	0.0	1.0	0.0	1.0 - entrada
0.37	0.0	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0 – saída

Figura 3.27: Exemplo de treinamento que usa todas as notas da escala de C maior

Em um segundo estudo, Rowe [2001] propôs uma rede neural seqüencial totalmente conectada em que os acordes são apresentados em seqüência, e não ao mesmo tempo. A arquitetura é semelhante ao estudo anterior, contêm doze neurônios de entrada, doze neurônios escondidos e doze neurônios de saída. A diferença é que os neurônios de entrada possuem conexões que retornam para si mesmos, e os neurônios de saída possuem conexões que retornam para os neurônios de entrada, como ilustrado parcialmente na Figura 3.28.

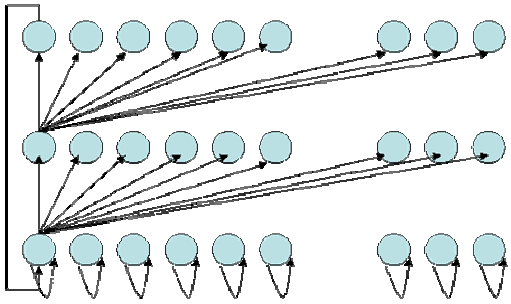


Figura 3.28: Rede neural seqüencial proposta por Rowe [2001, p.102]

As notas são representadas localmente nos neurônios de entrada e saída. Por exemplo, uma rede designada para representar quatro notas distintas precisaria de quatro neurônios de entrada e saída, uma para cada nota, como 0001, 0010, 0100, 1000 representando cada uma das quatro notas.

A rede neural é então treinada para reconhecer progressões de acordes e reconhecer a tonalidade dessa progressão. O conjunto de treinamento deve conter as seqüências relevantes para que a rede neural aprenda a estabelecer a tonalidade correta. A Figura 3.29 lista o conjunto de treinameto utilizado para que a rede neural

aprenda a reconhecer a progressão I-IV-V-I em C maior. Há, portanto, quatro pares entrada/saída para que o aprendizado dessa progressão (da escala de C maior) seja realizado. O conjunto de treinamento total deve conter progressões I-IV-V-I para todas as doze possíveis tônicas. Em cada par da Figura 3.29, a linha de cima representa o conjunto de valores dados aos doze neurônios de entradas, e a linha de baixo representa o conjunto de valores desejados para os neurônios de saída.

C	**C#**	**D**	**D#**	**E**	**F**	**F#**	**G**	**G#**	**A**	**A#**	**B**
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0 -entrada
0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0–saída desejada
0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0-entrada
0.5	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0–saída desejada
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0-entrada
0.5	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0–saída desejada
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0-entrada
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0–saída desejada

Figura 3.29: Pares de treinamento para a progressão I-IV-V-I em C maior

O conjunto de treinamento contém informações sobre os graus I-V (tônica e dominante) também importantes na análise tonal. Quando o primeiro acorde é apresentado à rede (sem nenhum contexto ainda apresentado), ele é indicado como uma evidência fraca (0,5) de ser a tônica da tonalidade. O segundo par do treinamento da Figura 3.29 indica uma progressão do acorde C para o acorde F. Essa progressão poderia representar uma relação I-IV em C maior, a interpretação desses acordes como a tonalidade C continua com a ativação 0,5. Porém, essa progressão também poderia representar a relação V-I (C-F) em F maior, que a rede também deve aprender como sendo uma relação importante. Portanto, a saída desejada é 1,0 para a nota F. Quando o terceiro acorde (G) é apresentado à rede, a progressão C-F-G ainda mantém F maior e C maior como possíveis tonalidades, mas não G maior. O último acorde (C) é apresentado à rede, e a progressão I-IV-V-I se completa corretamente, e a nota C deve ser estabelecida como a tonalidade correta de C maior.

3.3.8 – Abordagem por Eck e Schmidhuber [2002]

Conforme mencionado por Eck e Schmidhuber [2002], uma maneira simples de criar composições musicais com Redes Neurais Artificiais (RNA) sugere com que a rede trabalhe como um preditor de um passo. Assim, a rede é capaz de aprender notas no tempo *t + 1* utilizando como entrada notas referentes ao tempo *t*. Os autores ainda complementam que após o aprendizado a rede pode ser alimentada com valores de entrada do treinamento para que seja possível a criação de novas composições usando suas próprias saídas como entradas subseqüentes.

Eck e Schmidhuber [2002] observam que essa falha das redes neurais é conseqüência de suavização de gradientes (*vanishing gradients*). Quando se utiliza métodos tais como *Back Propagation Through Time* (BPTT) e *Real-Time Recurrent Learning* (RTRL) os erros logo desaparecem ou explodem exponencialmente, e assim se torna-se impossível para as redes neurais gerenciarem corretamente as dependências de longo termo (*long-term-dependencies*). Para música, essas dependências são importantes e permitem com que eventos de várias notas ou compassos contribuam para a formação métrica e estrutura da frase. Como exemplo desses eventos, pode-se citar as mudanças de acorde que permanecem por vários compassos, principalmente em estilos musicais como o *rock-and-roll*.

Portanto, Eck e Schmidhuber [2002] propuseram a utilização de uma rede neural artificial com a arquitetura LSTM *(Long Short-Term Memory*), na qual utiliza unidades lineares chamadas *Constant Error Carousels* (CECs) para minimizar o problema do decaimento do erro presentes em outras arquiteturas recorrentes e assim, essa rede consegue um fluxo de erro mais constante.

A representação dos dados proposta por Eck e Schmidhuber [2002] é feita de forma simples e local. Utiliza-se uma unidade/alvo por nota, com 1,0 representando *on* e 0,0 representando *off.*. Os autores explicam que a preferência dessa representação se deve à não diferenciação de acordes e melodias, porque é fácil obter distribuição de probabilidade sobre um conjunto de notas possíveis e é flexível no sentido que se pode tratar probabilidades com dependência ou não de notas anteriores. A Figura 3.30 mostra as notas utilizadas para o treinamento, tanto para melodias, que estão representadas por notas do C4 ao C5, quanto para acordes, que estão representados por notas do C3 ao C4.

O aprendizado proposto por Eck e Schmidhuber [2002] trabalha tanto com o método gradiente descendente que utiliza um algoritmo BPTT modificado

quanto um algoritmo customizado de RTRN[24] (*Real Time Recurrent Network*). Os dados de treinamento utilizaram uma seqüência popular de acordes de blues com 12 compassos[25], que não variam de melodia para melodia. Os autores utilizaram um preditor de 8 passos, assim a melodia foi gerada pela rede em 96 passos. As inversões de acordes foram possíveis desde que os mesmos não saíssem do intervalo de notas especificado. Nos experimentos foram utilizados tanto somente os acordes quanto melodias acompanhadas dos acordes.

Essas melodias utilizaram a escala pentatônica menor de blues[26]. O treinamento das melodias foi realizado concatenando segmentos de compassos que combinam musicalmente com os acordes. Não foram utilizadas figuras pontuadas e somente foram utilizadas semicolcheias. Segmentos melódicos que caracterizam a forma blues foram selecionados de forma aleatória para comporem o conjunto de dados.

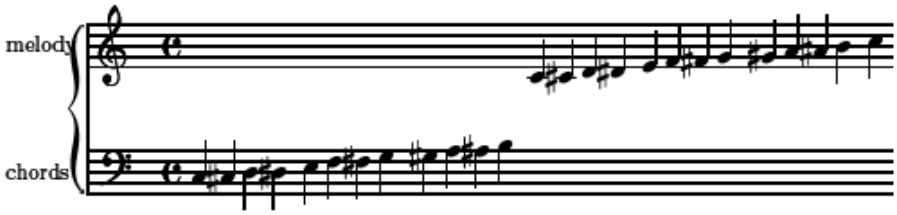


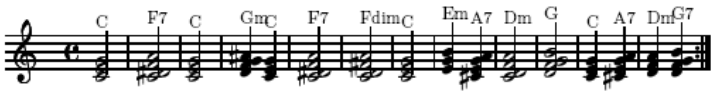
Figura 3.30: Notas utilizadas por Eck e Schmidhuber [2002] para o treinamento da rede neural

3.3.9 – Abordagem por Verbeurgt, Fayer e Dinolfo [2004]

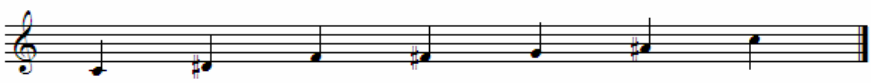
Verbeurgt, Fayer e Dinolfo [2004] usaram uma abordagem híbrida que utiliza redes neurais e cadeias de Markov para composição musical através de exemplos. Nessa abordagem, o primeiro passo consiste em extrair padrões musicais das seqüências de treinamento. Para isso, eles utilizaram uma estrutura de dados caracterizada por uma árvore de sufixos (Figura 3.31 (a)), em que as arestas representam os intervalos, em semitons, entre notas sucessivas. Cada nó na árvore

[24] Essas redes possuem como característica diferencial habilidade para lidar com entradas e saídas que variam no tempo, através do funcionamento presente nelas. [BRAGA, LUDEMIR E CARVALHO, 2000]

[25] Os acordes dessa seqüência são os seguintes (Eck e Schmidhuber,2002,p.4):



[26] Escalas pentatônicas contêm apenas cinco notas e possuem uma entonação triste. Elas existem tanto no modo maior quanto no modo menor. A escala pentatônica menor de blues é uma variação da escala pentatônica menor, acrescida da quarta maior (conhecida como *blue note*):



representa corresponde a um padrão em passos de intervalo. Nós internos correspondem a padrões que ocorrem mais de uma vez nas seqüências de treinamento, e as folhas correspondem a exatamente uma ocorrência do padrão. A posição inicial do padrão na seqüência de treinamento está indicada em cada folha, e a altura de referência da seqüência é dada pela nota nessa posição.

No segundo passo, uma cadeia de Markov é construída com bases nesses padrões (Figura 3.31 (b)), com cada estado correspondendo a um padrão e as transições representam seqüências de padrões permitidas. O estado nomeado como "nenhum" no diagrama corresponde a uma única nota, indicando que não há intervalos para notas sucessivas no padrão. As transições indicam as freqüências nas quais os padrões seguem uns aos outros nas seqüências de treinamento. Os estados iniciais do modelo são aqueles que representam padrões que geralmente ocorrem no começo das seqüências de treinamento.

Por fim, uma rede neural é treinada para aprender a distribuição das notas de referência do estado atual condicionada às notas de referências do estado anterior (Figura 3.31 (c)). Portanto, a entrada da rede neural representa o estado anterior do modelo de Markov juntamente com a nota e duração de referência; e a saída indica a nota e duração de referência do estado atual.

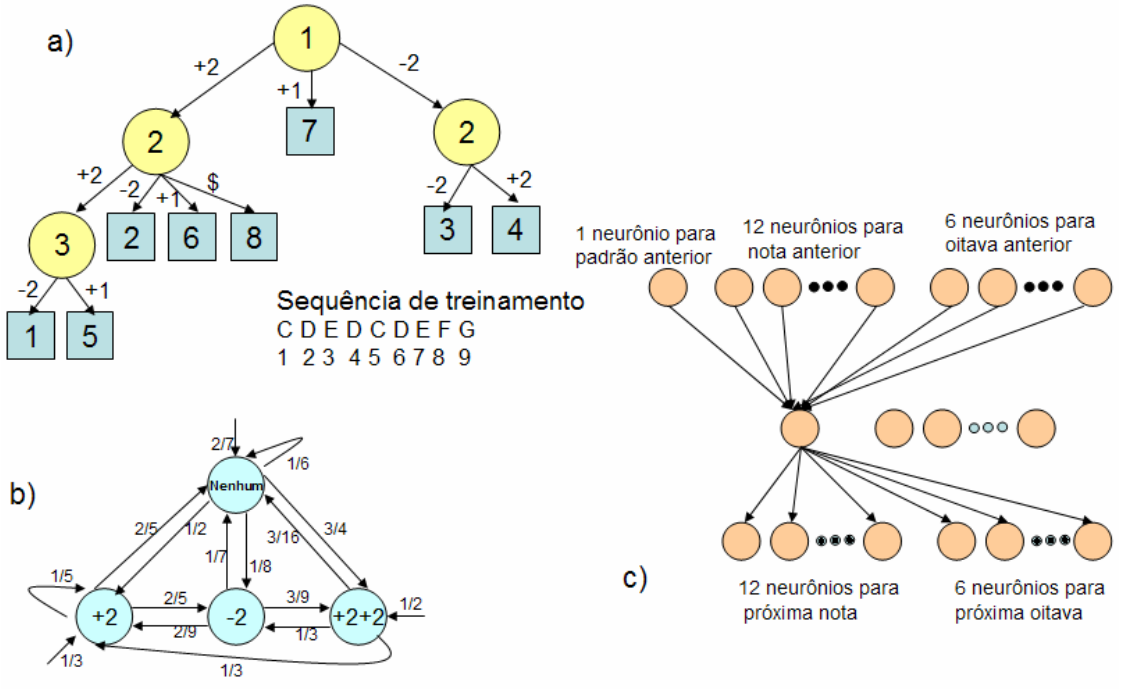


Figura 3.31: Abordagem híbrida Neural-Markov proposta por Verbeurgt, Fayer e Dinolfo [2004] (a) Árvore de Sufixos (b) Modelo de Markov (c) Topologia da Rede Neural (qualidade)

3.3.10 – Abordagem por Frankin [2005]

Franklin treinou uma rede neural LSTM (Long-Short Term Memory) para aprendizagem de seqüências de jazz. A representação das alturas das notas e acordes utilizou sete bits e foi baseada nos intervalos de terça maior e menor. Uma terça maior é formada por um intervalo de quatro semitons entre duas notas, e uma terça menor por uma diferença de três semitons. A Figura 3.32 apresenta os quatro ciclos de terça maiores utilizados por Franklin [2005], numerados de um a quatro, e três ciclos de terça menores, numerados de um a três. Cada ciclo é lido na direção anti-horária. Por exemplo, G# representa a terça maior de E, C representa a terça maior de G#, Eb representa a terça menor de C, F# representa a terça menor de D# e assim por diante.

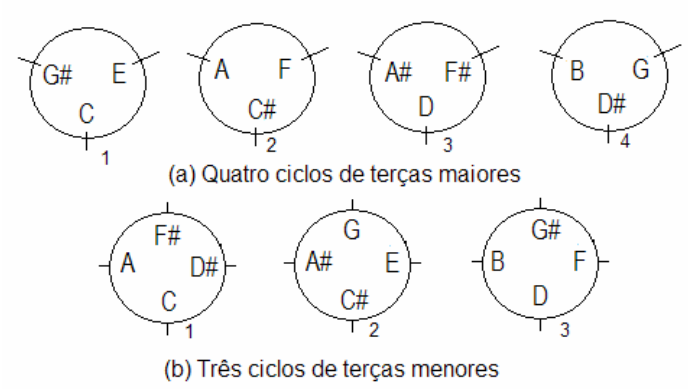


Figura 3.32: Representação por ciclos de (a) terças maiores e (b) terças menores [Franklin, 2005]

Nessa representação das notas, os primeiros quatro bits indicam qual ciclo de terças maiores a nota está localizada, e os três bits restantes indicam qual ciclo de terça menores a nota representa. A representação da nota A, por exemplo, é 0100100, indicando ciclo de terça maior dois e ciclo de terça menor um. A representação da nota E é 1000010, indicando ciclo de terça maior um e menor dois, e por assim em diante. Segundo o autor, essa representação permite um bom aprendizado da rede. A indicação de oitava é dada separadamente, com a inclusão de dois neurônios, um para indicar se a oitava é de C2 até B2, outro para indicar se a oitava é de C4 até B4. Se os dois neurônios possuem valores zero, então a oitava é de C3 até B3.

Conforme anteriormente mencionado, acordes são, no mínimo, tríades de notas representadas pela fundamental, terça e quinta. O autor utiliza a representação por ciclos de terças para cada nota em que o acorde é composto, em um total de vinte e um bits. Porém, o autor também relata que com essa

representação de vinte e um bits seria difícil para a rede aprender relações entre acordes. Por isso, é utilizada sobreposição das notas do acorde, resultando em uma representação de apenas sete bits. Por exemplo, o acorde de G com a representação de ciclos de terças em 21 bits é 0001010 (G), 0001001 (B) e 0010001 (D). A representação por sobreposição é a soma bit a bit dessas três notas: 0 0 1 2 0 1 2 (acorde de G). Nos experimentos realizados, Franklin [2005] verificou que a rede neural poderia aprender melhor se esses valores fossem escalados para o intervalo [0,1]. Portanto, a representação final do acorde de G seria 0 0 0,5 1 0 0,5 1.

Franklin [2005] utilizou uma representação modular para a duração das notas, em que a semínima é dividida em 96 subdivisões, um padrão conhecido como *ticks* na interface MIDI (Musical Instrument Digital Interface). Essa representação da duração permite a utilização de todas as figuras rítmicas.

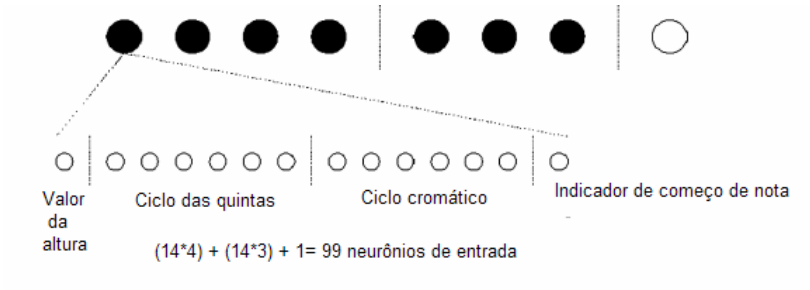
3.3.11 – Abordagem por Adiloglu e Alpaslan [2007]

Adiloglu e Alpaslan [2007] utilizaram redes neurais para geração de contrapontos[27]. Nessa aplicação, o algoritmo de retropropagação do erro é utilizado no treinamento da rede. Os autores escolheram as primeiras espécies[28] de contraponto com duas vozes. A entrada da rede é representada pelas quatro notas mais recentes da segunda voz e pelas três notas da primeira voz, correspondente às três primeiras notas da segunda voz. A rede deverá aprender a quarta nota da primeira voz, correspondente à quarta nota da segunda voz, portanto, a camada de saída possui apenas um neurônio. Um neurônio adicional é utilizado para enfatizar a tônica ou a dominante da peça em questão, para assegurar determinadas regras do contraponto.

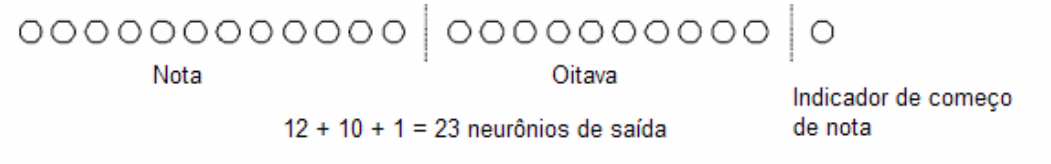
A representação das alturas na camada de entrada é dada através da combinação de valores absolutos (números MIDI, por exemplo) e dos ciclos cromáticos e de quintas, possibilitando a equivalência de oitavas. É utilizada a representação de classe de altura na camada de saída. Essa representação não faz distinção entre oitavas e um neurônio é reservado para cada nota dentro de uma oitava, em um total de 12 neurônios. A informação de oitava é dada explicitamente, com neurônios adicionais. Para duração, é utilizada a mesma idéia de Todd [1989], com o uso de um neurônio adicional para cada nota, para indicar se uma nova nota começa com uma nota fatia de tempo. A figura 1.33 (a) ilustra os neurônios de

[27] Contraponto: a combinação de duas ou mais linhas melódicas e os princípios técnicos a serem considerados para essa combinação.
[28] Contrapontos de primeira espécie, também conhecidos como "nota contra nota" permitem somente a utilização da semibreve. Portanto, cada nota de uma voz corresponde a uma nota nas outras vozes.

entrada, em que cada nota é representada por 14 neurônios. A nota de saída é representada por 23 neurônios (Figura 3.33 (b)).



(a)



(b)

Figura 3.33: Neurônios de entrada (a) e saída (b) por Adiloglu e Alpaslan [2007]

3.4 – CONSIDERAÇÕES FINAIS

Vários estudos foram realizados na tentativa de que sistemas de composição artificial oferecessem bom desempenho e criações melódicas de boa qualidade. Em cada uma das abordagens há vantagens e desvantagens que talvez sejam resultantes do entendimento incompleto do funcionamento da mente humana em domínios como a música. Com as abordagens descritas anteriormente, verificou-se que as redes neurais são bem indicadas como compositores musicais artificiais, pois tentam imitar o entendimento humano sobre percepção musical. Quando as redes neurais trabalham em conjunto com outras técnicas de aprendizado, como algoritmos genéticos ou sistemas baseados em conhecimento, há uma melhor probabilidade de se obter boas melodias, uma vez que as redes não possuem um conhecimento a priori (regras, restrições) e conseguem aprender através de exemplos e serem criativas.

A forma de representação dos elementos musicais também interfere nas composições geradas por esses sistemas. Formas de representação direta e locais possuem baixa complexidade, porém não conseguem armazenar informações psicológicas do entendimento musical humano.

O sistema de composição musical desenvolvido no próximo capítulo busca superar algumas dificuldades encontradas nas abordagens anteriores.

4 – PROPOSTA DE TRABALHO

4.1– CONSIDERAÇÕES INICIAIS

Esse capítulo descreve o projeto desenvolvido de mestrado que aborda um sistema de composição musical com a utilização de redes neurais que também usa como inspiração dados obtidos de contornos de relevos naturais. O sistema proposto pode ser dividido em quatro processos principais para a composição de uma nova melodia usando redes neurais: treinamento, aplicação, avaliação e correção.

Na sessão 4.2 é apresentada uma descrição geral do sistema. A sessão 4.3 apresenta a representação dos elementos musicais, como notas, duração e acordes. As arquiteturas das redes BPTT e LSTM, assim como aspectos de treinamento estão descritos na sessão 4.4. Nessa sessão também é apresentado o método proposto para inicialização de parte dos pesos da rede neural LSTM e para a estimação do número ideal de neurônios escondidos. A sessão 4.5 apresenta a abordagem proposta para avaliação e correção das melodias. Por fim, a sessão 4.6 descreve as considerações finais desse capítulo.

4.2– DESCRIÇÃO GERAL DO SISTEMA

O principal objetivo do sistema é proporcionar uma interface amigável para que os usuários possam utilizar o que foi desenvolvido para criar novas melodias de forma mais dinâmica. Também é adequado que o sistema possa ser aplicável para esses usuários sem a necessidade de um conhecimento aprofundado das técnicas utilizadas. Ainda, esse sistema pode estar disponível na Internet. A tela da interface amigável do sistema foi desenvolvida em inglês para obter maior abrangência em futuras contribuições.

A princípio, o sistema foi desenvolvido em Matlab, mas será migrado para Java, permitindo uma maior acessibilidade. Em conjunto com esse sistema, é utilizado o programa Encore para visualização das partituras e audição das melodias.

O usuário poderá escolher compor melodias com as duas arquiteturas de redes utilizadas nesse trabalho, a BPTT e a LSTM. Uma vez definida a rede utilizada, o usuário pode testar vários parâmetros disponíveis para o treinamento e verificar a influência desses parâmetros na composição de melodias. Após configurar o treinamento da rede, o usuário pode interagir com a tela de composição, para

visualizar e ouvir a melodia composta. O usuário poderá avaliar a melodia de acordo com os requisitos previamente definidos e otimizá-la, se julgar necessário. A Figura 4.1 apresenta a tela principal do sistema.

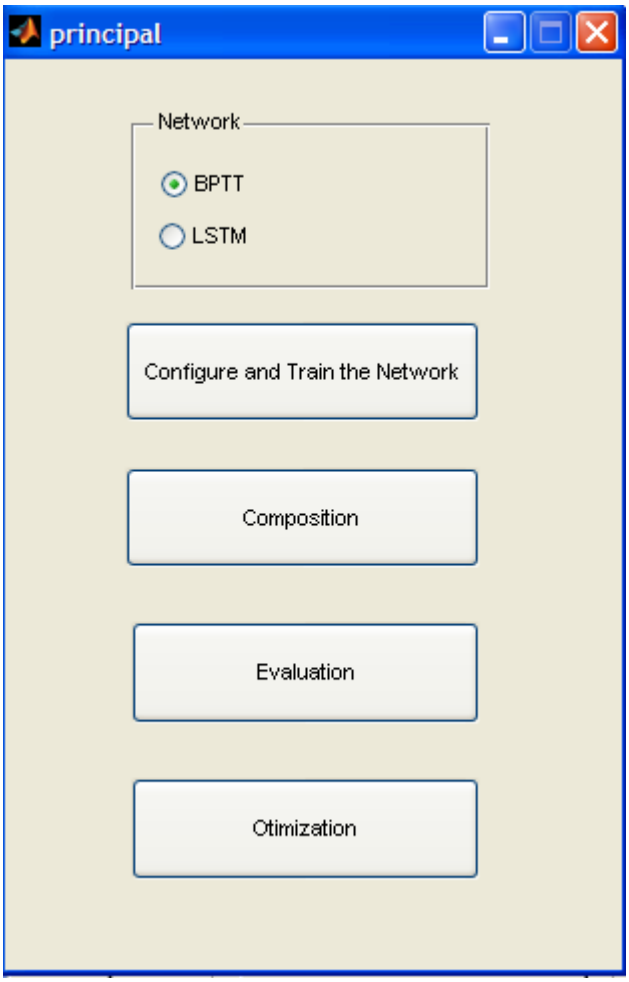


Figura 4.1: Tela principal do sistema

Primeiramente, o usuário escolhe qual rede deseja utilizar para compor. Em seguida ele pode configurar os parâmetros da rede escolhida e executar o treinamento. Quando o treinamento estiver concluído, o usuário usa o sistema para compor novas melodias. A partir da melodia composta, a avaliação pode ser realizada conforme as abordagens propostas pelo trabalho (sessão 4.5). Por fim, as melodias geradas e avaliadas podem ser otimizadas, caso seja o interesse do usuário.

A Figura 4.2 apresenta a tela que representa a parte do sistema para a configuração do treinamento da rede BPTT. O usuário poderá definir alguns parâmetros da rede (quadro superior esquerdo) e do treinamento (quadro superior direito), assim como a inspiração (foto geográfica) e as melodias que farão parte do conjunto de treinamento, permitindo assim, uma maior flexibilidade, visto que esses

parâmetros influenciam na melodia composta. A rede LSTM também possui uma tela para sua configuração e treinamento.

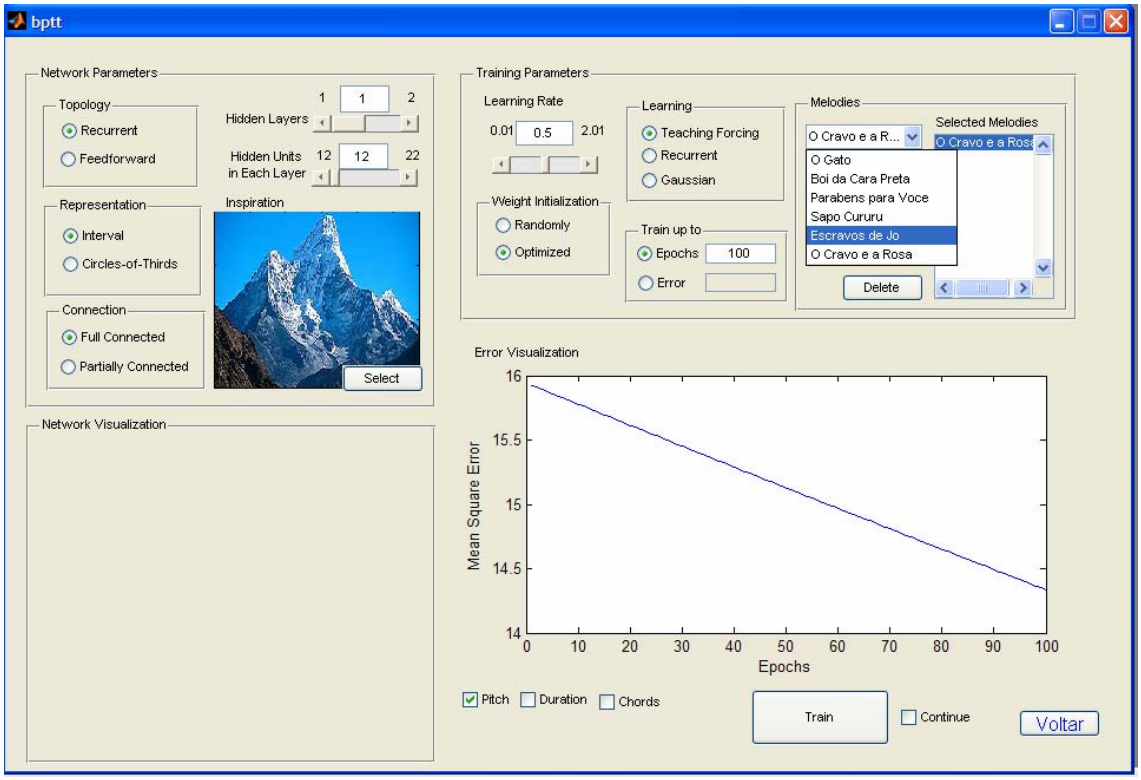


Figura 4.2: Tela da configuração da rede BPTT

4.3 – REPRESENTAÇÃO DOS ELEMENTOS MUSICAIS

Compassos musicais podem ser vistos como primitivas melódicas que se conectam com outros compassos musicais para a formação da melodia. Portanto, na fase de treinamento os compassos musicais devem pertencer a um mesmo estilo musical. Os compassos musicais desse trabalho consistem de notas, duração das notas, pausas e acordes.

4.3.1 – Representação da altura

Esse trabalho utiliza dois tipos de representação da altura: representação por intervalos de notas e representação por ciclos de terças. A representação por intervalos de notas é proposta neste trabalho como uma alternativa às representações existentes. A representação por ciclos de terças foi anteriormente proposta por Franklin [2005], como descrito no capítulo anterior. O objetivo da utilização dessa representação é permitir uma comparação do desempenho das redes.

Como explicado no primeiro capítulo, na representação por intervalos de notas, as alturas são representadas pela combinação de números inteiros e intervalos musicais. Cada nota possui seu próprio número inteiro. Cada intervalo musical determina uma distância de freqüência de uma nota para outra, em semitons ou em freqüência logarítmica. Mesmo com um número fixo de neurônios, vários intervalos de notas podem ser alcançados com essa representação. As notas para o treinamento da rede são normalizadas para o intervalo $[0,1]$ e as pausas são representadas por -1.

Para ilustrar melhor a representação por intervalos, considera-se a nota de referência C4 (representada por 0) e os dois compassos da Figura 4.3, que são os dois primeiros compassos da melodia “Escravos de Jó”. Exemplo da representação por intervalo desses compassos pode ser observada na matriz representada pela equação 4.1, sendo que a primeira linha da matriz corresponde ao primeiro compasso e a segunda linha corresponde ao segundo compasso.

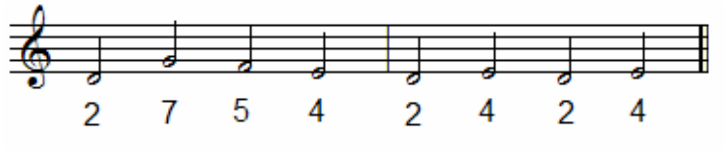


Figura 4.3: Dois compassos musicais

$$notas = \begin{bmatrix} 0.2500 & 0.6667 & 0.5000 & 0.4167 \\ 0.2500 & 0.4147 & 0.2500 & 0.4167 \end{bmatrix} \quad (4.1)$$

A representação por ciclos de terças utiliza sete bits. Os quatro primeiros bits indicam em qual ciclo de terça maior a nota está localizada, num total de quatro ciclos. Os três últimos bits indicam em qual ciclo de ciclo de terça menor a nota está localizada, num total de três ciclos. A informação de oitava é dada separadamente, com dois neurônios adicionais, um para indicar se a oitava é de C3 até B3, outro para indicar se a oitava é de C5 até B5. Se os dois neurônios possuem valor zero, então a oitava é de C4 até B4. Essa representação está mais detalhada na sessão 3.3.10 do capítulo 3. Os compassos da Figura 4.3 estão novamente ilustrados na Figura 4.4. Na matriz representada pela equação 4.2 foi utilizada a representação dos ciclos de terças, em que cada linha da matriz corresponde a uma nota da melodia.

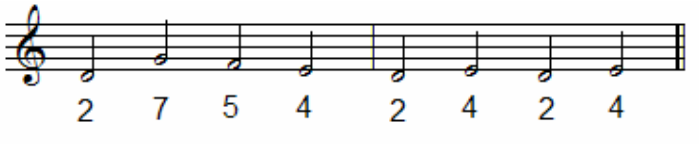


Figura 4.4: Dois compassos musicais

$$notas = \begin{bmatrix} 0\,0\,1\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,1\,0\,1\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,1\,0\,0 \\ 1\,0\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,1\,0\,0 \\ 1\,0\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,1\,0\,0 \\ 1\,0\,0\,0\,0\,1\,0\,0\,0 \end{bmatrix} \quad (4.2)$$

4.3.2 – Representação da duração e acordes

O treinamento da duração das notas é feito separadamente. A representação utilizada para duração é a representação local, em que existe um neurônio de saída para cada figura rítmica a ser representada. Nesse trabalho, são representadas dezesseis figuras rítmicas. A representação da duração pode ser observada na matriz representada pela equação 4.3. Portanto, são necessários dezesseis neurônios de saídas. Cada linha da matriz duração representa uma das figuras rítmicas. Por exemplo, a primeira linha representa a semibreve, a segunda linha representa a mínima pontuada, a terceira a mínima, a quarta a semínima pontuada, a quinta a semínima, a sexta a colcheia pontuada e assim por diante. As duas últimas linhas representam quiáltera[29] de colcheia e de semicolcheia, respectivamente.

[29] O termo rítmico quiáltera caracteriza-se pela execução de três notas no tempo de duas. Por exemplo, considerando a semínima como unidade de tempo, uma quiáltera de colcheia, formada por três colcheias, deve ser tocada em um tempo.

$$duracao = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

Os acordes são representados com a utilização de sete bits, que se caracterizam pela combinação da representação de ciclos de terças das notas que compõe o acorde. Por exemplo, o acorde de Em é formado pela soma das notas E, G e B, conforme Figura 4.5.

1	0	0	0	0	1	0	(E - fundamental)
1	0	0	0	0	0	1	(G – terça menor)
0	0	0	1	0	0	1	(B – quinta justa)
2	0	0	1	0	1	2	acorde de Em

Figura 4.5: Representação do acorde musical Em

Esses valores são normalizados para o intervalo $[0,1]$. Portanto, a representação do acorde de Em usada no treinamento é $1 \quad 0 \quad 0 \quad 0.5 \quad 0 \quad 0.5 \quad 1$. A Figura 4.6 apresenta alguns acordes e suas respectivas representações.

C	–	1	0	0	0.5	0.5	1	0
D	–	0	0.5	1	0	1	0	0.5
F	–	0.5	1	0	0	0.5	0	1
G	–	0	0	0.5	1	0	0.5	1
Am	–	1	0.5	0	0	1	0.5	0
$C7$	–	0.6	0	0.3	0.3	0.3	0.9	0

Figura 4.6: Exemplos da representação de acordes

4.4– ARQUITETURAS

Nessa sessão são apresentadas as arquiteturas das redes BPTT (*Back-Propagation Through Time*) e LSTM (*Long-Short Term Memory*) utilizadas no trabalho[30]. Alguns aspectos de treinamento também são discutidos. Como anteriormente mencionado, a rede LSTM foi criada com o objetivo de minimizar o problema do gradiente que desaparece, presente nas primeiras abordagens de redes neurais recorrentes, como o BPTT. A utilização da rede LSTM neste trabalho tem como objetivo propor novas abordagens para composição musical e comparar os resultados obtidos com a rede BPTT.

4.4.1 – BPTT

O modelo de rede BPTT utilizado nos treinamentos está ilustrado na Figura 4.7. Esse modelo consiste de entradas recorrentes (x_i), não-recorrentes (i_i), uma camada escondida (representada pelos neurônios z_i) e uma camada de saída (neurônios y_i). As entradas recorrentes representam a realimentação de saídas anteriores, ou seja, os compassos de treinamento. As entradas não-recorrentes representam os dados provenientes da inspiração da rede. A camada de saída representa as notas, duração e os acordes musicais. A rede é treinada com a implementação do algoritmo padrão de retropropagação do erro. Na fase de treinamento, a rede neural deve aprender as melodias escolhidas pelo usuário. Na fase de aplicação, a rede deve compor novas melodias baseadas nas melodias previamente treinadas.

[30] Um estudo comparativo entre as redes BPTT e SOM (*Self-Organizing Maps*) para composição musical assistida por computador pode ser observado em [CORREA, SAITO, LEVADA e MARI, 2008].

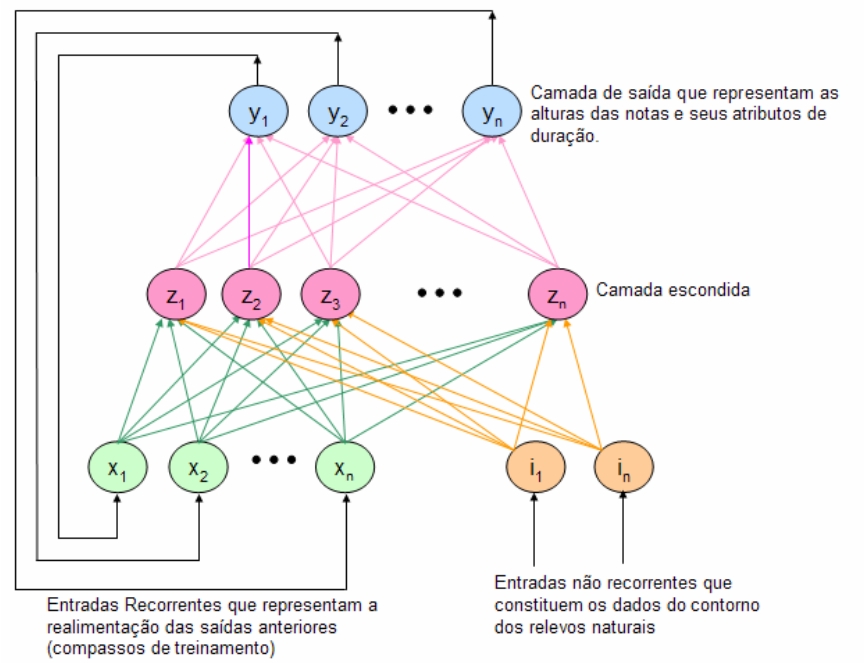


Figura 4.7: Arquitetura da rede BPTT

Em todos os treinamentos, a taxa de aprendizado muda dinamicamente, de acordo com o desempenho da rede. A atualização dos pesos é *offline*, ou seja, os pesos são ajustados depois que todos os pares (entrada, saída desejada) são apresentados à rede. Os neurônios da camada escondida e da camada de saída utilizam função de ativação sigmóide. As notas das melodias de treinamento e das notas dos contornos de relevos naturais formam o vetor de entrada.

Geralmente, no começo do treinamento, as saídas produzidas pela rede serão diferentes das saídas desejadas e isso pode interferir nas equações de atualização dos pesos. Conforme o treinamento continua, as saídas produzidas pela rede estarão mais próximas das saídas desejadas, até que estejam próximas o suficiente de forma que o treinamento possa ser concluído.

Este trabalho propõe otimizar o treinamento da rede BPTT de duas formas. Uma delas consiste no “treinamento forçado”. Considera-se para o treinamento da rede as saídas depois de totalmente treinada. O treinamento forçado considera as saídas produzidas pela rede iguais às saídas desejadas e as fornece aos neurônios de entrada, uma vez que os valores desejados são conhecidos durante o treinamento. Esse procedimento proporciona um treinamento mais rápido.

Uma das desvantagens do treinamento forçado é que ele não é aplicado para os neurônios escondidos, uma vez que os valores desejados para esses neurônios não são conhecidos. Além disso, mesmo quando a rede estiver totalmente treinada, as saídas produzidas podem não serem exatamente iguais às saídas

desejadas. Portanto, quando a rede for utilizada na fase de aplicação, para a composição de novas seqüências musicais depois do treinamento, os valores de saídas serão realimentados para os neurônios de entrada e irão conter variações não presentes no treinamento. A segunda forma de treinamento proposta neste trabalho busca minimizar essa segunda desvantagem. Para tanto, o treinamento incorpora uma função de probabilidade gaussiana nos neurônios de entrada, com média sendo o valor de saída desejado e uma pequena variância (por exemplo, 0.001). Isso significa que os neurônios de entrada não irão receber exatamente os valores de saída desejados, mas sim valores aleatórios que pertencem a um intervalo pequeno e que contém o valor desejado (centro do intervalo) do passo de treinamento anterior. Assim, é possível com que a rede aprenda a lidar com pequenas variações durante o treinamento, melhorando também a fase de aplicação.

Esses dois tipos de treinamentos produzem melodias diferentes e poderão ser escolhidos pelo usuário na interface proposta na sessão 4.2 como uma das opções de treinamento da rede BPTT.

4.4.2 – LSTM

O modelo geral da rede LSTM utilizado nos treinamentos está ilustrado na Figura 4.8 para dois blocos de memória. Os blocos de memória representam a camada escondida da rede. A Figura 4.8 apresenta apenas um tipo de cada conexão. Como na rede BPTT, a camada de entrada representa os compassos de treinamento e as notas da inspiração da rede. A camada de saída representa as notas, duração e os acordes musicais. A rede deve ser treinada com a implementação do algoritmo descrito na sessão 2.5 do capítulo 2. Similarmente à rede BPTT, na fase de treinamento, a rede neural LSTM deve aprender as melodias escolhidas pelo usuário. Na fase de aplicação, a rede deve compor novas melodias baseadas nas melodias previamente treinadas.

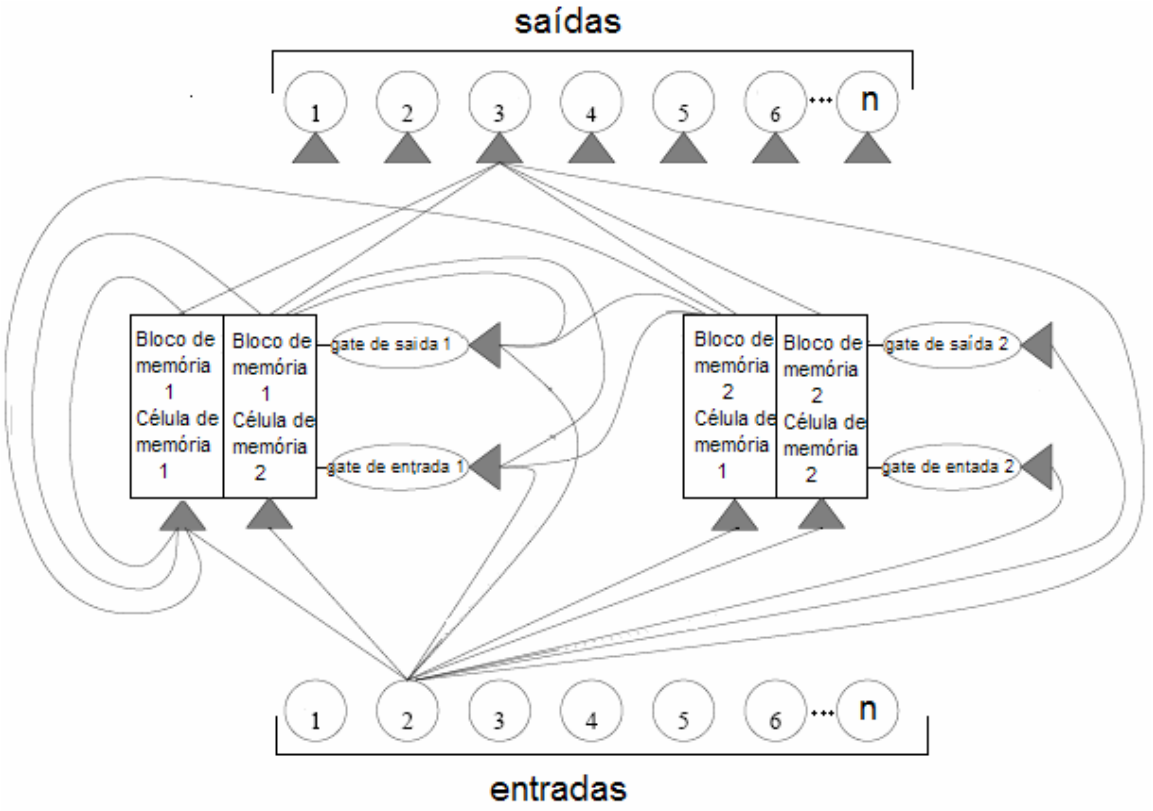


Figura 4.8: Arquitetura da rede LSTM (apenas algumas conexões estão ilustradas)

4.4.2.1 – Otimização da iniciação dos pesos e estimação do número de neurônios escondidos para a rede LSTM

O trabalho de mestrado propõe também um método novo para iniciar os pesos da rede LSTM e estimar a configuração dos neurônios escondidos com o objetivo de otimizar e estabilizar a fase de treinamento, baseado nos trabalhos de Hguyen e Widrow para redes neurais MLP [NGUYEN E WIDROW, 1990]. As equações obtidas para a iniciação dos pesos são baseadas no estudo do comportamento das saídas das células de memória na camada escondida da rede LSTM [CORREA, LEVADA, SAITO, 2008]. A estimação do número ideal de neurônios escondidos é baseada no número de pontos de mínimo e máxima de funções 1-D e 2-D. Para testar e avaliar o método proposto, uma rede neural LSTM de duas camadas foi treinada para aproximar funções não-lineares de uma e duas dimensões.

4.4.2.2 – O comportamento dos neurônios escondidos da rede LSTM na aproximação de funções não-lineares 1-D

Redes neurais com mais de uma camada podem ser usadas para aproximar quaisquer funções arbitrárias, uma vez que elas possuam uma quantidade suficiente de neurônios escondidos [IRIE e MIYAKE, 1988] [NGUYEN e WIDROW,1990].

Basicamente, durante o treinamento com o objetivo de aproximar uma função desejada $d(x)$, a rede neural constrói aproximações lineares por partes $y^i(x)$ da função $d(x)$. Então, as partes são somadas para gerar a função resultante. A idéia é que cada neurônio escondido é responsável por uma determinada aproximação linear $y^i(x)$.

Nessa abordagem, os pesos sinápticos e o *bias* controlam o comportamento dessas aproximações lineares por partes. Durante o processo de aprendizado, os pesos sinápticos da rede devem se movimentar de modo que a região de interesse seja dividida em subintervalos, sendo cada um deles responsável por uma pequena porção da função $d(x)$.

Portanto, é razoável admitir que o processo de treinamento pode ser reduzido iniciando-se os pesos dos neurônios escondidos de modo que cada um dos neurônios seja associado a um subintervalo já no início do treinamento. A rede, então, é treinada normalmente, com cada neurônio escondido ainda tendo a liberdade de se ajustar a um subintervalo durante o processo de aprendizado. Entretanto, a maior parte desses ajustes provavelmente será reduzida, visto que parte da movimentação necessária[31] dos pesos sinápticos já foi eliminada pelo método proposto de iniciação dos pesos.

No exemplo abaixo, $d(x)$ é uma função real a ser aproximada por uma rede neural no intervalo $[-1,1]$. Assim, o tamanho do intervalo é igual a 2. Seja H o número de blocos de memória e M o número de células em cada bloco utilizado para aproximar a função $d(x)$. Portanto, cada unidade de processamento será responsável, em média, por um subintervalo de tamanho $2/HM$.

As funções sigmóides $f(x)$, $g(x)$ e $h(x)$ adotadas nesse trabalho são dadas pelas equações (4.4), (4.5) e (4.6). De acordo com Nguyen e Widrow [1990],

[31] Definição de áreas de maior/menor contribuição de cada neurônio escondido

essas funções sigmóides podem ser consideradas aproximadamente lineares para $x \in [-1,1]$, mas saturam conforme x cresce em magnitude.

$$f(x) = \frac{1}{1+e^{-x}} \tag{4.4}$$

$$g(x) = \frac{4}{1+e^{-x}} - 2 \tag{4.5}$$

$$h(x) = \frac{2}{1+e^{-x}} - 1 \tag{4.6}$$

A partir das equações que fornecem o relacionamento entre a saída de uma rede LSTM contendo um único bloco de memória com uma única célula, para o caso unidimensional, e considerando a primeira iteração, quando não existe retorno das conexões recorrentes e adotando uma aproximação linear para a sigmóide no intervalo $[-1,1]$, é possível escrever:

$$-1 < w_{c_1^1} b_{in_1} b_{out_1} x < 1, \tag{4.7}$$

sendo que $w_{c_1^1} b_{in_1} b_{out_1} x$ é uma aproximação para a saída do $j-$ésimo bloco de memória na etapa de iniciação da rede. Isso implica em:

$$\frac{-1}{w_{c_1^1} b_{in_1} b_{out_1}} < x < \frac{1}{w_{c_1^1} b_{in_1} b_{out_1}}, \tag{4.8}$$

resultando num intervalo de tamanho $2/w_{c_1^1} b_{in_1} b_{out_1}$.

Como descrito anteriormente, é esperado que cada bloco de memória seja responsável, em média, por um subintervalo de tamanho $2/HM$. Portanto:

$$\begin{aligned} \frac{2}{HM} &= \frac{2}{w_{c_1^1} b_{in_1} b_{out_1}} \\ HM &= w_{c_1^1} b_{in_1} b_{out_1} \end{aligned} \tag{4.9}$$

Adotando o esquema de iniciação dos *bias* proposto por GERS [2001], tem-se:

$$\begin{cases} b_{in_1} = b_{out_1} = -0.5 \\ b_{in_2} = b_{out_2} = -1.0 \\ \text{M} \\ b_{in_j} = b_{out_j} = -(0.5 * j) \end{cases} \tag{4.10}$$

Como resultado, uma expressão geral para a iniciação dos pesos $w_{c_j^y 1}$ em uma rede LSTM com apenas uma entrada, é dada por:

$$w_{c_j^y 1} = \frac{HM}{(0,5 \times j)^2} \times \alpha \tag{4.11}$$

Como recomendado por Nguyen e Widrow [1990], é preferível se ter subintervalos com um pouco de sobreposição, de modo que $\alpha \in (0,1)$ representa o coeficiente que controla o nível de sobreposição entre os subintervalos. Ao longo dos experimentos nessa dissertação, considera-se $\alpha = 0,7$, Note que, de acordo com essa metodologia, $w_{c_1^1 1} = w_{c_1^2 1} = \text{K} = w_{c_1^M 1}$, no caso de *M* células em um bloco.

4.4.2.3 – Redes LSTM com múltiplas entradas

A interpretação da aproximação de uma função *N*-dimensional ($N > 1$) é um pouco mais complicada. Basicamente, são utilizados os mesmos conceitos adotados na reconstrução de imagens de tomografia computadorizada [KAK E SLANEY, 2001]. A ferramenta matemática fundamental para essa análise é o Teorema do Corte de Fourier (*Fourier Slice Theorem*). A idéia consiste em tentar se aproximar a Transformada de Fourier (TF) $D(U)$ da função desejada $d(x)$ a partir das fatias 1-D $D_i(U)$. Aplicando-se a Transformada de Fourier Inversa no resultado, é possível definir uma aproximação $\hat{d}(x)$ para a função desejada $d(x)$.

Considerando uma rede neural com duas entradas, uma saída típica de um neurônio escondido possui como sua Transformada de Fourier uma fatia da TF 2-D $D(U)$, denotada por $D_i(U)$. A versão no domínio do tempo de $D_i(U)$, denotada por $d_i(x)$, é uma função de uma única variável e pode ser aproximada por uma rede neural, como descrito na seção anterior. Isso motiva o desenvolvimento de uma metodologia que aproxime S fatias (funções 1-D) utilizando I intervalos cada, adotando *H* blocos de memória com *M* células cada.

Basicamente, a direção dos vetores de peso $\vec{W}_i$ determina a direção da *i-ésima* fatia $D_i(U)$ e a magnitude de $\vec{W}_i$ determina o tamanho do intervalo na aproximação linear por partes da Transformada Inversa de $D_i(U)$, ou $d_i(x)$. Uma saída típica de uma célula de memória da rede LSTM, denotada por $q(x, y)$ sendo x e y as entradas da rede está ilustrada na Figura 4.9. A Transformada de Fourier de $q(x, y)$ está ilustrada na Figura 4.10.

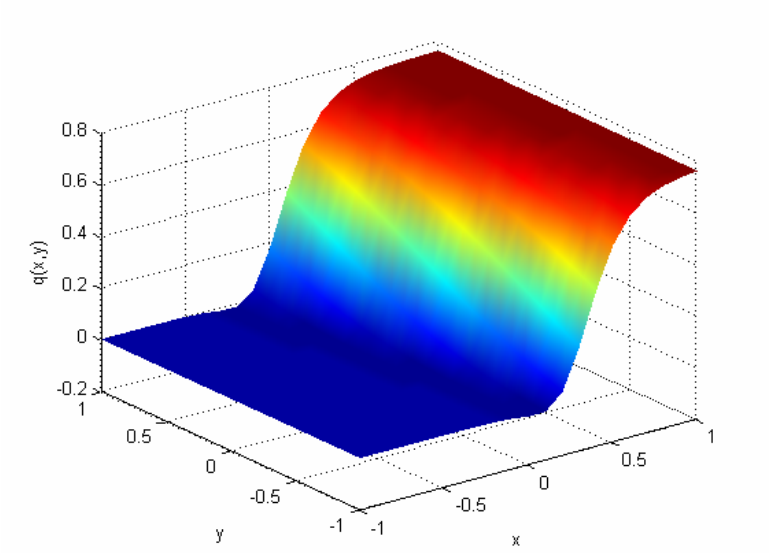


Figura 4.9. Ilustração de uma saída típica de uma célula de memória em uma rede LSTM

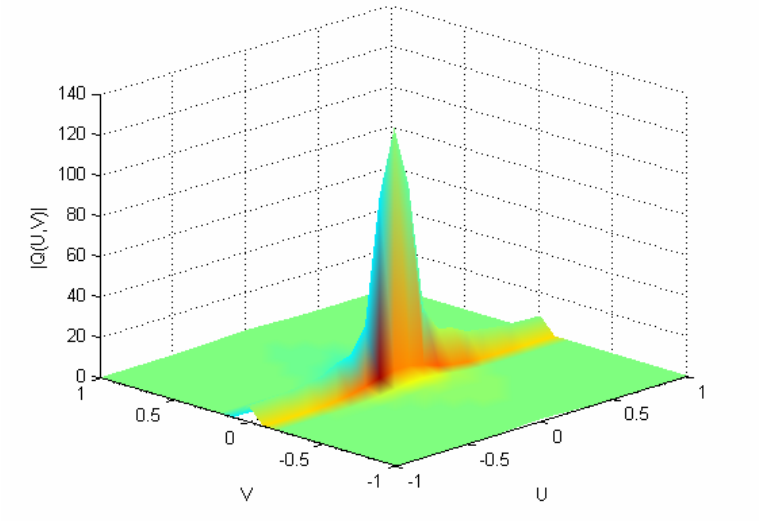


Figura 4.10. A Transformada de Fourier de uma saída típica de uma célula de memória em uma rede LSTM

Como antes do treinamento não é possível saber de antemão o número de fatias que a rede irá produzir, é sugerido em Nguyen e Widrow [1990], que o

número de fatias seja dado pela relação $S = I^{N-1}$, sendo que *N* é o número de entradas da rede. Além disso, como cada elemento do vetor de entrada pertence ao intervalo $[-1,1]$, isso significa que cada intervalo tem comprimento aproximado de $2/I$. Então, pode-se definir:

$$\begin{aligned} HM &= SI \\ HM &= I^N \\ I &= \left(HM\right)^{1/N} \end{aligned} \tag{4.12}$$

Analogamente à equação (4.9), é possível escrever a relação:

$$\frac{2}{\left(HM\right)^{1/N}} = \frac{2}{w_{c_j^y} b_{in_j} b_{out_j}}, \tag{4.13}$$

o que resulta em

$$w_{c_j^y m} = \frac{\left(HM\right)^{1/N}}{\left(0,5 \times j\right)^2} \times \alpha, \tag{4.14}$$

com $\alpha \in (0,1)$. Da mesma maneira, $\alpha = 0,7$ é adotado para permitir certa sobreposição aos subintervalos.

4.4.2.4 – Estimação do número de neurônios escondidos

O objetivo dessa seção é propor uma metodologia para estimar o número de células de memória de uma rede LSTM na aproximação de funções, através do uso do número de pontos de mínimo/máximo da função desejada. Tendo em vista que cada célula de memória é responsável por uma aproximação linear por partes, é razoável assumir que o número mínimo de células de memória deve ser igual ao número de subintervalos existentes entre os pontos de mínimo/máximo locais, como ilustra a Figuras 4.11 (a) e Figura 4.11 (b) para o caso da função não-linear $y(x) = \sin^2(\pi x) + \exp(x^5)$.

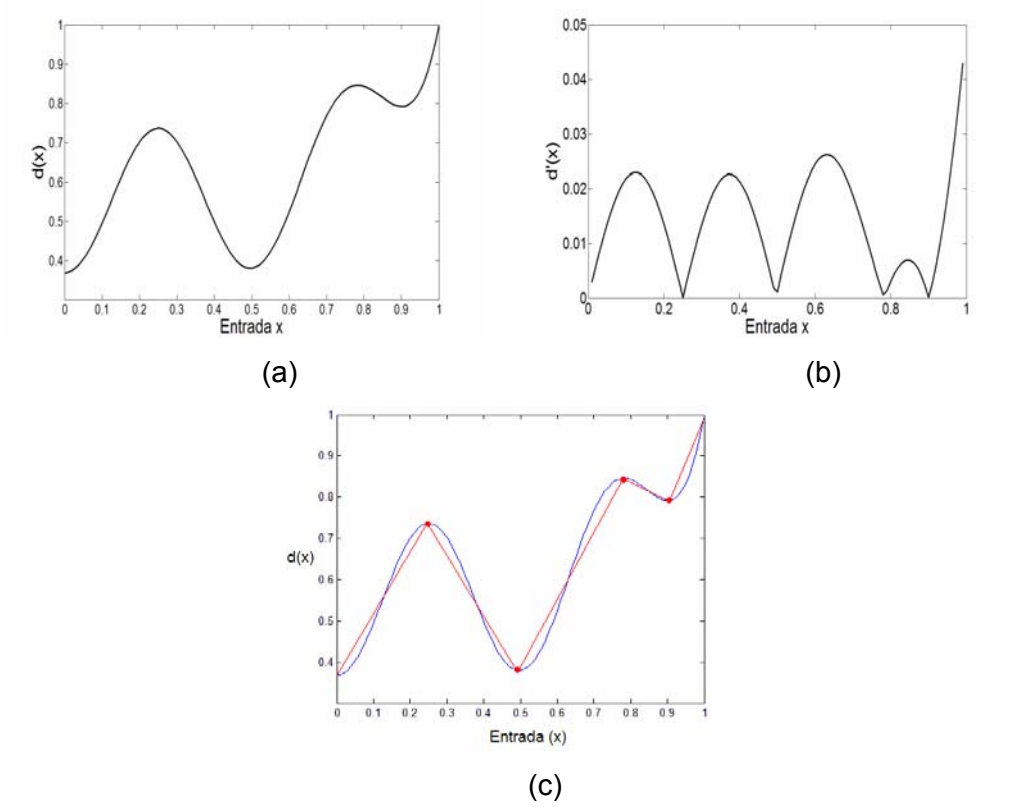


Figura 4.11: (a) Função não-linear 1-D (b) Detecção dos pontos extremos (c) Aproximação linear obtida através dos pontos extremos

Conectando os pontos extremos da função é possível se obter uma boa aproximação para a função (Figura 4.11 (c)). A motivação para essa abordagem consiste em um resultado extremamente importante da teoria de análise de formas: pontos de alta curvatura concentram informação geométrica [COSTA E CÉSAR, 2001]**,** fornecendo bons descritores de formas. Nesse trabalho, tais pontos podem ser bem aproximados pelos extremos locais das funções desejadas.

A metodologia proposta consiste em utilizar a informação presente na função desejada $d(x)$, mais precisamente, o número de mínimos/máximos locais, juntamente com o número de neurônios da camada de entrada da rede, para calcular uma estimativa inicial do número de unidades escondidas necessárias para aproximar a função $d(x)$. Nota-se que de acordo com a metodologia proposta, é possível associar uma configuração de rede específica dependendo da função objetivo observada, ou seja, a rede é configurada conforme o problema.

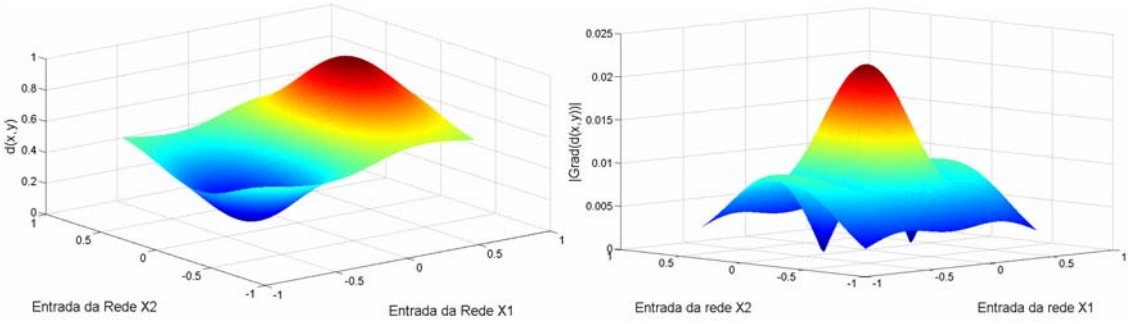
A detecção dos pontos extremos da função é implementada através do método de diferenças finitas para o cálculo das primeiras derivadas. Basicamente, existem três tipos fundamentais de aproximações utilizando diferenças finitas descritas na literatura: aproximação usando diferença para frente, aproximação usando

diferença para trás e aproximação utilizando diferença centrada. A vantagem da diferença centrada é que ela fornece a aproximação mais precisa em termos da série de Taylor. A expressão para a primeira derivada, em um ponto x, utilizando a diferença centrada é dada por [SMITH, 1985]:

$$\frac{d}{dx} f(x) \approx \frac{f(x+h) - f(x-h)}{2} \tag{4.15}$$

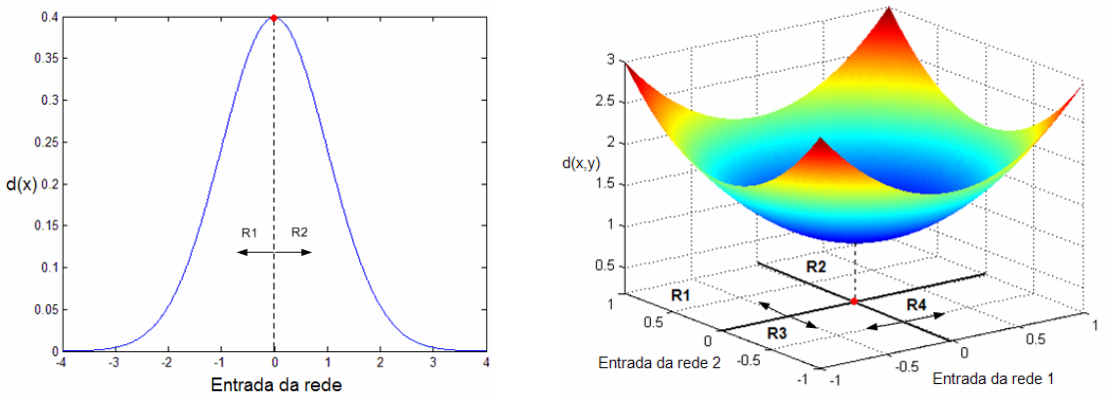
No caso de funções reais 2-D, o gradiente de $f(x,y)$ pode ser aproximado pelas diferenças centrais nas direções x e y. A Figura 4.12 (a) mostra a função 2-D $d(x,y) = 0,25 + x\exp\left\{-(x/0,5)^2 - (y/0,5)^2\right\}$ e a Figura 4.12 (b) ilustra os pontos extremos detectados (condição $\nabla d(x,y) = \vec{0}$).

Para funções 1-D, a definição de um ponto de mínimo/máximo divide o domínio da função em 2 regiões não-sobrepostas (Figura 4.13 (a)). No caso de funções definidas no ¡ ^{2}, o plano é subdividido em 4 quadrantes não-sobrepostos (Figura 4.13 (b)). Esse fato motiva a utilização de uma heurística para configurar o número de neurônios escondidos (HM) necessários para aproximar a função. Na verdade, HM é proporcional a $(n^2K)+1$, em que n denota o número de entradas da rede (dimensionalidade da função) e k é o número de pontos extremos da função em um dado intervalo de interesse.



(a) (b)

Figura 4.12: (a) Função 2-D (b) Pontos extremos detectados



(a) (b)

Figura 4.13: Divisão do domínio da função por um ponto extremo local (a) Duas Regiões (b) Quatro Regiões

4.5– COMPOSIÇÃO DAS MELODIAS

As redes são treinadas separadamente para cada uma das melodias do conjunto de treinamento. Depois que o treinamento é concluído, cada rede neural desempenha a fase de aplicação com a composição de uma nova melodia. Na fase de aplicação, as primeiras notas do treinamento são dadas para cada rede treinada, e a melodia é composta utilizando as saídas obtidas por cada uma das redes. Portanto, se o conjunto de treinamento é formado por cinco melodias, uma rede é treinada para cada melodia e assim, na fase de aplicação, teremos uma nova melodia composta por cada uma das cinco redes. A estratégia final de composição utilizada nesse trabalho consiste na obtenção de cada nota selecionada baseada na escolha de uma das notas das melodias geradas pelas redes, incluindo informações estatísticas sobre todas as melodias do conjunto de treinamento através da elaboração de uma tabela de probabilidade condicional para cada um dos atributos: notas, duração e acordes.

A tabela de probabilidade condicional das notas (alturas), por exemplo, contém informações sobre a freqüência de ocorrência de notas de todas as melodias do conjunto de treinamento. A tabela mostra a probabilidade de ocorrência da nota (x) dada a ocorrência da nota anterior (y). Essa análise foi feita com todas as notas de todas as melodias do conjunto de treinamento. Essa informação é então usada para escolher a nota que fará parte da melodia final. Seguindo o exemplo anterior, se há cinco melodias compostas por cinco redes diferentes, a melodia final é composta da seguinte forma:

- Seleciona-se a primeira nota aleatoriamente, ou seleciona-se a nota com maior freqüência nas primeiras notas das cinco melodias compostas.
- Com as cinco notas candidatas seguintes (uma de cada melodia composta), a próxima nota selecionada será aquela com maior probabilidade na tabela, dada a ocorrência da primeira já selecionada.
- Continua-se o passo anterior até não haver mais notas a serem selecionadas.

Para duração e acordes, a estratégia de composição é a mesma.

Como exemplo de tabela de probabilidade condicional das notas, a Tabela 4.1 apresenta as probabilidades das notas de 12 melodias usadas no conjunto de treinamento, extraídas de músicas tradicionais ou folclóricas brasileiras. As 12 melodias são: “O cravo e a rosa”, “Onde está a Margarida”, “Casinha pequenina”, “Samba lêlê”, “Peixe vivo”, “O pobre e o rico”, “O Gato”, “Oh! Minas Gerais”, “Mulher Rendeira”, “Escravos de Jó”, “Sapo Cururu” e “Boi da cara preta”.

Tabela 4.1: Exemplo de probabilidades condicionais das notas

$y \setminus x$	C	$C\#$	D	$D\#$	E	F	$F\#$	G	$G\#$	A	$A\#$	B
C	0.25	0	0.17	0	0.17	0.01	0	0.04	0	0.08	0	0.28
$C\#$	0	0	0	0	0	0	0	0	0	0	0	0
D	0.23	0	0.10	0	0.32	0.13	0.03	0.04	0	0	0	0.14
$D\#$	0	0	0	0	0	0	1.0	0	0	0	0	0
E	0.27	0	0.27	0	0.11	0.06	0	0.22	0	0.06	0	0.01
F	0.02	0	0.15	0	0.28	0.26	0	0.08	0	0.13	0	0.08
$F\#$	0	0	0.07	0	0.29	0	0	0.29	0	0.36	0	0
G	0.05	0	0.05	0	0.13	0.16	0.06	0.35	0	0.09	0	0.11
$G\#$	0	0	0	0	0	0	0	0	0	0	0	0
A	0.05	0	0.01	0.01	0	0.06	0.05	0.41	0	0.29	0	0.12
$A\#$	0	0	0	0	0	0	0	0	0	0	0	0
B	0.18	0	0.13	0	0	0.01	0	0.10	0	0.32	0	0.26

Para exemplificação do processo de composição, consideram-se três redes LSTM treinadas para as melodias “Escravos de Jó”, “O Boi da Cara Preta” e “Onde está a Margarida”, que fazem parte da tabela de probabilidade (4.1). Em seguida considera-se a geração da primeira etapa de novas melodias na fase de aplicação, mostradas pelas matrizes $EscJó$, $BoiCaraP$ e $OndeEstaM$, na qual cada

elemento está sendo mostrado por um número que representa a nota e pela nota em representação alfabética entre parênteses.

$$EscJó = \begin{bmatrix} 2(D) & 3(D\#) & 3(D\#) & 1(C\#) & 5(F) & 11(B) & 3(D\#) & 3(D\#) & 6(F) & 8(G) \\ 7(G) & 4(E) & 1(C\#) & 4(E) & 8(G\#) & 9(A) & 1(C\#) & 1(C\#) & 4(E) & 7(G) \\ 5(F) & 2(D) & -1(B2) & -1(B2) & 10(A\#) & 6(F\#) & -1(B2) & -1(B2) & 5(F) & 8(G\#) \\ 4(E) & 5(F) & 3(D\#) & 2(D) & 11(B) & 1(C\#) & 1(C\#) & 2(D) & 6(F\#) & 8(G\#) \end{bmatrix} \quad (4.16)$$

$$BoiCara\,\mathrm{P} = \begin{bmatrix} 1(C\#) & 9(A) & 4(E) & 1(C\#) & 7(G) & 3(D\#) \\ 1(C\#) & 5(F) & 0(C) & 2(D) & 4(E) & 6(F\#) \\ 1(C\#) & 1(C\#) & -1(B2) & 5(F) & 2(D) & 6(F\#) \\ 11(B) & 4(E) & -1(B2) & 8(G\#) & 0(C) & 9(A) \end{bmatrix} \quad (4.17)$$

$$OndeEstaM = \begin{bmatrix} -1(B2) & 7(G) & 10(A\#) & -1(B2) & 4(E) & 9(A) & 8(G\#) \\ -1(B2) & 9(A) & 8(G\#) & -1(B2) & 3(D\#) & 10(A\#) & 9(A) \\ 6(F\#) & 7(G) & 4(E) & -1(B2) & 7(G) & 9(A) & 9(A) \\ 6(F\#) & 8(G\#) & 1(C\#) & 4(E) & 5(F) & 3(D\#) & 5(F) \end{bmatrix} \quad (4.18)$$

O processo de composição proposto gera a matriz $Final$ que representa a melodia final gerada pela combinação das melodias $EscJó$, $BoidaCaraP$ e $OndeestaM$.

A matriz $Final(EscJó)$ (equação 4.19) mostra as notas finais selecionadas da matriz $EscJo$ (equação 4.16); a matriz $Final(BoiCaraP)$ (equação 4.20) mostra as notas selecionadas da matriz $BoidaCaraP$ (equação 1.17); e finalmente a matriz $Final(OndeestaM)$ (equação 4.21) as notas selecionadas da matriz $OndeestaM$ (equação 4.18). A obtenção das matrizes $Final(EscJó)$, $Final(BoiCaraP)$ e $Final(OndeestaM)$ é baseada na escolha de maior probabilidade conforme Tabela 4.1. A composição final é obtida pela união dessas três matrizes, conforme mostrado pela matriz $Final$ (equação 4.22).

$$Final(EscJo) = \begin{bmatrix} 2(D) & - & - & - & - & 11(B) & - & 3(D\#) & 6(F\#) & 8(G\#) \\ - & - & - & - & - & 9(A) & - & 1(C\#) & 4(E) & 7(G) \\ 5(F) & 2(D) & -1(B2) & - & - & - & - & -1(B2) & 5(F) & 8(G\#) \\ 4(E) & - & - & 2(D) & - & - & - & 2(D) & 6(F\#) & 8(G\#) \end{bmatrix} \quad (4.19)$$

$$Final(BoiCaraP) = \begin{bmatrix} - & 7(G) & 4(E) & - & - & - & - & - & - & - \\ - & 5(F) & 0(C) & - & 4(E) & - & - & - & - & - \\ - & - & - & - & 2(D) & - & - & - & - & - \\ - & 4(E) & -1(B2) & - & 0(C) & 9(A) & - & - & - & - \end{bmatrix} \quad (4.20)$$

$$Final(OndeestaM) = \begin{bmatrix} - & - & - & -1(B2) & 4(E) & - & 8(G\#) & - & - & - \\ -1(B2) & - & - & -1(B2) & - & - & 9(A) & - & - & - \\ - & - & - & -1(B2) & - & 9(A) & 9(A) & - & - & - \\ - & - & - & - & - & - & 6(F\#) & - & - & - \end{bmatrix} \quad (4.21)$$

$$Final = \begin{bmatrix} 2(D) & 7(G) & 4(E) & -1(B2) & 4(E) & 11(B) & 8(G\#) & 3(D\#) & 6(F\#) & 8(G\#) \\ -1(B2) & 5(F) & 0(C) & -1(B2) & 4(E) & 9(A) & 9(A) & 1(C\#) & 4(E) & 7(G) \\ 5(F) & 2(D) & -1(B2) & -1(B2) & 2(D) & 9(A) & 9(A) & -1(B2) & 5(F) & 8(G\#) \\ 4(E) & 4(E) & -1(B2) & 2(D) & 0(C) & 9(A) & 6(F\#) & 2(D) & 6(F\#) & 8(G\#) \end{bmatrix} \quad (4.22)$$

4.6– AVALIAÇÃO E OTIMIZAÇÃO DAS MELODIAS

O trabalho propõe avaliar as novas melodias compostas pelas redes. Como avaliar música às vezes é subjetivo, essa avaliação é aplicada com base em três requisitos: notas repetidas na melodia (NRM), mudanças abruptas de altura (MAA) e notas fora da tonalidade (NFT); e dois critérios: apropriadas ou inapropriadas. Portanto, uma melodia composta é classificada como não apropriada se existem muitas notas repetitivas, se há significante número de ocorrências de mudanças abruptas de altura de uma nota para a seguinte, por exemplo, de C4 para D6; ou se há significante quantidade de notas que não pertencem à tonalidade da melodia.

Para isso, um conjunto de melodias folclóricas ou tradicionais brasileiras é selecionado como exemplos de melodias apropriadas. Esse conjunto não necessariamente precisa conter as mesmas melodias do conjunto de treinamento. Os três atributos, NRM, MAA e NFT são extraídos desse conjunto, ou seja, para cada exemplo de melodia apropriada são coletadas informações sobre a incidência de notas repetidas, mudanças abruptas de altura e notas fora da escala.

A Tabela 4.2 apresenta os resultados obtidos do processo de extração de atributos para 10 melodias apropriadas. Todos os atributos foram normalizados para o intervalo $[0,1]$. Similarmente, foram criados 10 exemplos representando melodias inapropriadas e a mesma informação de incidência foi coletada. A Tabela 4.3

apresenta os vetores de atributos obtidos para 10 exemplos de melodias inapropriadas.

Tabela 4.2: Exemplos de atributos extraídos de 10 melodias apropriadas

	NRM	MAA	NFT
O Gato	0,26	0	0
Mulher Rendeira	0,28	0	0
O Cravo e a Rosa	0,20	0	0
Tocam os Sinos	0,31	0	0
Escravos de Jô	0,05	0	0
Oh! Minas Gerais!	0,03	0,015	0
Marinheiro Popeye	0,27	0	0
Era uma casa	0,10	0	0
Noite Feliz	0,18	0	0
Macaquinho	0,28	0	0,04
Média	**0,196**	**0**	**0,004**

Tabela 4.3: Exemplos de atributos extraídos de 10 melodias inapropriadas

	NRM	MAA	NFT
Melodia NA 1	0,4	0,13	0
Melodia NA 2	0,38	0,2	0,017
Melodia NA 3	0,08	0,017	0,45
Melodia NA 4	0,25	0,16	0,13
Melodia NA 5	0,33	0,06	0,1
Melodia NA 6	0,05	0,17	0,08
Melodia NA 7	0,16	0,08	0,25
Melodia NA 8	0,33	0,1	0,13
Melodia NA 9	0,08	0,17	0,2
Melodia NA 10	0,36	0,13	0,27
Média	**0,242**	**0,12**	**0,1627**

Os atributos NRM, MAA e NFT são extraídos das novas melodias criadas pelas redes BPTT e LSTM usando representação por intervalo e por ciclo das terças.

O processo de avaliação é realizado de duas maneiras. Basicamente, o objetivo é classificar a melodia gerada pela rede em duas classes: apropriada e inapropriada.

A primeira abordagem consiste em medir a similaridade entre o vetor de atributos da melodia gerada e o vetor média das duas classes (apropriada e inapropriada) através da distância euclidiana (norma L2). Se o vetor de atributos está mais perto do vetor média correspondente a melodias apropriadas, a melodia é classificada como apropriada. Caso contrário, ou seja, se o vetor de atributos da

melodia gerada estiver mais próximo do vetor média correspondente às melodias inapropriadas, então a melodia é classificada como inapropriada.

A segunda abordagem consiste em treinar uma rede MLP (*Multi-Layer Perceptron*) para classificação nas duas classes, cuja arquitetura está ilustrada na Figura 4.14. Para representar a classe de melodia apropriada, os padrões de treinamento consistem nos vetores de atributos extraídos dos exemplos de melodias apropriadas e a rede é treinada para produzir saída 0 quanto recebe um desses vetores. Da mesma forma, a rede MLP é treinada para produzir saída 1 se recebe como entrada um dos vetores de atributos que representam os exemplos de melodias inapropriadas.

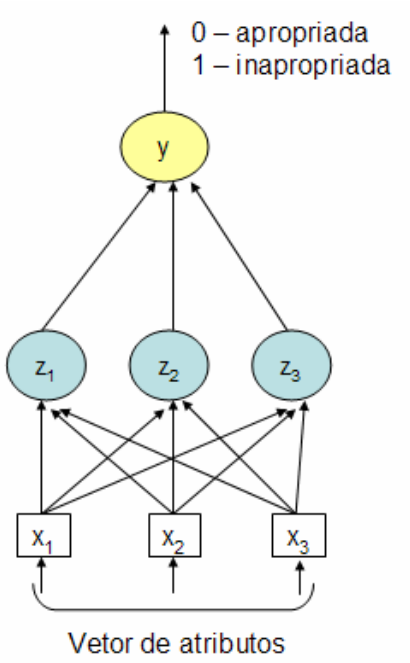


Figura 4.14: Arquitetura da rede MLP utilizada para avaliação das melodias

Na fase de aplicação, os padrões de entrada são os vetores de atributos das novas melodias geradas pela rede. É verificado então se a saída da rede está próxima de 0 (então essa melodia será classificada como apropriada) ou se a saída da rede está próxima de 1 (então a melodia será classificada como inapropriada).

As melodias classificadas como inapropriadas passam pelo processo de correção. A correção é feita pela identificação das notas pertencentes aos atributos NRM, MAA e NFT, ou seja, o algoritmo de correção proposto identifica ocorrências de notas repetidas, notas com mudanças abruptas de altura e notas fora de tonalidade nas melodias inapropriadas. Essas notas representam partes não desejadas nas melodias. As notas identificadas são corrigidas de acordo com a tabela de

probabilidade condicional que contém informações sobre a freqüência de ocorrência de notas de todas as melodias do conjunto de treinamento.

Por exemplo, considera-se a seguinte seqüência de notas: C4 D4 F5 G4 G4. A passagem da nota D4 para a F5 representa uma mudança abrupta de freqüência (altura). De acordo com a tabela de probabilidade condicional ocorre E depois de ocorrer D com maior probabilidade. Na seqüência anterior também há duas notas repetidas (G4 G4). Segundo a mesma tabela, é maior a probabilidade de ocorrer G depois de G. Como selecionar a nota G manteria o problema de notas repetidas, busca-se na tabela a segunda nota com maior probabilidade, no caso E. Portanto, a seqüência corrigida pelo método proposto é C4 D4 E4 G4 E4.

4.7–CONSIDERAÇÕES FINAIS

Esse capítulo descreveu a proposta de trabalho desta dissertação de mestrado. A arquitetura da rede e as formas de representação dos elementos musicais utilizadas seguem os exemplos propostos para a obtenção dos resultados. Também foi apresentado o método de avaliação e correção das melodias. O próximo capítulo apresenta os resultados obtidos de acordo com as especificações apresentadas.

5.1 – CONSIDERAÇÕES INICIAIS

Nesse capítulo são apresentados os resultados obtidos na dissertação de mestrado. Todos os treinamentos foram realizados com o programa MATLAB (versão 2006b) em um computador com as seguintes especificações: Processador Intel Core Duo 1,66 GHz, 667 MHz FSB, 2 MB L2 cache, 1GB DDR2.

Na sessão 5.2 são apresentados os resultados obtidos com o método proposto de iniciação dos pesos e configuração dos neurônios escondidos para otimizar o treinamento da rede neural LSTM em aplicações que envolvem aproximação de funções. A sessão 5.3 descreve como foram extraídos os contornos dos relevos naturais que são utilizados como inspiração da rede no processo de composição. Há também nessa sessão uma discussão sobre a influência dessa inspiração no treinamento da rede e nas composições finais obtidas. A sessão 5.4 apresenta as estratégias de composição musical desenvolvidas no trabalho e compara resultados obtidos pelas redes BPTT e LSTM. As abordagens desenvolvidas para a avaliação e correção das melodias obtidas pelas redes estão discutidas na sessão 5.5. As considerações finais desse capítulo estão descritas na sessão 5.6.

5.2 – EXPERIMENTOS COM O MÉTODO PROPOSTO DE INICIALIZAÇÃO DOS PESOS

Para testar e avaliar o método proposto, uma rede neural LSTM com quatro unidades de processamento (quatro blocos de memória com uma célula de memória em cada bloco) foi treinada para aproximar a função $d(x)$, como ilustrado na Figura 5.1.

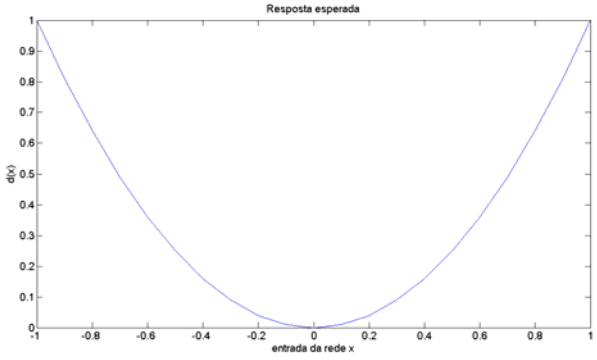


Figura 5.1: Resposta esperada para o primeiro experimento

Os valores iniciais dos pesos $w_{c_j^v}$ foram escolhidos aleatoriamente de uma distribuição uniforme entre -0,2 e 0,2. Figura 5.2 apresenta as saídas das células de memória $y^{c_j^v}(x)$ e a saída da rede $y^k(x)$ antes e depois do treinamento.

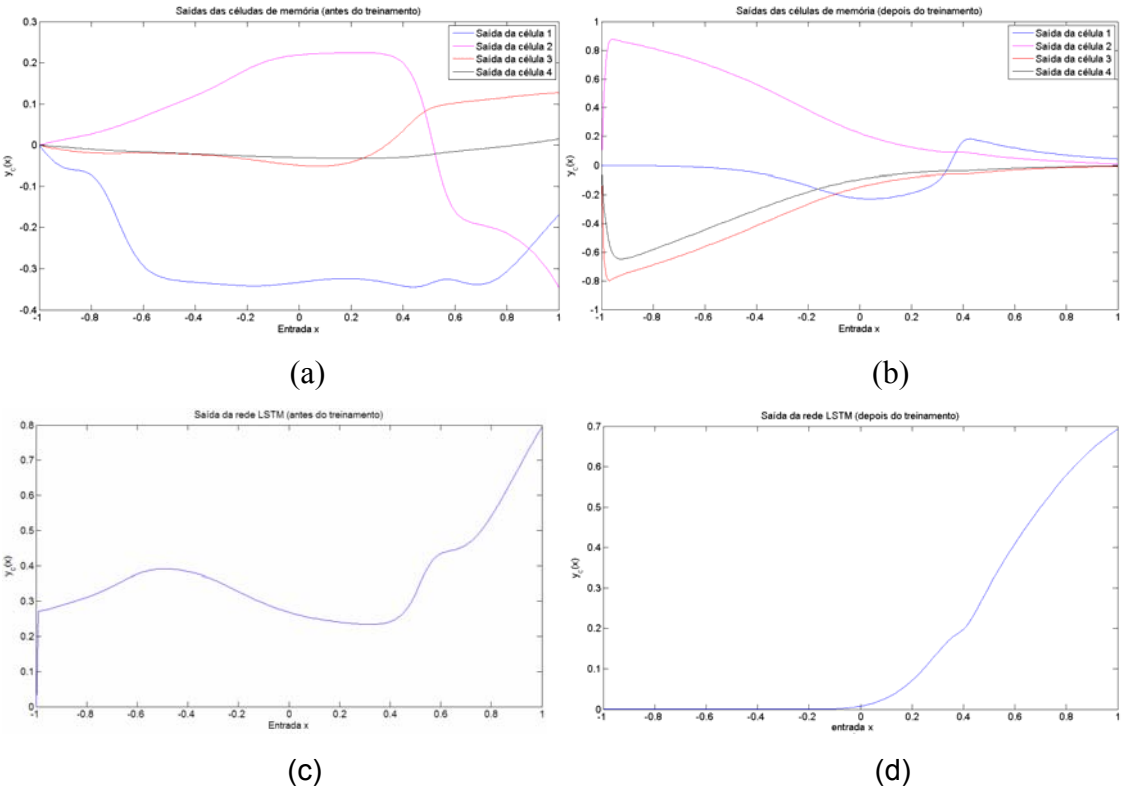


Figura 5.2: Saídas das células de memória com pesos iniciados aleatoriamente (a) antes do treinamento (b) depois do treinamento (c) Saída da rede antes do treinamento (d) Saída da rede depois do treinamento

No exemplo anterior, os pesos iniciais da rede foram selecionados com valores aleatórios pequenos. Essa é uma prática comum no treinamento de redes neurais. Entretanto, como observado no exemplo, os pesos precisam se mover de tal forma que a região de interesse seja dividida em pequenos intervalos.

Uma rede com pesos iniciais de acordo com o método proposto de iniciação, dado pela equação 4.11, foi treinada para aproximar a mesma função $d(x)$ descrita anteriormente. A Figura 5.3 apresenta, similarmente, as saídas das células de memória e a saída da rede antes e depois do treinamento.

O erro quadrático médio como função de tempo de treinamento é apresentado na Figura 5.4 para ambos os casos de treinamento, em que os pesos são selecionados aleatoriamente (linha sólida) e em que os pesos são selecionados de

acordo com o método proposto (linha pontilhada). Todos os outros parâmetros foram os mesmos nos dois treinamentos.

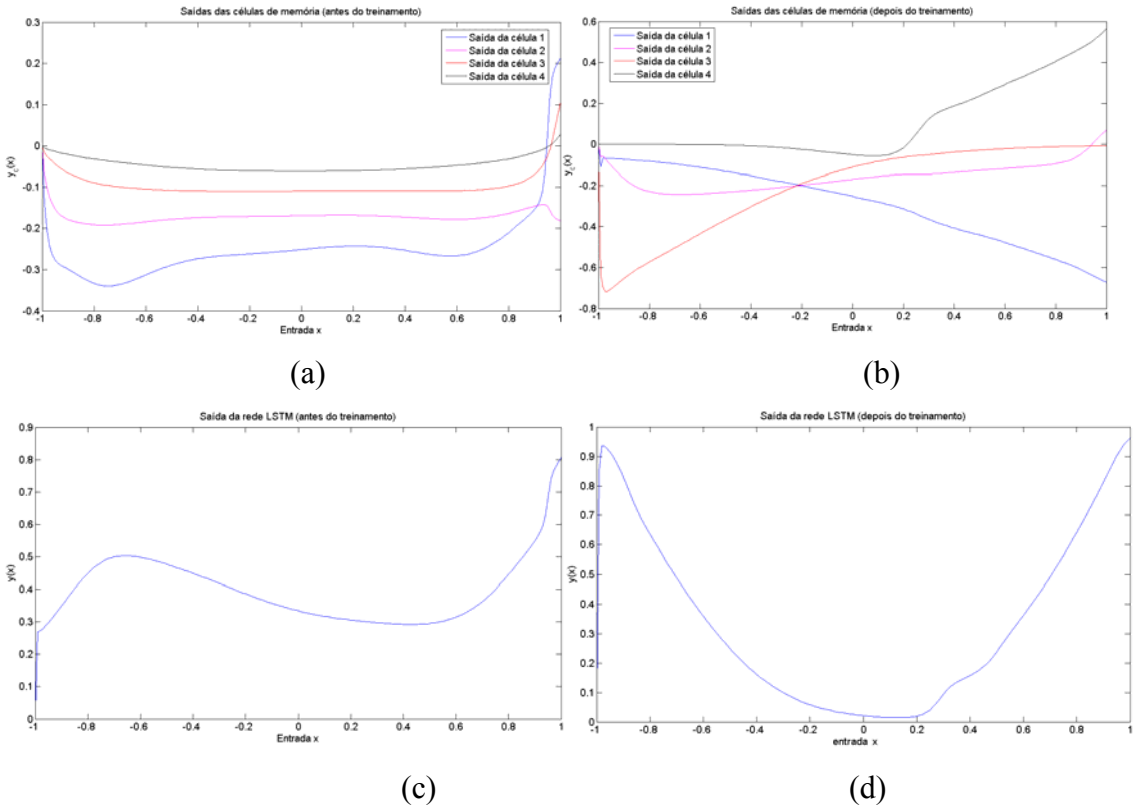


Figura 5.3: Saída das células de memória com iniciação de pesos de acordo com o método proposto (a) antes do treinamento (b) depois do treinamento (c) Saída da rede antes do treinamento (d) Saída da rede depois do treinamento

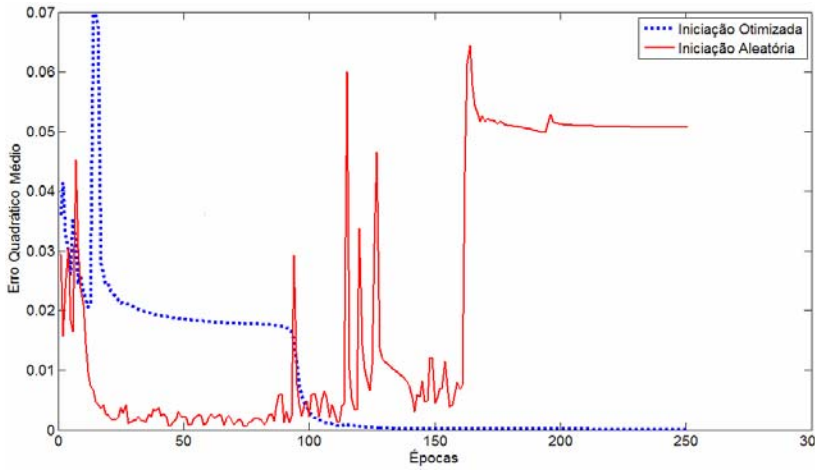
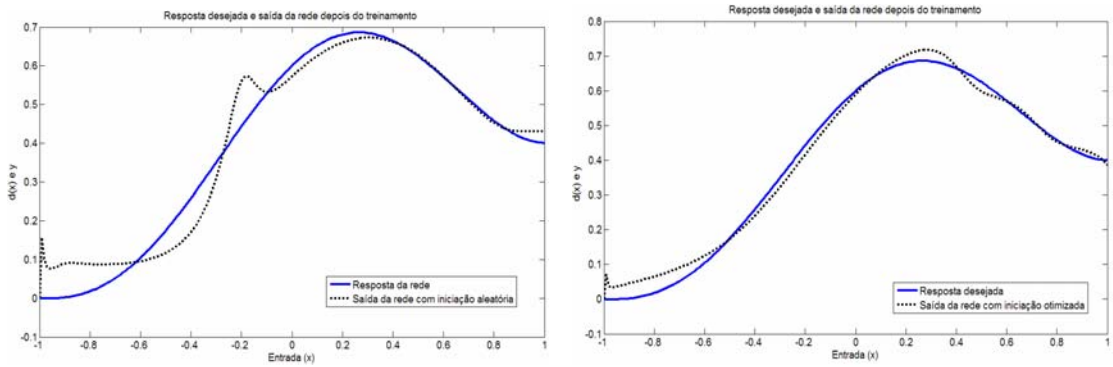


Figura 5.4: Erro quadrático médio para os dois casos de treinamento (iniciação aleatória e iniciação otimizada)

Para ilustrar outro exemplo, a rede neural LSTM foi treinada para aproximar a seguinte função não-linear:

$$d(x) = \sin(\pi x) + \cos(\pi x) + x^3 \tag{5.1}$$

A Figura 5.5 apresenta os resultados obtidos. Figura 5.5 (a) ilustra a função desejada (linha sólida) e a saída da rede (linha pontilhada) depois do treinamento com iniciação aleatória dos pesos. Figura 5.5 (b) ilustra a função desejada $d(x)$ (linha sólida) e a saída da rede (linha pontilhada) após a fase de treinamento com iniciação dos pesos otimizada de acordo com o método desenvolvido nesse trabalho. Finalmente, a Figura 5.6 apresenta o erro quadrático médio para os dois casos de treinamento.



(a) (b)

Figura 5.5: Função desejada d(x) e saída da rede após treinamento (a) com iniciação aleatória (b) com iniciação otimizada

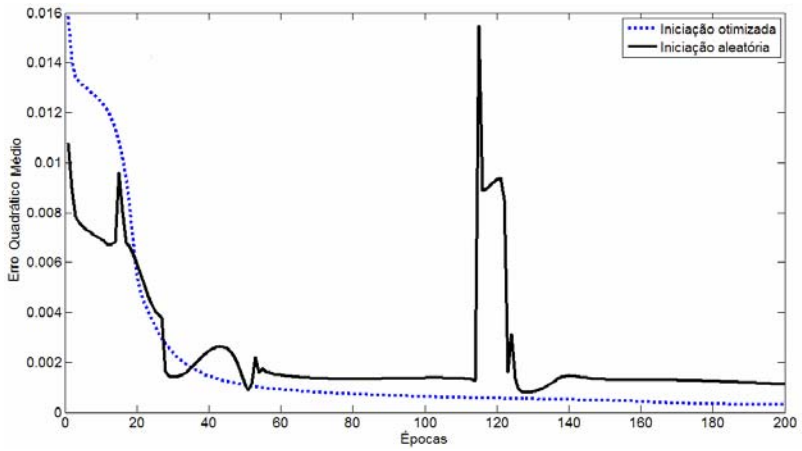


Figure 5.6: Erro quadrático médio para os dois casos de treinamento, com iniciação aleatória e otimizada

Com os resultados obtidos, foi observado que o método proposto oferece uma maior estabilidade para o treinamento da rede, fazendo com que a rede LSTM seja menos dependente das condições iniciais quando iniciada com pesos escolhidos de forma aleatória. Essa comparação pode ser observada nas Figuras 5.7 e 5.8. A mesma rede dos exemplos anteriores foi treinada três vezes, com 100 épocas cada, adotando valores iniciais aleatórios dos pesos. A Figura 5.7 apresenta o erro

quadrático médio em função do número de épocas de treinamento para cada um desses três treinamentos. É possível observar que a rede apresenta um comportamento instável.

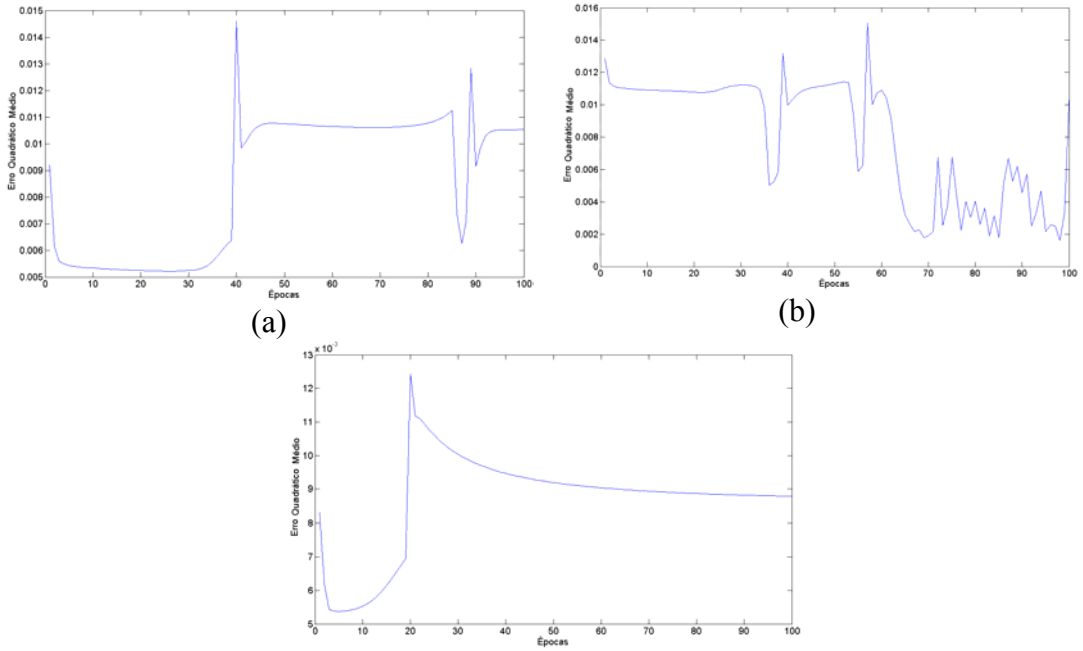


Figure 5.7: Erro quadrático médio com pesos iniciais aleatórios (a) primeiro treinamento (b) segundo treinamento (c) terceiro treinamento

O mesmo experimento foi realizado novamente, porém com o método desenvolvido para iniciação dos pesos. A Figura 5.8 apresenta o erro quadrático médio em função do número de épocas para os três treinamentos. Os resultados indicam um processo de treinamento mais estável.

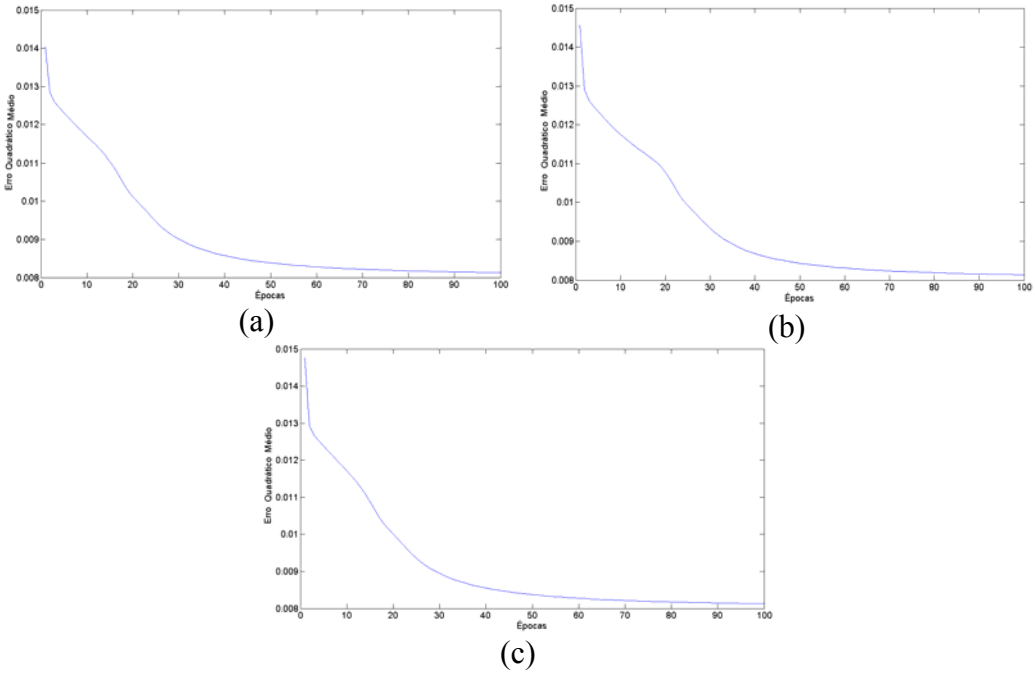


Figure 5.8: Erro quadrático médio com iniciação otimizada dos pesos

A iniciação proposta dos pesos foi usada para treinar uma rede com duas entradas para aproximar a superfície ilustrada na Figura 5.9. A função que descreve a superfície é a seguinte:

$$d(x,y) = x^2 + y^2 \tag{5.2}$$

O erro quadrático médio em função do número de épocas está apresentado na Figura 5.10 para o treinamento com pesos iniciados aleatoriamente em um intervalo de -0,2 a 0,2 (curva sólida); e para o treinamento com pesos iniciados com o método desenvolvido nesse trabalho (curva pontilhada).

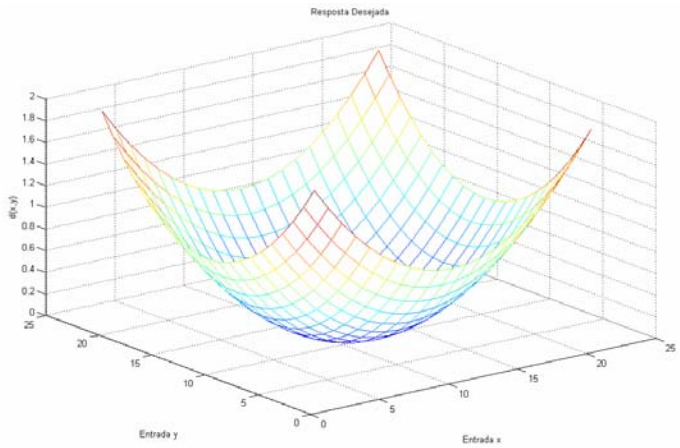


Figure 5.9: Função desejada 2-D

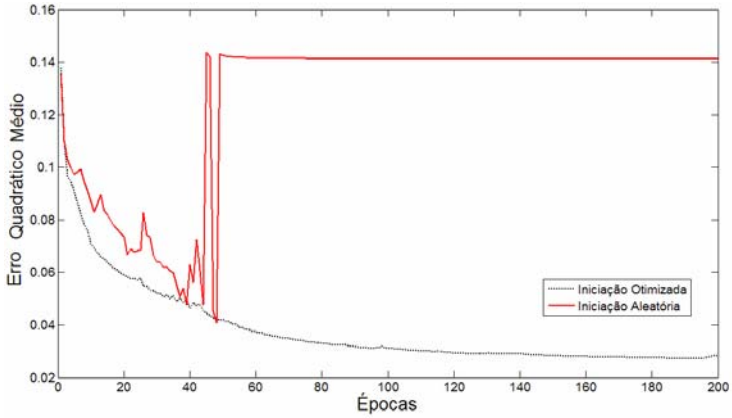


Figure 5.10: Curva de aprendizado para o treinamento da rede para aproximar d(x,y) descrita anteriormente.

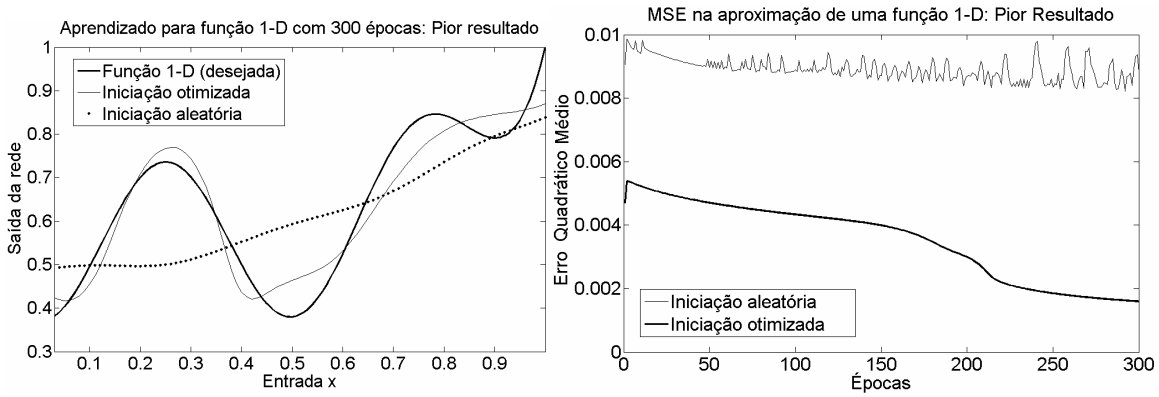
Em outro experimento, uma rede neural LSTM foi treinada para aproximar as funções 1-D e 2-D representadas pelas Figuras 4.11 e 4.12, respectivamente. A Tabela 5.1 apresenta os erros médios quadráticos para a iniciação aleatória, na qual os valores iniciais dos pesos $w_{c_j^y}$ foram escolhidos aleatoriamente de uma distribuição entre -0,2 e 0,2.

A Tabela 5.1 também apresenta os erros quadráticos médios para a iniciação otimizada com o método desenvolvido. Foram utilizadas várias configurações de blocos e células de memória com diferentes taxas de aprendizado. Cada configuração foi treinada por 300 épocas, 10 vezes. Para cada configuração é apresentado o melhor e pior caso. Erros com magnitude inferior a 10^{-4} foram considerados como zero. A otimização do treinamento da rede pode ser observada, uma vez que todos os casos de treinamento com iniciação otimizada dos pesos apresentam erros pequenos. Além disso, o método proposto reduz a diferença entre o melhor e pior caso, refletindo um comportamento mais estável. As configurações da rede próximas da estimativa proposta (5,1) apresentam os melhores resultados, sugerindo que o critério adotado pelo método proposto de estimação do número de neurônios escondidos é valido.

Tabela 5.1: Erro quadrático médio para o treinamento de aproximação de função 1-D utilizando iniciação aleatória e otimizada

H	M	Erro quadrático médio (Iniciação Aleatória)				Erro quadrático médio (Iniciação Otimizada)			
		$\alpha = 0.5$	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$
2	**1**	0.0076	0.0022	0.0017	0.0013	0.0030	0.0020	0.0017	0.0014
		0.0087	0.0048	0.0093	0.0093	0.0031	0.0020	0.0023	0.0014
3	**1**	0.0070	0.0020	0.0010	0.001	0.0021	0.0014	**0**	**0**
		0.0074	0.0097	0.0086	0.0091	0.0024	0.0020	0.0012	0.001
4	**1**	0.0060	0.0028	0.0023	0.0017	0.0018	0.0011	**0**	**0**
		0.0084	0.0086	0.0082	0.0089	0.0034	0.0026	0.0011	0.0010
5	**1**	**0.0031**	**0.0012**	**0.001**	**0**	**0.001**	**0**	**0**	**0**
		0.0076	0.0088	0.0070	0.0086	0.0030	0.001	0	0
1	**2**	0.0081	0.0087	0.0053	0.0037	0.0093	0.0081	0.0048	0.0032
		0.0097	0.0097	0.0098	0.0098	0.0094	0.0097	0.0049	0.0034
1	**3**	0.0065	0.0081	0.0089	0.002	0.0058	0.0044	0.0023	0.0018
		0.0097	0.0093	0.0094	0.0061	0.0069	0.0045	0.0024	0.0018
1	**4**	0.0068	0.0031	0.0020	0.0024	0.0052	0.0029	0.0018	0.0015
		0.0096	0.0090	0.0092	0.0098	0.0052	0.0029	0.0018	0.0015
1	**5**	0.0050	0.0028	0.0020	0.0016	0.0022	0.0021	0	0
		0.0086	0.0084	0.0097	0.0088	0.0034	0.0023	0.0016	0.0013

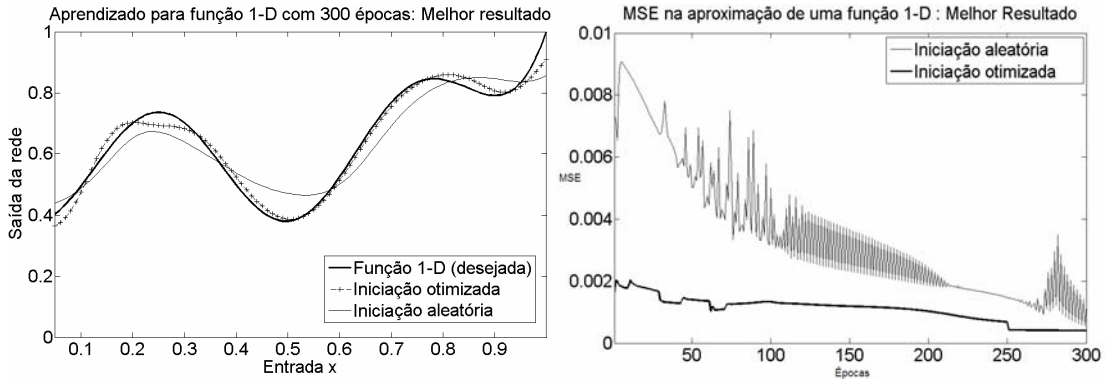
Como exemplos ilustrativos, são plotados a seguir as saídas da rede e o erro quadrático médio para o melhor e pior caso, utilizando iniciação aleatória e otimizada dos pesos e configuração proposta de cinco blocos de memória e uma célula de memória por bloco. Para o pior caso, a Figura 5.10 (a) mostra a função desejada 1-D $y^2(x) = \sin^2(\pi x) + \exp(x^5)$ e as saídas da rede depois do treinamento com a iniciação aleatória e otimizada e a Figura 5.10 (b) apresenta o erro quadrático médio em função do número de épocas para dos dois casos de treinamento.



(a) (b)

Figura 5.10 (a) Saídas da rede para o pior caso de aproximação de função 1-D com iniciação aleatória e otimizada (b) Erro quadrático médio do treinamento em (a)

Similarmente, a Figura 5.11 (a) apresenta a mesma função desejada 1-D (linha sólida) e as saídas da rede para o melhor caso novamente utilizando os dois tipos de iniciação. Figura 5.11 (b) ilustra o erro quadrático médio para os dois casos de treinamento.



(a) (b)

Figura 5.11 (a) Saídas da rede para o melhor caso de aproximação de função 1-D com iniciação aleatória e otimizada (b) Erro quadrático médio do treinamento em (a)

Uma rede neural foi treinada para aproximar a função 2-D ilustrada na Figura 4.12 com diferentes configurações, de maneira similar ao caso 1-D. Os

resultados indicaram que, como no treinamento do caso 1-D, quanto mais perto a configuração dos neurônios escondidos se encontra da ideal (nesse caso, 9 neurônios escondidos) menor é o erro quadrático médio. A Figura 5.12 mostra os erros médios quadráticos para o melhor caso de treinamento para iniciação aleatória e otimizada.

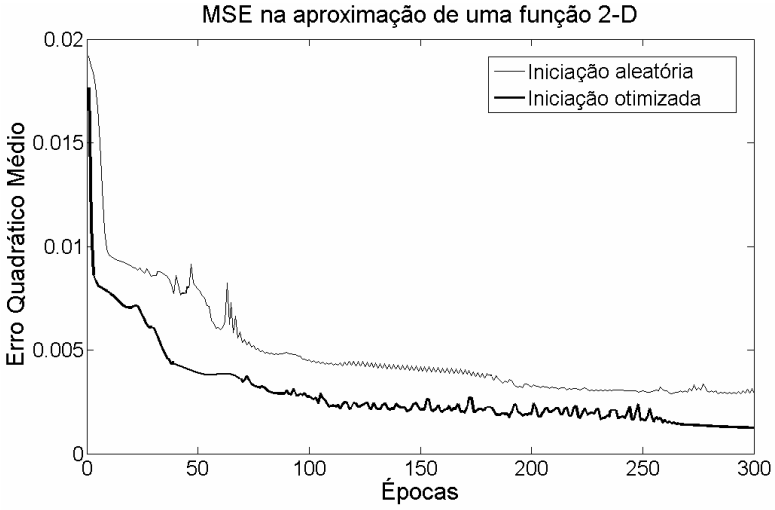


Figura 5.12: Erro quadrático médio para a função 2-D descrita na Figura 4.12

5.3 – OBTENÇAO E INFLUÊNCIA DA INSPIRAÇÃO

Como descrito anteriormente, na fase de treinamento da rede os compassos musicais devem pertencer a um mesmo estilo musical. Neste trabalho o conjunto de treinamento é formado por músicas brasileiras folclóricas e tradicionais. Esse estilo musical está mais detalhado na sessão 5.3.1, pois será utilizado nas próximas sessões.

Além dos conhecimentos necessários sobre música, os compositores geralmente consideram uma inspiração na composição de uma nova melodia. Esse trabalho propõe complementar o processo de composição com a adição de atributos externos, referenciada como inspiração da rede. Essa inspiração é representada pelos contornos de relevos naturais e tem por objetivo simular um processo de composição mais realístico e aprimorar a capacidade da rede na fase de aplicação. Portanto, a sessão 5.3.2 descreve como essa inspiração foi obtida e sua influência nas fases de treinamento e aplicação.

5.3.1 – Musicas brasileiras folclóricas e tradicionais

O folclore faz parte da cultura popular e é caracterizado pelo conjunto de mitos, crenças, tradições, festas populares, costumes que são passados de

geração em geração. A palavra folclore é composta por duas palavras provenientes do inglês: *folk* que significa povo e *lore* que significa conhecimento. Portanto, diz-se que o folclore expressa a sabedoria do povo[32].

As músicas folclóricas se caracterizam por serem simples, tonais, geralmente estarem contidas num intervalo de uma oitava, e por apresentarem certa monotonia e lentidão. A música folclórica está principalmente presente nas cantigas de roda, brincadeiras infantis, danças, cantos religiosos, etc. São exemplos de músicas folclóricas brasileiras:

- Cantigas de roda: Escravos de Jô, Atirei o Pau no Gato, Ciranda Cirandinha, O Cravo e a Rosa, Sapo Cururu, O Pobre e o Rico, Peixe Vivo.
- Cantigas de Ninar: Boi da Cara Preta.
- Modinhas: Casinha Pequenina.

A música tradicional brasileira está ligada à música folclórica. Também são músicas simples e monofônicas. Alguns autores consideram músicas folclóricas e tradicionais brasileiras como pertencentes a um mesmo estilo [ARAÚJO, 2007]. A música tradicional é caracterizada como a música própria de um povo de uma determinada região ou de um determinado contexto social. São exemplos de músicas tradicionais brasileiras: Mulher Rendeira, Oh! Minas Gerais, O Cravo e a Rosa, Onde está a Margarida.

As características das melodias folclóricas ou tradicionais brasileiras influenciaram na escolha desses dois estilos musicais (referenciados como um único estilo) para formação do conjunto de treinamento.

5.3.2 – Obtenção da Inspiração

A inspiração da rede é codificada a partir de imagens previamente selecionadas contendo relevos naturais. Para essa codificação foi utilizada uma combinação de filtros morfológicos, como por exemplo, operações de erosão e dilatação.

A morfologia matemática tem aplicação em diversas áreas de processamento e análises de imagens, como por exemplo, segmentação, realce, detecção de bordas, filtragem, entre outras [FILHO e NETO, 1999] e se caracteriza por um conjunto de operações que são aplicadas em uma imagem (representada por

[32] Disponível em http://ifolclore.vilabol.uol.com.br. Acesso: 1 de Maio de 2008.

conjunto de *pixels*) [PRATT, 1991]. A base da morfologia matemática é a teoria de conjuntos, caracterizada pela extração de informações relativas à geometria e à topologia de uma imagem desconhecida através de transformações de outra imagem bem definida, denominado elemento estruturante. Em imagens binárias, cada elemento do conjunto é um vetor 2-D representando as coordenadas (x,y) do pixel.

Sejam A e B conjuntos de membros do espaço inteiro bidimensional Z^2, com componentes $a=(a_1,a_2)$ e $b=(b_1,b_2)$, respectivamente. A translação de A por $x=(x_1,x_2)$, denotada por $(A)_x$, é dada por:

$$(A)_x=\{c \mid c=a+x, \text{ para } \quad a\in A\} \tag{5.3}$$

A reflexão de $B\left(\hat{B}\right)$, é caracterizada por:

$$\hat{B}=\{x \mid x=-b, \text{ para } b\in B\} \tag{5.4}$$

O complemento do conjunto A é definido como:

$$A^c=\{x \mid x\notin A\} \tag{5.5}$$

A diferença entre dois conjuntos A e B, $(A-B)$, é dada por:

$$A-B=\{x \mid x\in A, x\notin B\}=A\cap B^c \tag{5.6}$$

A dilatação entre dois conjuntos A e B, indicada por $A\oplus B$, é dada por:

$$A\oplus B=\left\{x \mid \left[(\hat{B})_x\cap A\right]\subseteq A\right\} \tag{5.7}$$

Assim, é possível definir o processo de dilatação pela reflexão de B sobre sua origem e posteriormente pelo seu deslocamento de x. A dilatação de A e B é, portanto, o conjunto dos x deslocamentos para os quais a interseção com A esteja contida em A. O elemento estruturante está representado pelo conjunto B.

A erosão entre A e B, indicada por A e B, é definida como:

$$A \text{ e } B=\{x \mid (B)_x\subseteq A\} \tag{5.8}$$

Portanto, a erosão consiste no conjunto de pontos x, de tal forma que B, uma vez translado de x, esteja contido em A.

Exemplos de dilatação e erosão estão ilustrados na Figura 5.13 (a).

Outras duas operações morfológicas importantes são a abertura e o fechamento. Geralmente a abertura[33] é usada na suavização do contorno de uma imagem, na eliminação de objetos pequenos e na quebra de extremidades estreitas. A abertura de um conjunto A por B, denotada como $A \circ B$, representa a erosão de A por B e em seguida a dilatação do resultado por B:

$$A \circ B = (A \text{ e } B) \oplus B \tag{5.9}$$

Por outro lado, o fechamento[34] do conjunto A pelo elemento estruturante B, denotado por $A \bullet B$, é definido como a dilatação de A por B seguida da erosão do resultado obtido por B:

$$A \bullet B = (A \oplus B) \text{ e } B \tag{5.10}$$

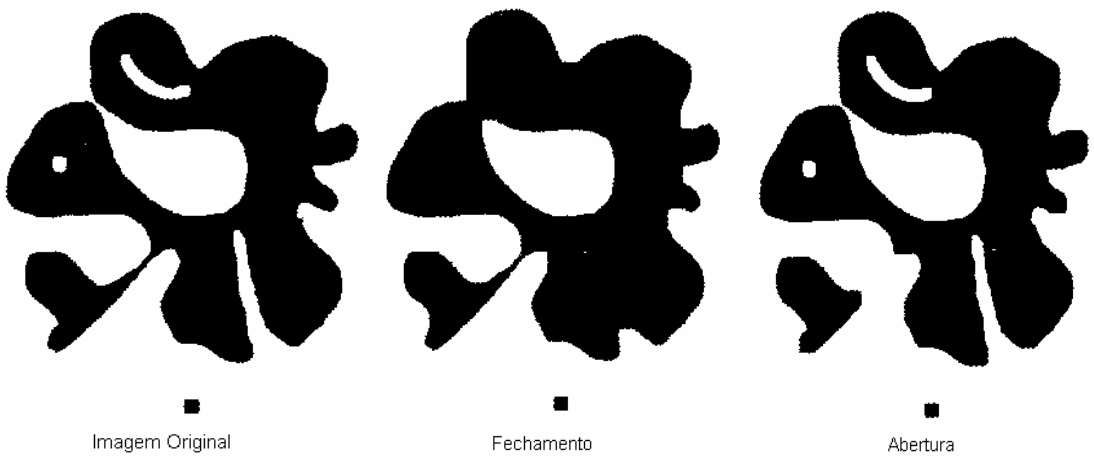
O fechamento também suaviza contornos de objetos, elimina buracos pequenos e une espaços pequenos entre objetos.

Os efeitos da abertura e fechamento podem ser notados na Figura 5.13(b).

[33] Propriedades da abertura:
- i. $A \circ B$ é uma subimagem de A.
- ii. Se C é uma subimagem de D, portanto $C \circ B$ é uma subimagem de $D \circ B$.
- iii. $(A \circ B) \circ B = A \circ B$

[34] Propriedades do fechamento:
- i. A é uma subimagem de $A \bullet B$.
- ii. Se C é uma subimagem de D, então $C \bullet B$ é uma subimagem de $D \bullet B$.
- iii. $(A \bullet B) \bullet B = A \bullet B$

(a)



(b)

Figura 5.13: Exemplos de (a) dilatação e erosão (b) abertura e fechamento [PRATT, 1991]

A idéia geral do algoritmo para a extração do contorno da inspiração utilizado nas composições emprega a morfologia matemática e pode ser resumido nos seguintes passos:

1. Obter imagem com contornos naturais;
2. Transformar para imagem binária;
3. Criar elemento estruturante;
4. Aplicar erosão da imagem binária com elemento estruturante;
5. Computar diferença entre imagem binária com o resultado da erosão;
6. Obter vetor de *pixels* do contorno da imagem.

A Figura 5.14 ilustra os seis passos do algoritmo para obtenção dos contornos dos relevos naturais. As Figuras 5.15 e 5.16 apresentam outros exemplos de imagens utilizadas como inspiração e o contorno obtido.

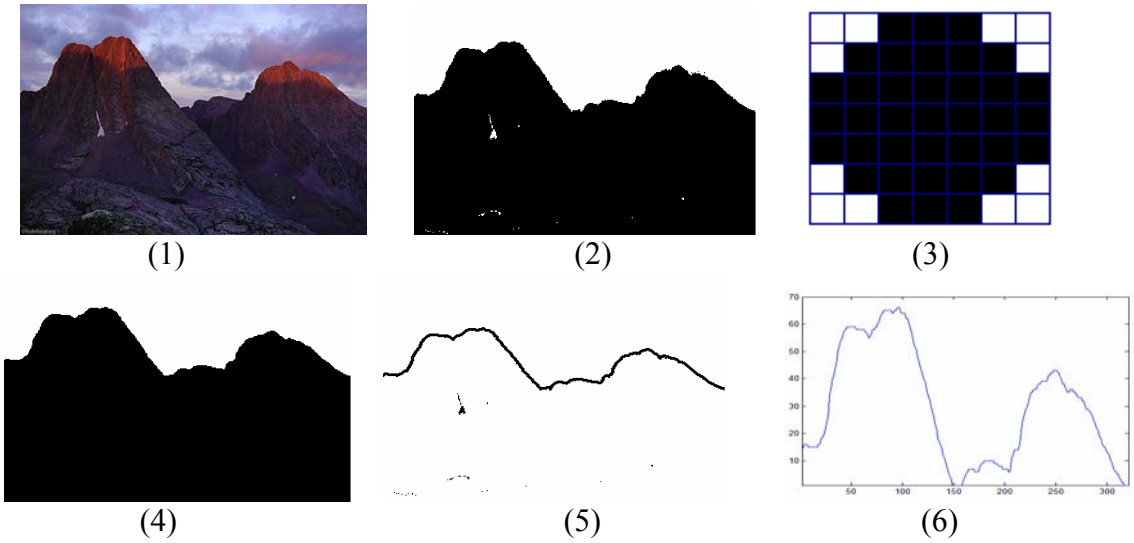


Figura 5.14: Passos para extração do contorno dos relevos naturais

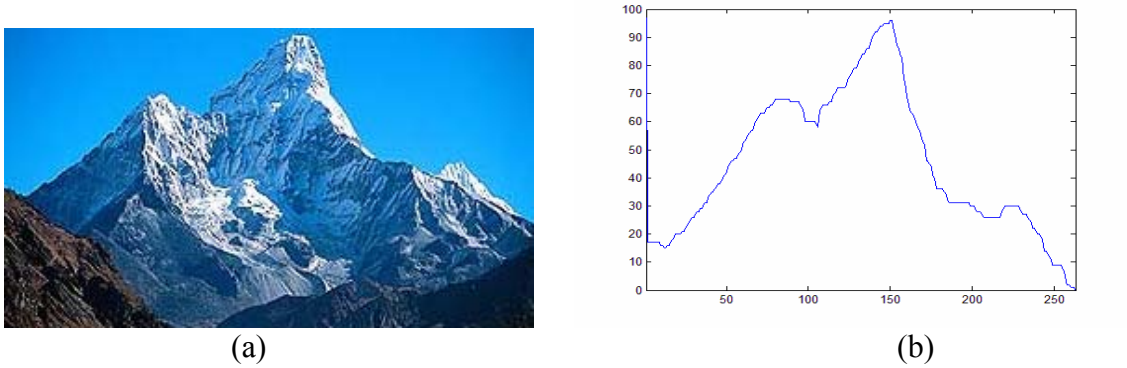
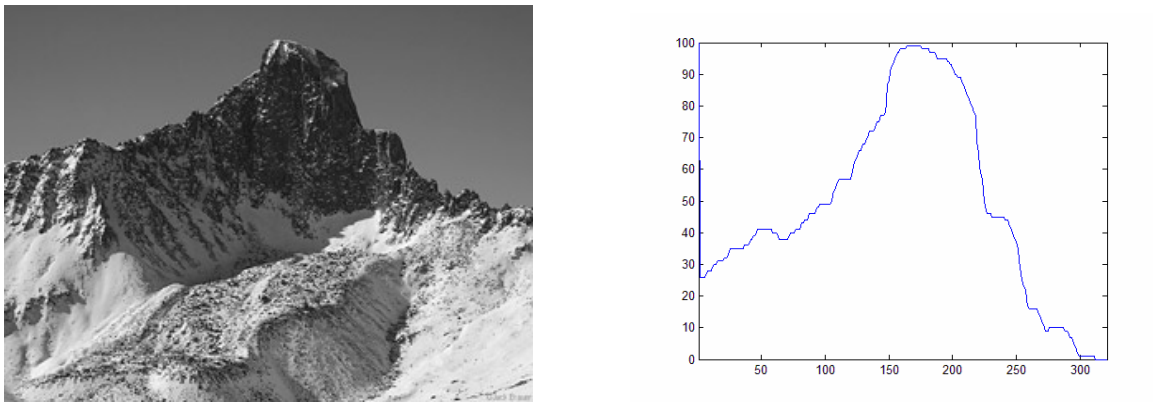


Figura 5.15: Imagem original (a) Extração do contorno (b)



(a) (b)

Figura 5.16: Imagem original (a) Extração do contorno (b)

O contorno então é convertido para uma seqüência de números inteiros que representam as notas musicais (Figura 5.17). Essa seqüência é normalizada para o treinamento da rede.

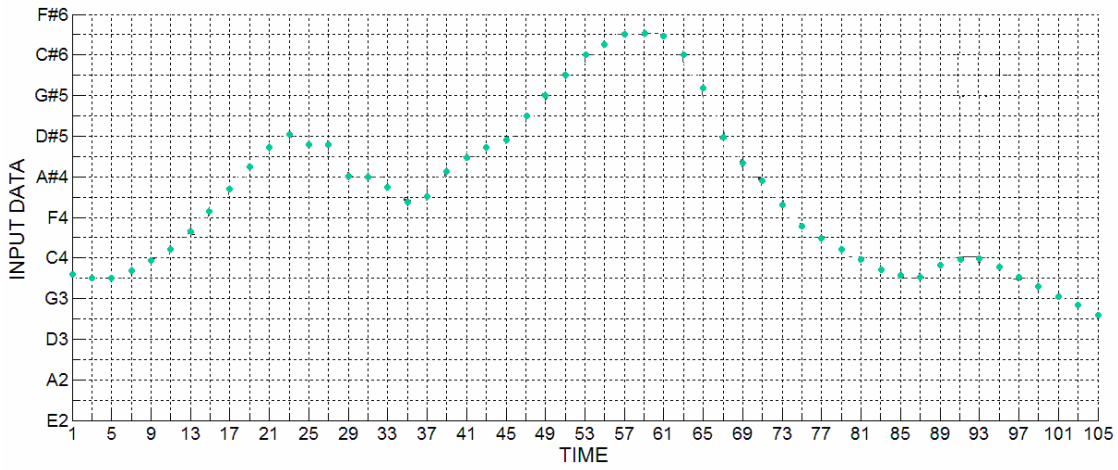


Figura 5.17: Conversão do contorno da Figura 5.15 para seqüência de notas musicais

5.3.3 – Influência da inspiração

O objetivo dessa sessão é investigar a influência da inspiração nas composições musicais e descrever o comportamento da rede na presença dessa inspiração. O termo “inspiração” diz respeito a um vetor que representará as notas extraídas do contorno de imagens com relevos naturais.

A melodia utilizada no treinamento foi “Escravos de Jô.” Suas notas estão ilustradas na Figura 5.18. Para melhor visualização, as melodias serão representadas em uma matriz, como seqüências das notas no tempo, coluna por coluna, como na equação 5.11.

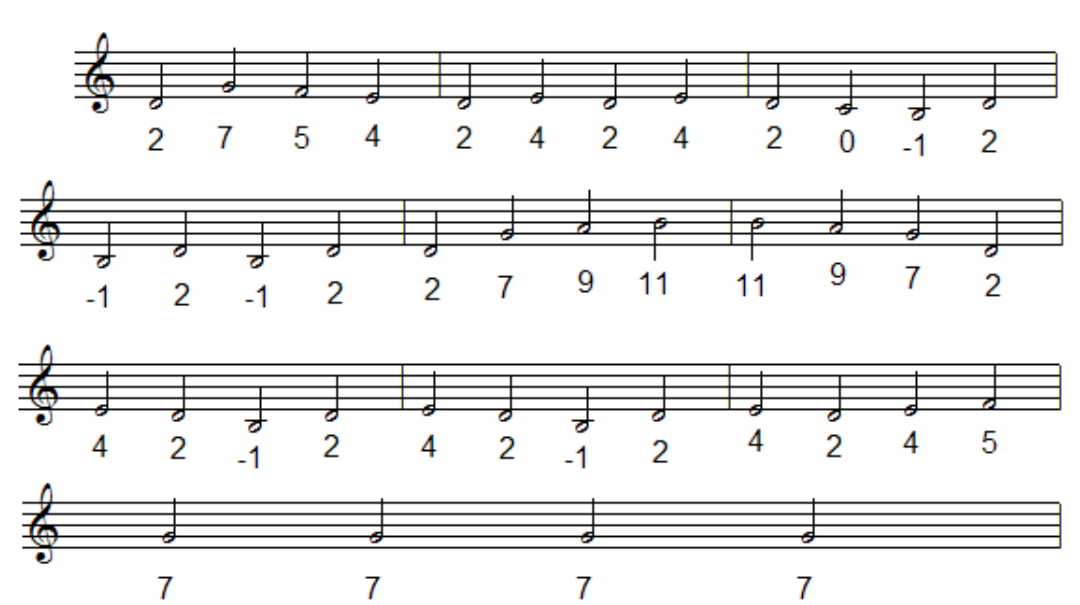


Figura 5.18: Partitura da melodia Escravos de Jô

$$EscJó = \begin{bmatrix} 2 & 2 & 2 & -1 & 2 & 11 & 4 & 4 & 4 & 7 \\ 7 & 4 & 0 & 2 & 7 & 9 & 2 & 2 & 2 & 7 \\ 5 & 2 & -1 & -1 & 9 & 7 & -1 & -1 & 4 & 7 \\ 4 & 4 & 2 & 2 & 11 & 2 & 2 & 2 & 5 & 7 \end{bmatrix} \tag{5.11}$$

Sem a utilização da inspiração, a entrada da rede é caracterizada pelas notas da melodia. Uma época de treinamento consiste em apresentar todas as notas da matriz $EscJó$. Cada padrão de entrada é representado por quatro notas da matriz seqüencialmente selecionadas, coluna por coluna. Cada coluna representa um conjunto de quatro notas, referenciada nesse estudo como compasso. A tarefa da rede é aprender as próximas quatro notas ou o próximo compasso da coluna seguinte. Quando a rede atingir o erro determinado (nesse caso, $0,03$ como a soma dos erros quadráticos de todos os padrões) o treinamento é terminado. Para esse treinamento, foi utilizada uma rede LSTM com quatro blocos de memória com uma célula de memória por bloco. Os padrões de entradas foram normalizados para o intervalo $[0,1]$. A taxa de aprendizado utilizada foi $a = 0,04$. Foram necessárias 62910 épocas de treinamento, num tempo de duração de 4432 segundos (especificar máquina).

Na fase de aplicação, a primeira coluna (primeiro compasso) de notas são dadas para a rede e suas saídas são realimentadas para a geração das próximas notas até que a melodia seja composta. Portanto, a matriz $EscJó(0)$ representa as notas compostas pela rede após o treinamento, sem nenhuma informação adicional de inspiração.

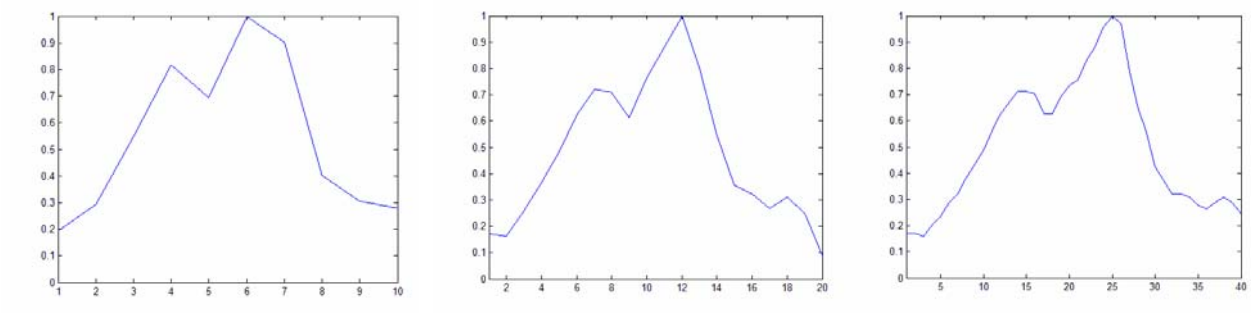
$$EscJó(0) = \begin{bmatrix} 2 & 2 & 1 & 0 & 1 & 11 & 6 & 5 & 7 & 6 \\ 7 & 4 & 1 & 1 & 6 & 9 & 4 & 1 & 3 & 7 \\ 5 & 2 & -1 & -1 & 7 & 9 & -1 & -1 & 5 & 4 \\ 4 & 4 & 2 & 2 & 10 & 3 & 3 & 4 & 4 & 7 \end{bmatrix} \quad (5.12)$$

As notas circuladas de vermelho representam as diferenças em relação à melodia de treinamento. É possível notar que, apesar das diferenças, a melodia composta pela rede ainda apresenta algumas relações entre alturas, encontradas na melodia de treinamento. Por exemplo, as duas notas diferentes ao quarto compasso estão um semitom abaixo em relação às mesmas notas da melodia de treinamento.

Em seguida, os treinamentos foram realizados com a informação complementar da inspiração. A melodia de treinamento é a mesma e também foi utilizada mesma taxa de aprendizado e mesmo valor mínimo do erro.

A inspiração utilizada no treinamento é representada pelo contorno da Figura 5.15, com 263 notas de inspiração. Para o treinamento as notas são selecionadas de acordo com a quantidade a ser utilizada nos padrões de treinamento, como descrito a seguir.

Com a utilização de apenas uma nota de inspiração na composição, cada padrão de entrada possui cinco elementos (quatro elementos representando as notas e um para a nota da inspiração). São necessárias 10 notas de inspiração no total (Figura 5.19 (a)), representadas na matriz $insp(1)$ da equação 5.13. Cada conjunto de quatro notas da melodia é acompanhado de uma nota da inspiração; e a tarefa da rede continua sendo a aprendizagem das próximas quatro notas. Para esse treinamento, foram utilizados cinco blocos de memória com uma célula cada. Foram necessárias 53.371 épocas de treinamento, o que resultou em 7752 segundos.



(a) (b) (c)

Figura 5.19: Inspiração usada no treinamento: (a) 1 nota (b) 2 notas e (c) 4 notas

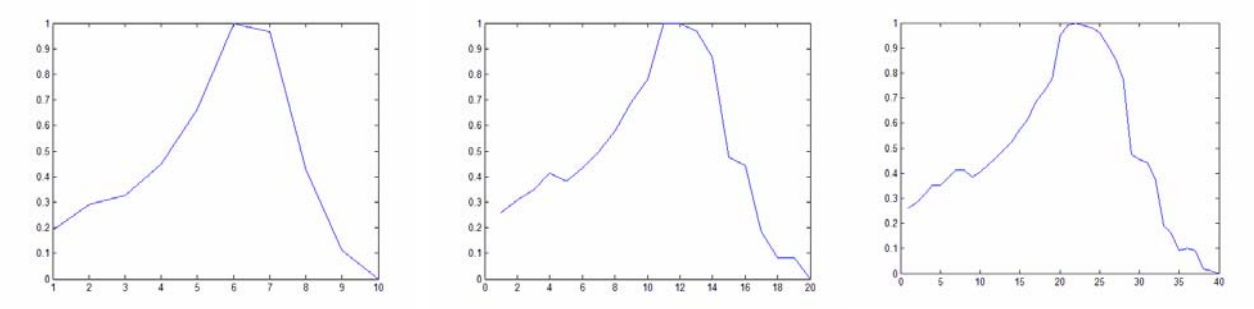
$$insp(1) = \begin{bmatrix} -13 \\ -5 \\ 16 \\ 38 \\ 28 \\ 53 \\ 45 \\ 4 \\ -4 \\ -6 \end{bmatrix} \quad (5.13) \quad insp(2) = \begin{bmatrix} -13 & -14 \\ -5 & 5 \\ 16 & 29 \\ 38 & 37 \\ 28 & 42 \\ 53 & 64 \\ 45 & 22 \\ 4 & 1 \\ -4 & 0 \\ -6 & -21 \end{bmatrix} \quad (5.14) \quad insp(4) = \begin{bmatrix} -13 & -13 & -14 & -10 \\ -7 & -2 & 1 & 7 \\ 12 & 17 & 24 & 30 \\ 34 & 38 & 38 & 37 \\ 30 & 30 & 36 & 40 \\ 42 & 49 & 54 & 61 \\ 65 & 62 & 45 & 32 \\ 23 & 11 & 6 & 1 \\ 1 & 0 & -3 & -4 \\ -2 & 0 & -2 & -6 \end{bmatrix} \quad (5.15)$$

Na fase de aplicação, o primeiro compasso é dado para a rede juntamente com a primeira nota da inspiração. O próximo padrão de entrada é formado pelas quatro notas geradas pela rede e pela nota seguinte da inspiração. O procedimento continua até que a melodia seja composta. Nesse caso, a inspiração na fase de composição é exatamente a mesma do treinamento. A matriz $EscJó(1)$ representa as notas, em valores inteiros, compostas pela rede.

$$escJó(1) = \begin{bmatrix} 2 & 2 & 1 & 0 & 2 & 11 & 4 & 3 & 6 & 8 \\ 7 & 4 & 0 & 1 & 6 & 9 & 3 & 2 & 5 & 9 \\ 5 & 2 & -1 & -1 & 7 & 8 & -1 & 0 & 6 & 5 \\ 4 & 4 & 1 & 2 & 10 & 3 & 2 & 3 & 6 & 6 \end{bmatrix} \quad (5.16)$$

A melodia composta pela rede com a informação de apenas uma nota na fase de treinamento e composição possui novamente várias notas diferentes em relação a melodia original. Ainda, $escJó(1)$ está bem parecida com $escJó(0)$ nos primeiros seis compassos. As notas a partir do sétimo compasso possuem notas mais agudas em relação à $escJó(0)$ por causa da influência das notas provenientes da inspiração.

Em seguida a fase de aplicação é alterada pela inclusão de uma nova inspiração, semelhante à usada no treinamento, a fim de obter melodias ainda mais diferentes. Para tanto, o treinamento foi executado da mesma forma como descrito anteriormente. Na fase de aplicação, a inspiração é caracterizada pelos dados provenientes do contorno da Figura 5.16. Assim, a composição da melodia $escJó(1.1)$ foi influenciada pelas notas ilustradas na Figura 5.20 (a), representada na matriz $insp(1)$ da equação 5.17.



(a) (b) (c)

Figura 5.20: Inspiração usada na composição: (a) 1 nota (b) 2 notas e (c) 4 notas

$$insp(1)=\begin{bmatrix} -4 \\ 5 \\ 8 \\ 19 \\ 38 \\ 68 \\ 65 \\ 17 \\ -11 \\ -21 \end{bmatrix} \quad (5.17) \qquad insp(2)=\begin{bmatrix} -4 & 1 \\ 5 & 11 \\ 8 & 13 \\ 19 & 27 \\ 38 & 47 \\ 68 & 68 \\ 65 & 55 \\ 17 & 14 \\ -11 & -21 \\ -21 & -29 \end{bmatrix} \quad (5.18) \qquad insp(4)=\begin{bmatrix} -4 & -2 & 1 & 5 \\ 5 & 8 & 11 & 11 \\ 8 & 10 & 13 & 16 \\ 19 & 22 & 27 & 31 \\ 38 & 42 & 47 & 64 \\ 68 & 69 & 68 & 67 \\ 65 & 60 & 55 & 47 \\ 17 & 15 & 14 & 7 \\ -11 & -14 & -21 & -20 \\ -21 & -28 & -29 & -30 \end{bmatrix} \quad (5.19)$$

Mesmo com a utilização de inspirações semelhantes na fase de treinamento e composição, é possível observar que $escJó(1.1)$ está mais distante da melodia de treinamento $escJó$, apesar de ainda manter algumas relações entre as notas. Em relação à $escJó(1)$, $escJó(1.1)$ possui notas mais agudas, principalmente nos últimos compassos por causa das notas da inspiração usada (Figura 5.20 (a)), que, comparadas com as notas da inspiração usada no treinamento (Figura 5.19 (a)), apresentam valores inteiros maiores.

$$escJó(1.1)=\begin{bmatrix} 2 & 2 & 1 & 0 & 0 & 8 & 9 & 4 & 11 & 10 \\ 7 & 4 & 0 & 1 & 4 & 7 & 7 & 5 & 10 & 11 \\ 5 & 2 & -1 & -1 & 1 & 8 & 3 & 4 & 11 & 10 \\ 4 & 4 & 1 & 1 & 6 & 6 & 4 & 9 & 10 & 10 \end{bmatrix} \quad (5.20)$$

Num segundo experimento, foram utilizadas duas notas da inspiração nos processos de treinamento e composição. O treinamento é realizado da mesma forma, com cada padrão de entrada sendo representado por seis elementos (quatro notas e duas notas da inspiração). São necessárias, portanto, 20 notas de inspiração para o treinamento da melodia (Figura 5.19 (b)), representada pela matriz $insp(2)$ da

equação 5.14. Cada conjunto de quatro notas da melodia é acompanhado de duas notas da inspiração; e a tarefa da rede continua sendo a aprendizagem das próximas quatro notas. Para esse treinamento, foram utilizados seis blocos de memória com uma célula cada. Foram necessárias 68.332 épocas de treinamento, o que resultou em 11732 segundos de tempo de processamento.

Na fase de aplicação, o primeiro compasso é dado para a rede juntamente com as duas primeiras notas da inspiração; e a rede utiliza as próprias saídas para a composição das notas seguintes. A matriz $EscJó(2)$ representa as notas compostas pela rede.

$$escJó(2) = \begin{bmatrix} 2 & 2 & 2 & 0 & 2 & 11 & 4 & 3 & 4 & 6 \\ 7 & 4 & 0 & 2 & 7 & 9 & 2 & 1 & 3 & 5 \\ 5 & 2 & -1 & -1 & 9 & 7 & -1 & -1 & 3 & 6 \\ 4 & 4 & 2 & 2 & 11 & 2 & 1 & 1 & 4 & 6 \end{bmatrix} \quad (5.21)$$

É possível observar que a melodia $escJó(2)$ está semelhante a melodia usada no treinamento, com apenas 12 notas diferentes, que, de forma geral, estão um semitom abaixo das respectivas notas da melodia $escJó$. As duas notas da inspiração contribuíram para um aprendizado com notas mais fiéis às notas usadas como padrões de treinamento. De maneira semelhante como no experimento anterior, a fase de aplicação foi alterada com a utilização de uma inspiração diferente, representada na Figura 5.20 (b), porém com a utilização de 20 notas (matriz $insp(2)$ da equação 5.18). A melodia resultante está representada na matriz $escJó(2.2)$.

$$escJó(2.2) = \begin{bmatrix} 2 & 3 & 3 & 1 & 5 & 11 & 3 & 3 & 6 & 8 \\ 7 & 4 & 1 & 4 & 8 & 9 & 1 & 1 & 4 & 7 \\ 5 & 2 & -1 & -1 & 10 & 6 & -1 & -1 & 5 & 8 \\ 4 & 5 & 3 & 2 & 11 & 1 & 1 & 2 & 6 & 8 \end{bmatrix} \quad (5.22)$$

A melodia $escJó(2.2)$ difere significamente da melodia $escJó(1.1)$ por causa da contribuição maior da inspiração, porém ainda guarda passagens de notas presentes na melodia de treinamento.

Num terceiro experimento, foram utilizadas quatro notas da inspiração nos processos de treinamento e composição. O treinamento é como anteriormente. Portanto, cada padrão de entrada é representado por oito elementos (quatro notas e quatro notas da inspiração). Esse experimento utilizou 40 notas de inspiração para o treinamento e composição da melodia (Figura 5.19 (c)), representada pela matriz $insp(4)$ da equação 5.15. Cada conjunto de quatro notas da melodia é acompanhado

de quatro notas da inspiração; e a tarefa deve aprender as próximas quatro notas da melodia de treinamento. Para esse treinamento, foram utilizados oito blocos de memória com uma célula cada. Foram necessárias 2409 épocas de treinamento, o que resultou em 594 segundos de duração.

Na fase de aplicação acontece de forma semelhante, ou seja, o primeiro compasso da melodia de treinamento é dado para a rede juntamente com as quatro primeiras notas da inspiração; e a rede utiliza as próprias saídas para a composição das notas seguintes. A matriz $EscJó(4)$ representa as notas compostas pela rede.

$$EscJó(4) = \begin{bmatrix} 2 & 2 & 2 & -1 & 2 & 11 & 4 & 4 & 4 & 7 \\ 7 & 4 & 0 & 2 & 7 & 9 & 2 & 2 & 2 & 6 \\ 5 & 2 & -1 & -1 & 9 & 7 & -1 & 0 & 3 & 7 \\ 4 & 4 & 2 & 2 & 11 & 2 & 2 & 2 & 5 & 7 \end{bmatrix} \quad (5.23)$$

É possível observar que a melodia composta pela rede representa quase que totalmente a melodia de treinamento. Isso foi possível por causa da redundância dos dados. Haykin [1999] afirma que uma das heurísticas para melhorar o desempenho do algoritmo de retropropagação é maximar o conteúdo da informação. Segundo Haykin:

> Como regra geral, todo exemplo de treinamento apresentado ao algoritmo de retropropagação deve ser escolhido de forma que seu conteúdo de informação seja o maior possível para a tarefa considerada. Dois modos de alcançar este objetivo são:
> - O uso de um exemplo que resulte no maior erro de treinamento.
> - O uso de um exemplo que seja radicalmente diferente de todos os outros usados anteriormente [HAYKIN, 1999, p.205].

O uso de exemplos de treinamento que sejam diferentes dos exemplos anteriores é possível com a inclusão de mais notas de inspiração.

Como nos exemplos anteriores, a fase de aplicação foi alterada com a utilização de uma inspiração diferente, representada na Figura 5.20 (c), porém com a utilização de 40 notas (matriz $insp(4)$ da equação 5.19). A melodia resultante está representada na matriz $escJó(4.4)$.

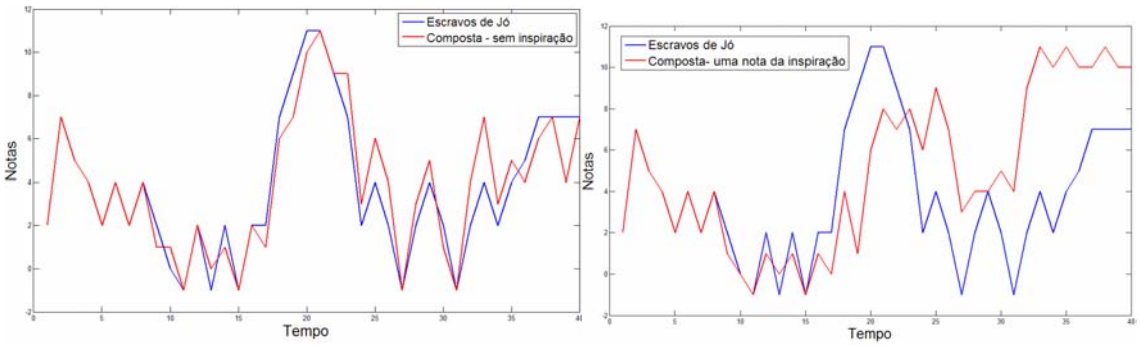
$$escJó(4.4) = \begin{bmatrix} 2 & 4 & 6 & 0 & 1 & 11 & 11 & 11 & 11 & 11 \\ 7 & 5 & 1 & 4 & 10 & 11 & 11 & 11 & 11 & 11 \\ 5 & 3 & -1 & -1 & 8 & 11 & 11 & 11 & 11 & 11 \\ 4 & 6 & 3 & 2 & 11 & 11 & 11 & 11 & 11 & 11 \end{bmatrix} \quad (5.24)$$

Observa-se que a melodia composta com a utilização de quatro notas de uma inspiração semelhante, porém diferente à do treinamento ficou saturada a

partir sexto compasso. Isso pode ter ocorrido pela presença de várias notas muito agudas e depois muito graves na inspiração.

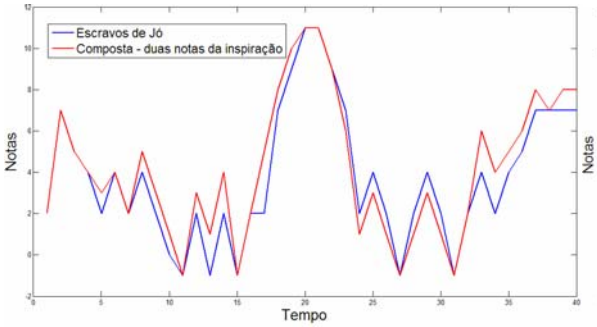
Portanto, quando a mesma inspiração é utilizada nas fases de treinamento e aplicação, quanto mais notas de inspiração utilizadas, mais próxima ficará a melodia final com a melodia de treinamento. Quando inspirações diferentes são aplicadas nas fases de treinamento e aplicação, a utilização de mais notas de inspiração tende a gerar melodias menos similares às melodias do treinamento.

Para uma melhor ilustração, a melodia “Escravos de Jó” e as melodias compostas pela rede com a utilização da inspiração da Figura 5.20 foram transformadas em vetores 2-D e são plotadas na Figuras 5.21. A Figura 5.21 (a) apresenta a melodia “Escravos de Jó” e a melodia obtida pela rede com treinamento e fase de aplicação sem nenhuma informação de inspiração. Na Figura 5.21 (b) tem-se “Escravos de Jó” e $escJó(1.1)$, ou seja, a melodia composta pela rede com a informação de uma nota da inspiração. Na Figura 5.21 (c) tem-se “Escravos de Jó” e $escJó(2.2)$, que é a melodia gerada com a utilização de duas notas da inspiração. A Figura 5.21 (d) apresenta “Escravos de Jó” e $escJó(4.4)$. Na Figura 5.21 (e) tem-se $escJó(1.1)$ e $escJó(2.2)$. Finalmente, na Figura 5.21 (f) tem-se $escJó(2.2)$ e $escJó(4.4)$.

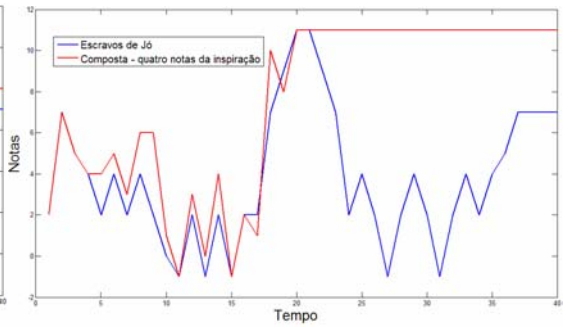


(a)

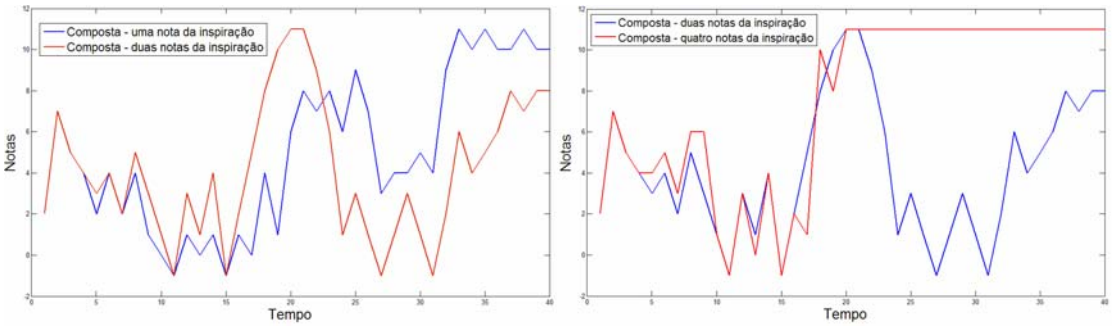
(b)



(c)



(d)



(e) (f)

Figura 5.21: Melodias geradas pela rede com inspiração semelhante na fase de aplicação

5.4 – RESULTADOS DE COMPOSIÇÃO DAS MELODIAS

As sessões 5.4.1 e 5.4.2 apresentam exemplos de melodias compostas com a utilização dos algoritmos de treinamento BPTT e LSTM, respectivamente. As redes foram treinadas com os dois tipos de representação discutidos anteriormente, representação por intervalos e por ciclos de terças. A taxa de aprendizado utilizada no treinamento é $\alpha = 0,3$. Os compassos musicais do conjunto de treinamento e os dados dos relevos geográficos formam o vetor de entrada na fase de treinamento.

Duas melodias foram selecionadas para comparar os treinamentos da rede BPTT e LSTM. No final da sessão 5.4.2 é apresentada uma tabela comparando os treinamentos das doze melodias citadas acima pelas duas redes em termos de tempo de processamento e épocas necessárias para atingir um erro médio pré-estabelecido.

5.4.1 – Aspectos de composição com BPTT

O modelo da rede BPTT está apresentado na sessão 4.4.1. Para duração, foram utilizados dezesseis entradas, dezesseis neurônios escondidos e dezesseis neurônios de saída. Para os acordes, foram utilizadas sete entradas, sete neurônios escondidos e sete neurônios de saída.

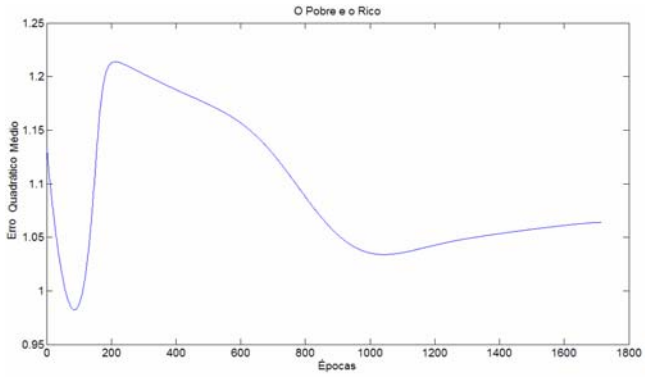
A representação por ciclos de terças utiliza quinze entradas, quinze neurônios escondidos e nove neurônios de entrada, de tal forma que a cada passo de treinamento é apresentado para a rede uma nota da melodia de treinamento e sua informação de oitava e uma nota da inspiração e a saída da rede representa a próxima nota da melodia de treinamento e a respectiva informação de oitava. Na fase de aplicação, que consiste na composição de uma nova melodia, a rede utiliza suas próprias saídas, a partir da primeira nota do treinamento que é apresentada para a

rede juntamente com uma nota de uma nova inspiração, semelhante à utilizada no treinamento.

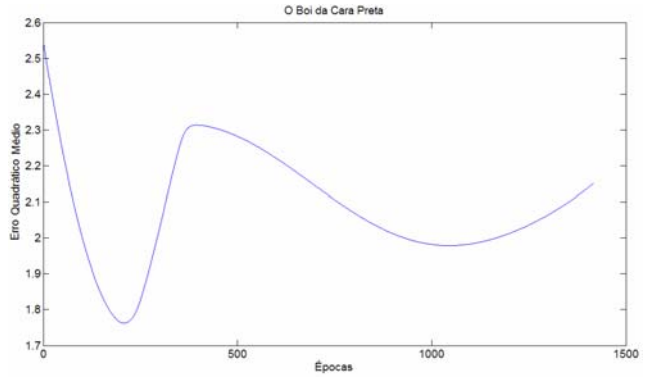
Na representação por intervalo, cada passo de treinamento consiste em apresentar para a rede quatro notas da melodia de treinamento e duas notas da inspiração sendo a tarefa da rede produzir as próximas quatro notas da melodia de treinamento. Portanto, para esse treinamento foram utilizados seis neurônios de entrada, seis neurônios escondidos e quatro neurônios de saída. Na fase de aplicação, a rede recebe as quatro primeiras notas da melodia de treinamento e duas notas representando os dados de relevos geográficos parecidos aos utilizados na composição e cada saída é realimentada para a formação da melodia.

Outro estudo foi realizado quanto ao treinamento da rede BPTT. Esse estudo se caracteriza pela configuração dos neurônios na camada escondida. Verificou-se que para essa aplicação de composição musical, acrescentar camadas escondidas não necessariamente melhora o desempenho da rede.

A Figura 5.22 mostra o erro quadrático médio para o treinamento das melodias “O Pobre e o Rico” e “O Boi da Cara Preta” utilizando representação por intervalo. A Figura 5.23 mostra o erro quadrático médio das melodias “Escravos de Jó” e “O Cravo e a Rosa” utilizando a representação por ciclos de terças.

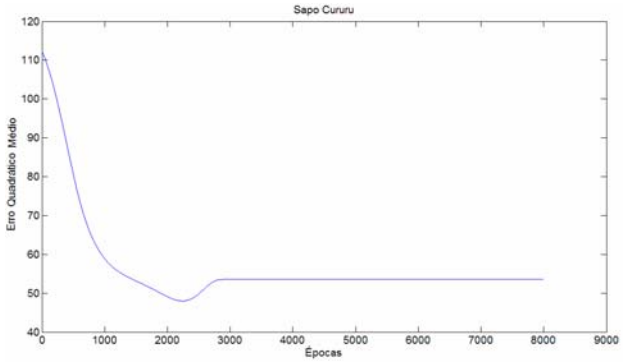


(a)

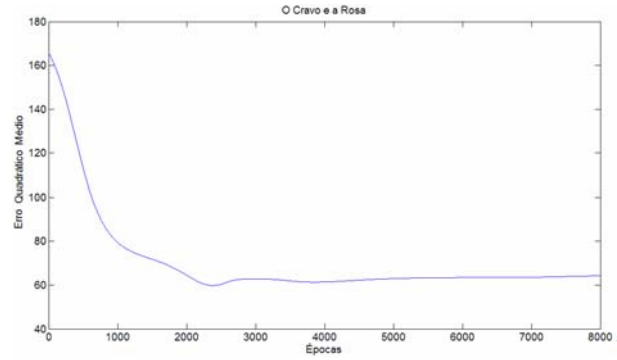


(b)

Figura 5.22: Erro quadrático médio do treinamento da rede BPTT com representação por intervalo (a) O Pobre e o Rico (b) O Boi da Cara Preta



(a)



(b)

Figura 5.23: Erro quadrático médio do treinamento rede BPTT com representação de ciclos de terças (a) Sapo Cururu (b) O Cravo e a Rosa

As Figuras 5.24 e 5.25 apresentam as melodias finais geradas pela rede BPTT, utilizando representação por intervalo e por ciclo de terças, respectivamente.

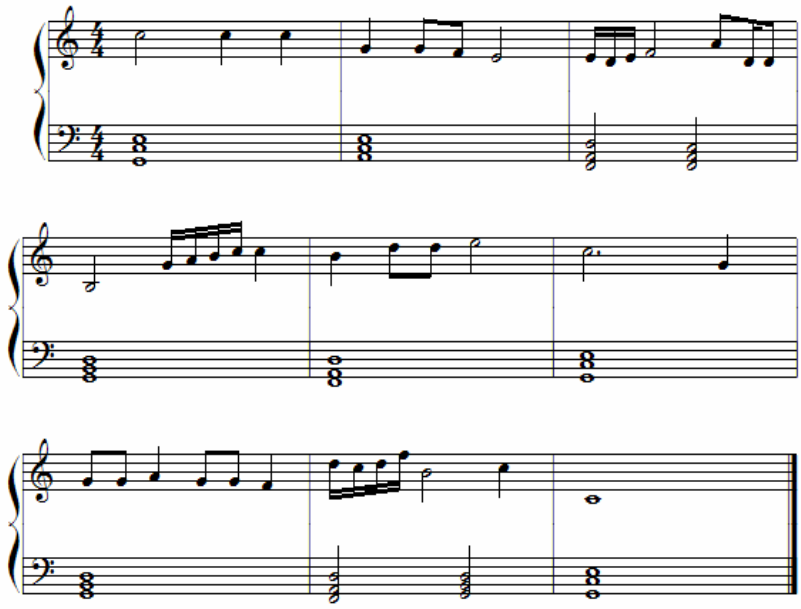


Figura 5.24: Melodia final composta pela rede BPTT com representação por intervalo

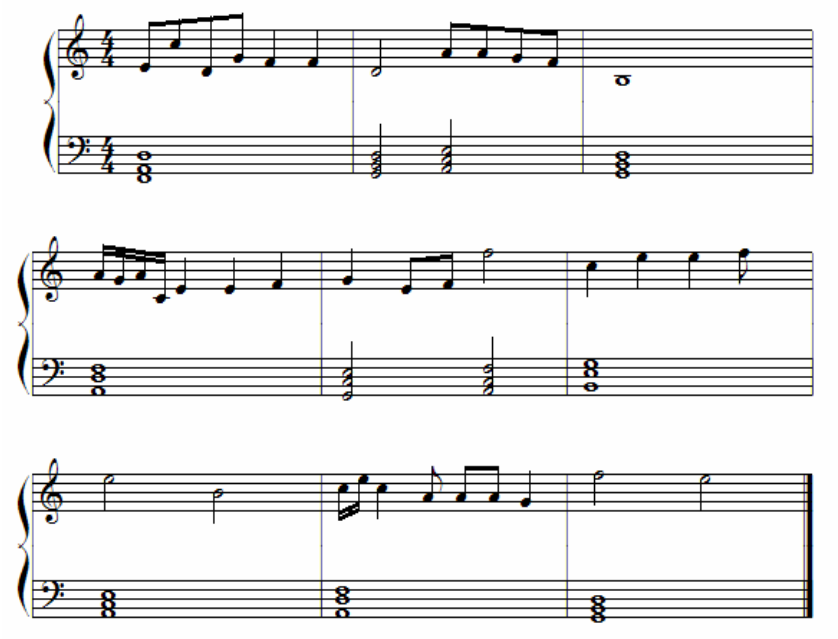


Figura 5.25: Melodia final composta pela rede BPTT com representação por ciclo de terças

5.4.2 – Aspectos de composição com LSTM

O modelo de rede está apresentado na sessão 4.4.2. Como no caso da rede BPTT, a rede LSTM possui uma camada escondida. Para a duração, também foram utilizados dezesseis entradas, dezesseis neurônios escondidos (dezesseis blocos de memória com uma célula de memória cada) e dezesseis saídas. Para acordes, foram utilizadas sete entradas, sete blocos de memória com uma célula cada e sete blocos de saída.

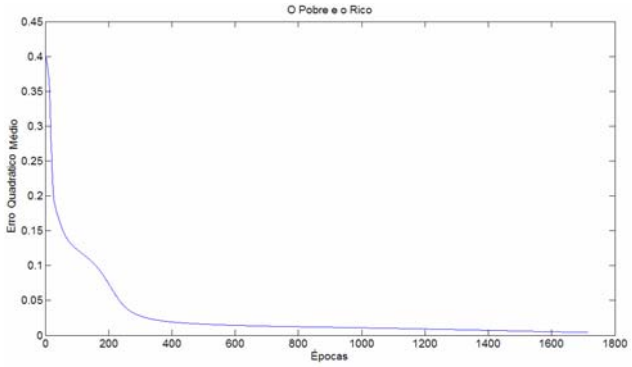
A representação por ciclo de terças utilizou quinze entradas, quinze blocos de memórias com uma célula cada e nove neurônios de saída, de tal forma que cada passo de treinamento e aplicação seja feito como na rede BPTT. Não diferentemente, para a representação por intervalos foram necessários seis entradas, seis blocos de memória com uma célula cada e quatro neurônios de saída.

Para a rede LSTM, além do método desenvolvido de iniciação dos pesos e estimação dos neurônios escondidos que está descrito na sessão 4.4.2.1, alguns estudos foram realizados com o objetivo de estimar outras configurações. Verificou-se que a rede apresenta melhor desempenho (em termos de tempo de treinamento e convergência) nos seguintes casos:

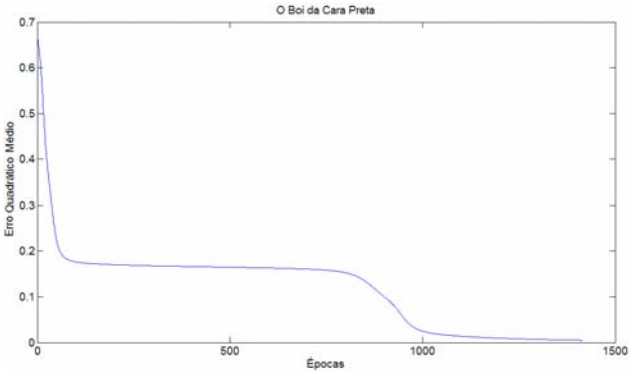
- Os neurônios de entrada possuem conexões diretas com os neurônios de saída. Conexões diretas, nesse caso, estão relacionadas à não existência de pesos nessas conexões.

- As saídas das células de memória possuem auto-realimentação e realimentação para as células de memória do mesmo bloco e de outros blocos.
- As saídas das células de memórias são zeradas a cada época de treinamento.
- O estado inicial da célula e as derivadas parciais são zeradas a cada época de treinamento.
- Inclusão de *bias* no *gate* de entrada, no *gate* de saída e no neurônio de saída.

As mesmas melodias foram usadas para ilustrar os erros obtidos pela rede LSTM. A Figura 5.26 mostra o erro quadrático médio para o treinamento das melodias “O Pobre e o Rico” e “O Boi da Cara Preta” utilizando representação por intervalo. A Figura 5.27 mostra o erro quadrático médio com a representação por ciclos de terças para as melodias “Sapo Cururu” e “O Cravo e a Rosa”.

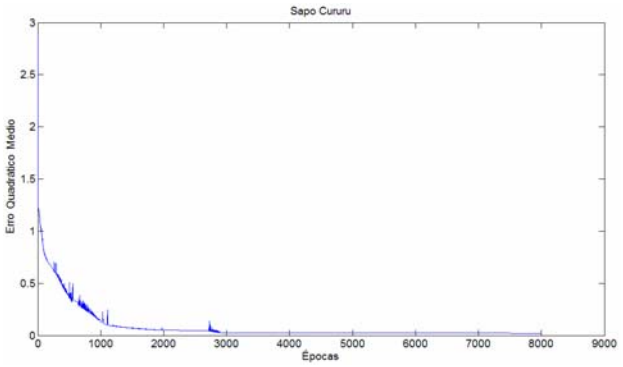


(a)

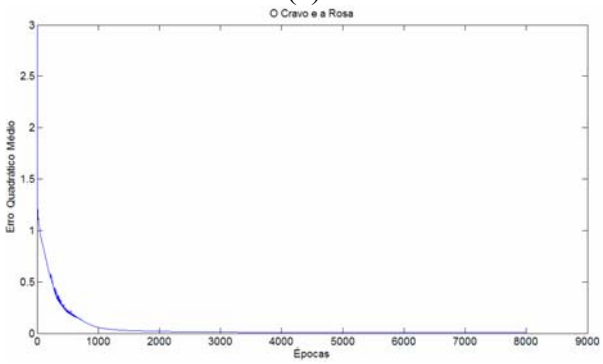


(b)

Figura 5.26: Erro quadrático médio do treinamento rede LSTM com representação por intervalo (a) O Pobre e o Rico (b) O Boi da Cara Preta



(a)



(b)

Figura 5.27: Erro quadrático médio do treinamento rede LSTM com representação de ciclos de terças (a) Sapo Cururu (b) O Cravo e a Rosa

As Figuras 5.28 e 5.29 apresentam as melodias finais geradas pela rede LSTM, utilizando representação por intervalo e por ciclo de terças, respectivamente.

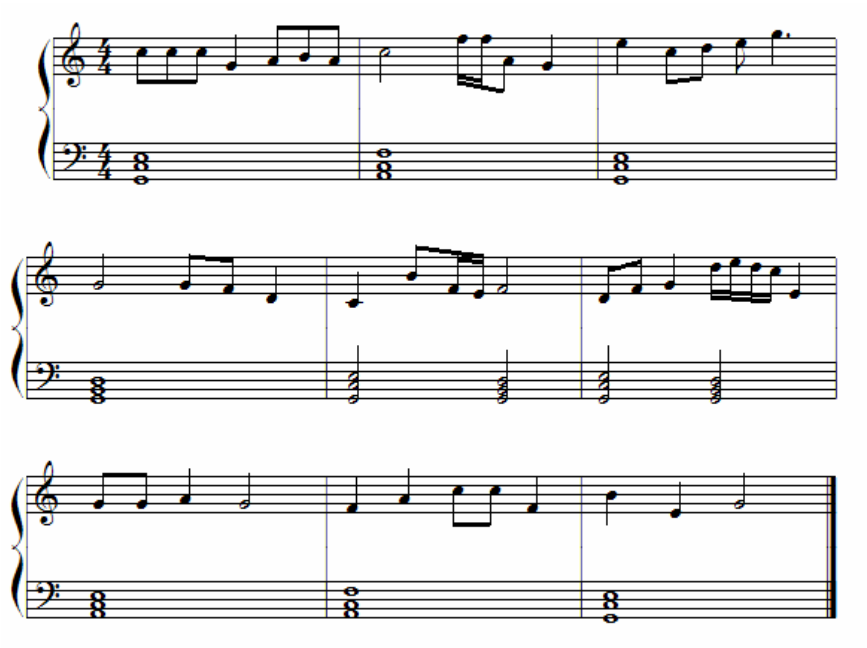


Figura 5.28: Melodia final composta pela rede LSTM com representação por intervalo

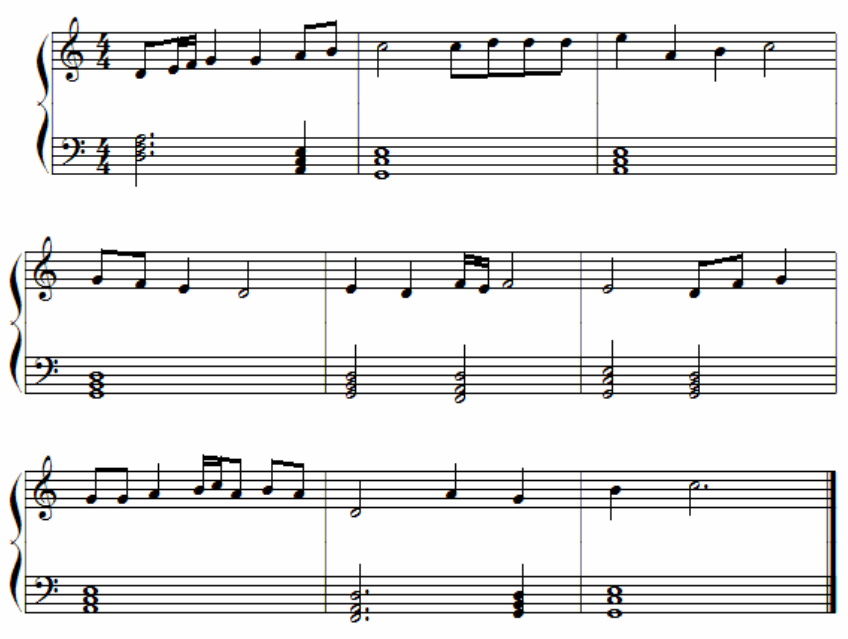


Figura 5.29: Melodia final composta pela rede LSTM com representação por ciclo de terças

5.4.3 – Comparação dos treinamentos das redes BPTT e LSTM

A Tabela 5.2 apresenta o número de épocas necessárias e a duração (em segundos) do treinamento para que as redes LSTM atingissem um erro médio de 0,03 sobre todos os padrões de entrada, para 10 melodias do conjunto de treinamento, utilizando a representação por intervalos. Os resultados da rede BPTT indicam a duração do treinamento e o erro atingido para a mesma quantidade de épocas que a rede LSTM precisou para atingir o erro médio 0,03. Nota-se que a rede LSTM apresenta melhor desempenho (em termos de tempo de treinamento e convergência) no treinamento de todas as melodias.

Tabela 5.2: Épocas e duração de treinamento das redes LSTM e BPTT

	LSTM – Erro Médio 0,03		BPTT		
	Épocas	Tempo	Erro médio	Épocas	Tempo
Boi da cara preta	1417	67 seg	2,15	1417	197 seg
Escravos de Jó	14417	1164 seg	2,00	14417	3651 seg
Onde está a margarida	3416	187 seg	1,83	3416	548 seg
O pobre e o rico	1716	88 seg	1,06	1716	274 seg
O gato	4483	340 seg	2,41	4483	1067 seg
Mulher Rendeira	85392	10928 seg	3,13	85392	52755 seg
Sapo Cururu	26827	1241 seg	0,77	26827	5294 seg
Samba lêlê	30438	4278 seg	2,6	30438	154479 seg
O cravo e a rosa	2640	178 seg	1,30	2640	483 seg
Peixe vivo	91 097	18774 seg	7,21	91097	67239 seg

A Tabela 5.3 apresenta o erro atingido e a duração do treinamento para 8000 épocas de treinamento das duas redes, LSTM e BPTT utilizando a representação de ciclos de terças, para 11 melodias do conjunto de treinamento. Novamente, a rede LSTM apresentou melhores resultados.

Tabela 5.3: Erro médio e duração de treinamento das redes LSTM e BPTT para 8000 épocas de treinamento

	LSTM		BPTT	
	Erro médio	Tempo	Erro médio	Tempo
Boi da cara preta	0,04	6214 seg	62,39	14592 seg
Escravos de Jó	2,90	10444 seg	65,75	24326 seg
Onde está a margarida	2,11	7852 seg	31,96	18456 seg
O pobre e o rico	4,52	7854 seg	63,33	18532 seg
O gato	3,08	11084 seg	80,53	25538 seg
Oh! Minas Gerais	5,48	17934 seg	170,77	41344 seg
Mulher Rendeira	3,18	17334 seg	82,96	41296 seg
Sapo Cururu	0,52	6600 seg	53,56	15866 seg
Samba lêlê	3,36	20520 seg	212,73	47244 seg
O cravo e a rosa	0,17	8910 seg	64,29	21062 seg
Peixe vivo	1,63	22068 seg	147,42	51728 seg

5.5 –AVALIAÇÃO E OTIMIZAÇÃO DAS MELODIAS

Os atributos NRM, MAA e NFT foram extraídos das quatro novas melodias criadas pelas redes BPTT e LSTM usando representação por intervalo e por ciclo das terças. Essa informação está apresentada na Tabela 5.4. A Tabela 5.5 apresenta a classificação obtida por essas duas abordagens para as quatro melodias da tabela 5.4.

Tabela 5.4: Exemplos de atributos extraídos para as novas melodias compostas pelas redes BPTT e LSTM.

	NRM	MAA	NFT
BPTT intervalo	0,27	0,03	0
LSTM intervalo	0,15	0	0
BPTT ciclo de terças	0,13	0	0
LSTM ciclo de terças	0,14	0	0

Tabela 5.5: Avaliação obtida para as novas melodias

	Distância Euclidiana		MLP	Classificação
	Vetor média das melodias apropriadas	Vetor média das melodias inapropriadas		
BPTT intervalo	0,0794	0,188	1,000	Inapropriada
LSTM intervalol	0,0462	0,2221	0,0004	Apropriada
BPTT ciclo de terças	0,0661	0,2311	0,0002	Apropriada
LSTM ciclo de terças	0,0562	0,2264	0,0003	Apropriada

Como pode ser observado, a primeira melodia foi classificada como apropriada pela distância euclidiana e como inapropriada pela rede MLP. Em situações com diferentes conclusões, a decisão final será a classificação da rede MLP. Portanto, a melodia composta pela rede BPTT (Figura 5.24) com representação por intervalos de notas foi classificada como inapropriada e deverá ser otimizada. Um exemplo dessa otimização pode ser observado na Figura 5.30, em que as notas circuladas em vermelho indicam as notas que foram corrigidas.

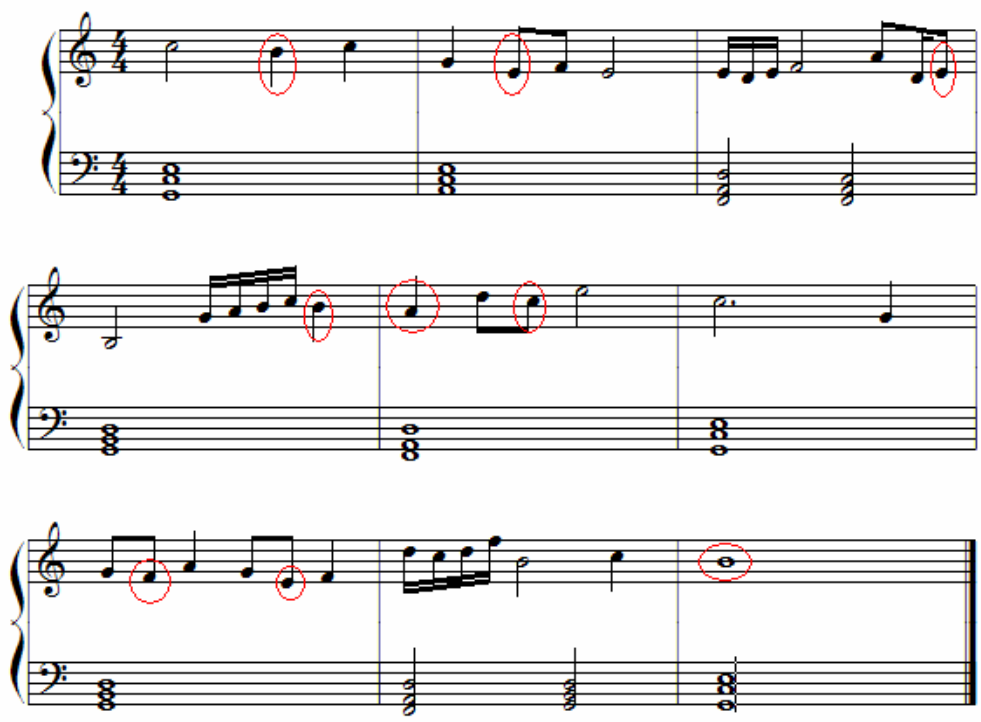


Figura 5.30: Melodia final composta pela rede BPTT com representação por intervalos depois da correção

Como comparação dos resultados da avaliação proposta no trabalho, as quatro novas melodias da Tabela 5.4 foram ouvidas por 14 pessoas diferentes. Essas pessoas classificaram as melodias como “ruim”, “regular” ou “boa”, como uma analogia à apropriada e inapropriada, respectivamente. A Tabela 5.6 apresenta os resultados obtidos da avaliação subjetiva. Das 14 pessoas que avaliaram as melodias 8 pessoas relataram que possuem conhecimento básico sobre música, 4 pessoas relataram que possuem conhecimento intermediário, e 2 pessoas relataram conhecimento avançado. Comparando os resultados na Tabela 5.6 nota-se que uma quantidade significativa (16) de avaliações boas foi atribuída para as melodias da rede LSTM em relação às melodias da rede BPTT (8). Nota-se também que uma quantidade maior (6) de avaliações ruins foi dada as melodias da rede BPTT e quantidade menor (2) em relação a rede LSTM. Comparando os resultados das tabelas 5.5 e 5.6 verifica-se uma

compatibilidade entre os dois procedimentos de avaliação, devido a melhor avaliação dos resultados de composição da rede LSTM.

Tabela 5.6: Resultado das avaliações

Melodias	Avaliação		
	Ruim	Regular	Boa
BPTT intervalo	2	7	5
BPTT ciclo de terças	4	7	3
Total BPTT	6	14	8
LSTM intervalo	1	4	9
LSTM ciclo de terças	1	6	7
Total LSTM	2	10	16
Total geral	8	24	24

5.6 – CONSIDERAÇÕES FINAIS

Esse capítulo apresentou os resultados obtidos nessa dissertação de mestrado para o sistema proposto de composição musical. Nota-se que, independente do algoritmo de aprendizagem, o sistema proposto é útil para a composição primária de melodias, de tal forma que o usuário possa desempenhar análises posteriores. Apesar do conjunto de treinamento ser o mesmo, a arquitetura da rede e a representação do conjunto de treinamento pode influenciar no resultado final. Todas as melodias compostas são diferentes.

O próximo capítulo descreve as conclusões e comentários sobre esses resultados. Também são apresentadas propostas para trabalhos futuros.

Nesse trabalho de dissertação de mestrado foi proposto e desenvolvido um sistema de composição musical assistido por computador que desempenha desde o treinamento até a avaliação e correção das melodias. A avaliação foi baseada em três atributos e foram propostas duas abordagens para determinar se a nova melodia é apropriada ou inapropriada. Dependendo das normas, dos classificadores e dos atributos utilizados, as duas abordagens para avaliação das melodias podem obter resultados diferentes. As melodias consideradas não apropriadas são corrigidas com base na tabela de probabilidade condicional que é montada sobre todas as melodias do conjunto de treinamento.

Apesar da utilização da inspiração baseada nos contornos dos relevos geográficos, nota-se que o sistema pode generalizar a origem da inspiração, o que pode produzir resultados diferentes de composição.

No processo de treinamento estudado neste trabalho, foi observado que a rede LSTM pode aprender de maneira mais apropriada que a rede BPTT, principalmente na representação por ciclo de terças. Dessa forma, as novas melodias apresentaram, mais adequadamente, a informação estatística do conjunto de treinamento. Além disso, as duas melodias geradas pela rede LSTM apresentam algumas seqüências de notas similares, dado que os dois treinamentos foram bem desempenhados. As melodias geradas pela rede BPTT são mais distantes do conjunto de treinamento e da tabela de probabilidade condicional das notas, uma vez que a rede não conseguiu aprender adequadamente. Portanto, a disparidade entre as seqüências de notas do conjunto de treinamento e seqüências de notas das novas melodias é significantemente maior. Pode ser notado que na aplicação da tabela de probabilidade condicional, as notas escolhidas nas melodias geradas pela rede BPTT correspondem a valores com menores probabilidades de ocorrência em comparação com as notas escolhidas nas melodias geradas pela rede LSTM. Isso significa que a contribuição das probabilidades condicionais para a geração da melodia final é menor para a rede BPTT do que para a rede LSTM.

O melhor desempenho no treinamento da rede LSTM em relação à rede BPTT também é verificado pela velocidade através dos tempos de treinamento. Ou seja, a rede LSTM também desempenha a fase de treinamento de forma mais rápida, o que lhe confere características desejadas para a aplicação desenvolvida.

O trabalho de mestrado também descreve o comportamento dos neurônios escondidos de uma rede neural LSTM na tarefa de aproximar funções não lineares. A aproximação é feita pela união de aproximações lineares por partes. Foi

proposto aprimorar o treinamento da rede LSTM estimando o número ideal de neurônios escondidos em conjunto com um método para iniciação dos pesos da rede. Com o método proposto de iniciação, alguns pesos são definidos perto dos locais desejados finais depois do treinamento, otimizando assim o tempo de treinamento da rede. Também foi observado que quanto maior o número de neurônios escondidos utilizados para aproximar uma função desejada, melhores são os resultados obtidos com o método de iniciação proposto.

Os resultados também mostraram que, com a iniciação aleatória, aumentar o número de blocos de memória não necessariamente garante um melhor desempenho da rede, uma vez que blocos de memória adicionais podem não ser bem iniciados (valores iniciais são definidos aleatoriamente) e suas configurações iniciais estão distantes das configurações finais ideais. Além disso, dois ou mais blocos de memória podem ser iniciados bem próximos um do outro e se manter próximos depois do processo de treinamento, não permitindo contribuições significativamente diferentes, uma vez que suas saídas podem estar altamente correlacionadas (dois ou mais blocos podem ter saídas quase que equivalentes). A iniciação aleatória geralmente apresenta grandes diferenças entre o melhor e o pior desempenho, refletindo um comportamento instável na fase de treinamento. Diferentemente, o método proposto de iniciação combinado com a detecção de pontos de mínimos e máximos locais oferecem uma boa estimativa da configuração ideal da rede. Ainda, o método minimiza o problema de instabilidade que pode ocorrer nos treinamentos das redes neurais.

Como uma união das duas partes da dissertação (sistema para composição musical envolvendo redes neurais e método de iniciação de pesos da rede LSTM) verifica-se que uma das maiores contribuições desse trabalho é a possibilidade de redução significativa do tempo de treinamento em aplicações de composição musical envolvendo redes neurais, que pode ser conseguida primeiramente pelo uso da rede LSTM ao invés de redes *back-propagation* tradicionais e, segundo, pela otimização dos pesos iniciais das redes LSTM. A combinação dessas duas abordagens pode trazer um ganho significativo no tempo de aprendizado.

6.1 – TRABALHOS FUTUROS

Trabalhos futuros incluem a inserção de outros atributos musicais, como, por exemplo, elementos de dinâmicas e frases musicais. Outra possibilidade de melhoria é a inclusão de um conjunto de treinamento com músicas mais complexas,

com a utilização de duas claves musicais como uma alternativa a utilização de acordes.

Trabalhos futuros também podem incluir o uso de dimensões fractais da música, calculados por diferentes métodos, como o FBM (*Fractal Brownian Motion*) [BIGERELLE e LOST, 2000] para caracterizar a melodia gerada com o objetivo de avaliar melhor a fase de composição. É também interessante o uso do coeficiente de Kappa [COHEN, 1960] para medir o desempenho da classificação, o que permite a definição de um *framework* completo para análise quantitativa dos dados.

Quanto ao método proposto para iniciação dos pesos e configuração dos neurônios escondidos da rede LSTM, trabalhos futuros incluem a tentativa de otimizar a iniciação de outros parâmetros, como os parâmetros recorrentes e do *bias.* Também sugere-se verificar o desempenho do método para funções mais complexas e para outras aplicações, como por exemplo, tarefas de classificação e reconhecimento de padrões com o objetivo de comparar o desempenho da rede com e sem o método proposto em problemas reais.

ADILOGLU, K.; ALPASLAN, F. N. **A machine learning approach to two-voice counterpoint composition.** *Knowledge-Based Systems*, V. 20, pp. 300-309, 2007.

ARAUJO, A. M. **Cem melodias folclóricas**. Martins Fontes, 2007.

AZEVEDO, F. M.; BRASIL, L. M.; OLIVEIRA, R. C. L. **Redes Neurais com Aplicações em Controle de Sistemas Especialistas.** SC: Visual Books, 2000.

BASILIO JOAQUIM, M.;SARTORI J.C. **Análise de Fourier.** SP: Departamento de Engenharia Elétrica – São Carlos, 2003.

BENSON, D. J., **Music: A Mathematical Offering.** Cambridge, USA, 2007.

BHARUCHA, J.J.; TODD, P.M. Modeling the Perception of Tonal Structure with Neural Nets. Computer Music Journal, Vol.13, No.4, 1989

BHARUCHA, J. J. **Pitch, Harmony, and Neural Nets: A Psychological Perspective**. *Music and Connectionism*, MIT Press, 1991.

BIGERELLE, M., LOST, A. **Fractal dimension and classification of music**. *In Chaos, Solutions and Fractals,* Vol. 11, pp.2179 2192, 2000.

BRAGA, A. P.; LUDEMIR, T. B.; CARVALHO, A. P. **Redes Neurais Artificiais – Teoria e Aplicações.** R.J.: JC, 2000.

CARPINTEIRO, O. A. S. **A neural model to segment musical pieces.** *In Proceedings of the Second Brazilian Symposium on Computer Music, Fifteenth Congress of the Brazilian Computer Society*, p. 114 – 120. Brazilian Computer Society, 1995.

CHEN, C.-C. J.; MIIKKULAINEN, R. **Creating Melodies with Evolving Recurrent Neural Network**. *In Proceedings of the International Joint Conference on Neural Networks, IJCNN'01,* p. 2241 – 2246, Washington - DC, 2001.

CHOMSKY, N. **Syntactic Structures**. The Hague: Mouton and Co., 1957

CORREA, D. C.; LEVADA A. L. M.; SAITO, J. H.; MARI, J. F. **Neural Network based Systems for Computer-Aided Musical Composition: Supervised x Unsupervised Learning**. *In: Proceedings of the 2008 ACM Symposium on Applied Computing,* v.3, p. 1738-1742, Fortaleza – CE, 2008.

CORREA, D. C.; LEVADA A. L. M.; SAITO, J. H. **Stabilizing and Improving the Learning Speed of 2-Layered LSTM Network.** *In: Proceedings on the 2008 IEEE 11th International*

Conference on Computational Science and Engineering, Los Alamitos, CA : IEEE Computer Society, 2008. p. 293-300.

COSTA, L. F.; CÉSAR, R. M. **Shape Analysis and Classification: Theory and Practice**. CRC Press: 2001.

COHEN, J. **A coefficient of agreeement for nominal scales.** *In Edu. Psycol. Measurement*, Vol 20, N. 1, pp. 37-46, 1960

DOLSON, M. **Machine Tongues XII: Neural Networks**. *Computer Music Journal*, Vol.13, No. 3, 1989.

ECK, D.; SCHMIDHUBER, J. **A First Look at Music Composition using LSTM Recurrent Neural Networks.** *Technical Report: IDSIA-07-02,* 2002.

FAUSETT, L. **Fundamentals of Neural Networks (Architectures, Algorithms, and Applications)**. New Jersey: Prentice Hall International, Inc, 1994.

FILHO, O. M.; NETO, H.V. **Processamento Digital de Imagens**. Rio de Janeiro, Brasport, 1999.

FRANKLIN, J. A. Franklin. **Recurrent Neural Networks for Music Computation**. *Informs Journal on Computing*, Vol.18, No.3, pp.321-338, 2006.

GAVES A., SCHMIDHUBER J., **Framewise phoneme classification with bidirectional LSTM and other neural network architectures**. *Neural Networks*, Vol. 18, Issues 5-6, p. 602-610, 2005.

GERS, F. A.: **Long Short-Term Memory in Recurrent Neural Networks**. PhD thesis (2001)

GERS, F. A.; SCHMIDHUBER, J.; CUMMINS, F. **Learning to Forget: Continual Prediction with LSTM**. *Neural Computation,* 12(10): 2451 – 2471, 2000.

GOGA, M.; GOGA, N. **Feelings Based Computer Music.** *In Electrical and Computer Engineering*, p.673 – 676. Canadian Conference: 2004

GRIFFITH, N.; TODD, P. M. **Musical Networks: Parallel Distributed Perception and Performance**. Cambridge, MA: MIT Press, 2001.

HAYKIN, S. **Neural Networks – A Comprehensive Foundation.** Prentice Hall, 1999.

HOCHREITER S., BENGIO Y., FRASCONI P., SCHMIDHUBER J., **Gradient flow in recurrent nets: The difficulty of learning long-term dependencies**. **A Field Guide to Dynamical Recurrent Networks**, *IEEE Press*, New Your, 2001.

HOCHREITER S., SCHMIDHUBER J., **Long Short-Term Memory**. *Neural Computation*, 9(8):1735-1780, 1997.

IRIE, B.; MIYAKE, S. **Capabilities of three-layerd perceptrons.** *In proceedings of the IEEE International Conference on Neural Networks*, pp. I-641, 1998.
KAK, A.C.; SLANEY M. **Principles of Computerized Tomographic Imaging**. *Society of Industrial and Applied Mathematics*, 2001.

KOHONEN, T. **A self-learning musical grammar or "Associative memory of the second Kind"**. *In Proceedings on the International Joint Conference on Neural Network,* p. 1 – 5, 1989.
KOHONEN, T.; LAINE, P.; TIITS, K.; TORKKOLA, K. **A Nonheuristic Automatic Composing Method**. *Music and Connectionism*, MIT Press: 1991.

LADEN, B.; KEEFE, D. H. **The Representation of Pich in a Neural Net Model for Chord Classification**. *Computer Music Journal*, Vol. 13, No.4, 1989.

LENT, R. **Cem bilhões de neurônios – Conceitos Fundamentais de Neurociência**. Atheneu, São Paulo: 2002.

LEWIS, J. P. **Creation by Refinement and the Problem of Algorithmic Music Composition**. *Music and Connectionism*, MIT Press: 1991.

LONGUET – HIGGINS, H. C. **The Perception of Music.** *In Proceedings of Royal Society of London,* B. 205, p. 307 – 322, 1979.

LOY, D. G. **Connectionism and Musiconomy**. *Music and Connectionism*, MIT Press: 1991.

MCLLOCH, W. S.; PITTS, W. **A logical calulus of ideas immanent in nervous activity.** *Bulletin of Mathematical Biophysics*, 5: 115-133, 1943.

MIRANDA, E. R. **Composing Music with Computers.** Burlington, MA: Focal Press, 2001.

MOZER, M. C. **Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multiscale processing**. *Connection Science*, 6(2-3), p. 247 – 280), 1994.

NGUYEN D., WIDROW B. **Improving the learning speed of 2-layer neural networks by choosing initial values of adaptive weights.** *In Proc. IJCNN*, vol. 3, pp. 21-26, 1990.

PAULA FILHO, W.P. **Multimídia Conceitos e Aplicações**. R.J : LTC, 2000.

PAPADOPOULOS, G.; WIGGINS, G. **AI Methods for Algorithmic Composition: A Survey, a Critical View and Future Prospects**. *Proceedings of the AISB'99 Symposium on Musical Creativity*, p. 110 – 117. Brighton, UK: SSAISB, 1999.

PÉREZ –ORTIZ J. A., GERS F. A., ECK D., SCHMIDHUBER J., **Kalman Filters improve LSTM network performance in problems unsolvable by traditional recurrent nets**. *Neural Networks*, vol. 16, issue 2, p. 241-250, 2003.

RATT,W.K. **Digital Image Processing.** John Wiley & Sons, Estados Unidos, 1991.

ROEDERER, J.G. **Introdução à física e psicofísica da Música.** São Paulo, Edusp: 1998.

ROWE, R. **Machine Musicianship**. Massachusetts: MIT Press, 2001.

SANO, H.; JENKINS, B. K. **A neural Network Model for Pitch Perception.** *Computer Music Journal*, Vol. 13, No. 3, 1989.

SCARBOROUGH, D.L.; MILLER, B. O.; JONES, J. A. **Connectionist Models for Tonal Analysis**. *Computer Music Journal,* Vol.13, No.3, 1989.

SCHMIDHUBER J., WIERSTRA D., GAGLIOLO M., Gomez F., **Training Recurrent Networks by Evolino**. *Neural Computation*, MIT Press Journal, 19: 757-779, 2007.

SCHOENBERG, A. **Fundamentals of Musical Composition**. Erwin Stein, 1967.

SEJNOWSKI, T. J.; ROSENBERG, C.R. **Parallel Networks that learn to pronounce English text**. *Complex Systems*, Vol.1, p. 145-168, 1987.

SHEPARD, R. N. **Geometrical approximations to the structure of musical pith.** *Psychological Review,* 89, p. 305-333, 1982

SMITH, G. D. **Numerical Solution of Partial Differential Equations: Finite Difference Methods**. Third Edition, Clarendon Press, Oxford: 1985.

SMITH, S.W. **The Scientist and Engineer's Guide to Digital Signal Processing.** California Technical Publishing, 1997.

TANG,H., TAN K.C.; YI Z. **Neural Networks: Computational Models and Applications**, *Springer*, New York, 2007.

TEMPERLEY, D. **The Cognition of Basic Musical Structures**. Cambridge, Massachusetts: MIT Press, 2001.

TODD, P. M. **A Connectionist Approach to Algorithmic Composition**. *Computer Music Journal:* Vol.13, No. 4, 1989.

TODD, P. M.; LOY, D. G. **Music and Connectionism**. MIT Press, 1991.

TODD, P. M.; WERNER, G. M. **Frankensteinian Methods for Evolutionary Music Composition**. *Musical Networks: Parallel Distributed Perception and Performance*. Cambridge, MA: MIT Press, 1998.

VERBEURGT, K.; FAYER, M.; DINOLFO, M. **A Hybrid Neural-Markov Approach for Learning to Compose by Example.** *LNAI,* 3060, pp.480-484, 2004.

WILLIAMS R. J., ZIPSER D., Gradient-Based Learning Algorithms for Recurrent Networks and Their Computational Complexity. In Back-propagation:Theory, Architectures and Applications. Hillsdale, NJ: Erlbaum, 1992.