UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

REPRESENTAÇÃO MULTIMODAL PARA CLASSIFICAÇÃO DE INFORMAÇÃO

FERNANDO TADAO ITO

ORIENTADORA: PROFA. DRA. HELENA DE MEDEIROS CASELI
CO-ORIENTADOR: PROF. DR. JANDER MOREIRA

São Carlos – SP

Abril/2018

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

REPRESENTAÇÃO MULTIMODAL PARA CLASSIFICAÇÃO DE INFORMAÇÃO

FERNANDO TADAO ITO

Dissertação 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, área de concentração: Inteligência Artificial

Orientadora: Profa. Dra. Helena de Medeiros Caseli

São Carlos – SP

Abril/2018

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Fernando Tadao Itó, realizada em 08/06/2018:

Helena de M. Caseli
Profa. Dra. Helena de Medeiros Caseli
UFSCar

João Paulo Papa
Prof. Dr. João Paulo Papa
UFSCar

Prof. Dr. Viviane Pereira Moreira
UFRGS

Certifico que a defesa realizou-se com a participação à distância do(s) membro(s) Viviane Pereira Moreira e, depois das arguições e deliberações realizadas, o(s) participante(s) à distância está(ao) de acordo com o conteúdo do parecer da banca examinadora redigido neste relatório de defesa.

Helena de M. Caseli
Profa. Dra. Helena de Medeiros Caseli

A meus pais. Para um dia eu ser um professor tão completo quanto eles foram para mim.

AGRADECIMENTOS

Obrigado a todos que fizeram parte desta jornada comigo, ao CNPq, que forneceu a bolsa nos primeiros meses do mestrado, à FAPESP (Projeto MMeaning: 2016/13002-0) e à Monitora Soluções Tecnológicas, por me liberar tempo para o estudo e pesquisa necessários para completar este trabalho.

Epígrafe

Donald Knuth

Science is knowledge which we understand so well that we can teach it to a computer; and if we don’t fully understand something, it is an art to deal with it.

RESUMO

O significado mais básico de "multimodalidade" é a utilização de múltiplos meios de informação para compor um "artefato", um objeto criado pelo homem que expressa um conceito. Em nosso dia-a-dia, diversos meios de comunicação expressam conceitos a partir de multimídia: notícias com narração, vídeos e textos auxiliares; peças de teatro que contam uma história a partir de atores, gestos e músicas; jogos eletrônicos que utilizam os gestos físicos do jogador como ações, e respondem com sinais visuais ou musicais. Para interpretar tais "artefatos", temos que extrair informações de múltiplos meios de informação e combiná-los matematicamente. A extração de características é feita a partir de modelos matemáticos que recebem um dado bruto (textos, imagens, sinais de áudio) e o transforma em um vetor numérico, onde a distância entre instâncias denota a sua relação: dados próximos codificam significados similares. Para criar um espaço semântico multimodal, utilizamos modelos que "fundem" as informações de múltiplos tipos de dados. Neste trabalho, investigamos a interação entre diferentes modos de representação de informação na formação de representações multimodais, apresentando alguns dos algoritmos mais usados para a representação vetorial de textos e imagens e como fundi-los. Para medir a performance relativa de cada combinação de métodos, utilizamos tarefas de classificação e similaridade em bancos de dados com imagens e textos pareados. Verificamos que, em nossos conjuntos de dados, diferentes métodos de representação unimodal podem levar a resultados vastamente diferentes. Também notamos que a performance de uma representação na tarefa de classificação de dados não significa que tal representação não codifique o conceito de um objeto, tendo diferentes resultados em tarefas de similaridade.

Palavras-chave: representação multimodal, representação distribuída, Word2Vec, SIFT, autoencoder, inteligência artificial, aprendizado não-supervisionado

ABSTRACT

The most basic meaning of ”multimodality”is the use of multiple means of information to compose an ”artifact”, a man-made object that expresses a concept. In our day-to-day life, most media outlets use multimedia to express information: news are composed of videos, narrations and ancillary texts; theater plays tell a story from actors, gestures and songs; electronic games use the player's physical gestures as actions, and respond with visual or musical cues. To interpret such ”artifacts,”we have to extract information from multiple media and combine them mathematically. The extraction of characteristics is done from mathematical models that receive raw data (texts, images, audio signals) and turns it into a numerical vector, where the distance between instances denotes its relation, where close data encode similar meanings. To create a multimodal semantic space, we use models that “ fuse ” information from multiple data types. In this work, we investigate the interaction between different modes of information representation in the formation of multimodal representations, presenting some of the most used algorithms for vector representation of texts and images and how to merge them. To measure the relative performance of each combination of methods, we use classification and similarity tasks in databases with images and paired texts. We found that in our data sets different methods of unimodal representation can lead to vastly different results. We also note that the performance of a representation in the data classification task does not mean that such representation does not encode the concept of an object, having different results in similarity tasks.

Keywords: multimodal representation, distributed representation, Word2Vec, SIFT, autoencoder, artificial intelligence, unsupervised learning

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

Capítulo 1

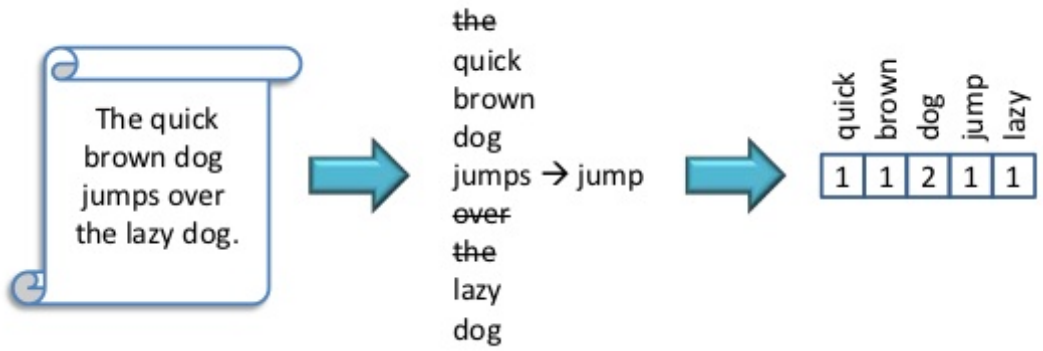
INTRODUÇÃO

A passagem e a disseminação de informações pela humanidade se dá por múltiplos meios de comunicação. Por meio de linguagem, podemos expressar ideias abstratas de maneira ordenada e padronizada; por meio de imagens, podemos visualizar eventos, pessoas e objetos; por meio de áudio, podemos interpretar sinais em determinadas frequências de acordo com nossa audição. No mundo real, múltiplas modalidades se juntam para transmitir informações (vídeos, textos ilustrados e trechos anotados de áudio, por exemplo) que podem ser absorvidas pela imensa capacidade humana de interpretação. Mas, em termos matemáticos, cada tipo de dado tem características intrinsecamente diferentes entre si. Como traduzir tais dados para um domínio onde suas características (atributos) podem ser comparadas e correlacionadas?

Com a disponibilização cada vez maior de informação na web, o processamento e a recuperação de informação textual e visual são atividades imprescindíveis na geração automática de conhecimento. Como a maior parte da informação disponibilizada na web é composta de texto em língua natural e de imagens (STAMOU; KOLLIAS, 2005), interpretar a informação que eles transmitem é necessário para processá-los de maneira automática, sem supervisão humana. Por exemplo, para permitir tal interpretação não-supervisionada do significado de um texto, é necessário representá-lo de maneira matemática.

A área de representação distribuída de dados trata da criação de funções que mapeiam dados brutos (texto, áudio e imagem) em domínios mais simples, revelando padrões não-triviais nestes dados. Muitas informações em um texto não são relevantes para determinadas tarefas: artigos e preposições são conectores sem significado especial sozinhos e podem ser descartados ao se analisar o sentimento de um parágrafo, por exemplo (LIU, 2010). Uma maneira de representar um texto é usando um vetor que mostra o número de ocorrências das palavras em um determinado vocabulário (retirando palavras descartáveis) (ZHANG; JIN; ZHOU, 2010), como ilustrado na Figura 1.1.

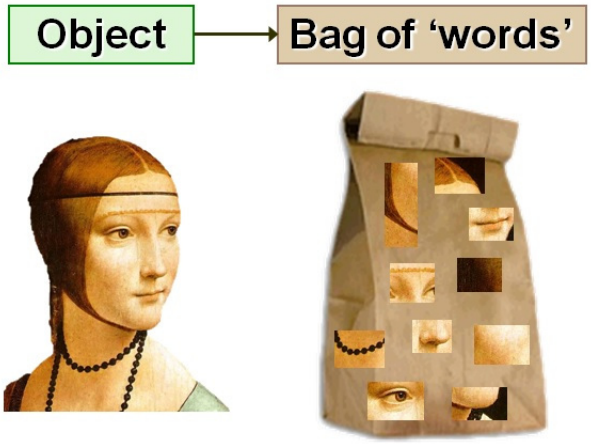
Figura 1.1: Exemplo de representação de texto.



Fonte: Retirado de https://www.slideshare.net/mgrcar/text-and-text-stream-mining-tutorial-15137759

Da mesma maneira, uma imagem pode ser representada pela presença de detalhes comuns ao seu domínio (rodas, janelas e portas são partes comuns de um automóvel, por exemplo) (LI, 2011). A Figura 1.2 ilustra a representação de uma imagem fazendo uma analogia com uma *bag-of-words*[1], na qual as palavras são partes representativas da imagem. Cada uma das modalidades possui seus próprios métodos de extração e refinamento de atributos para a definição deste mapeamento vetorial.

Figura 1.2: Exemplo de representação de imagem.



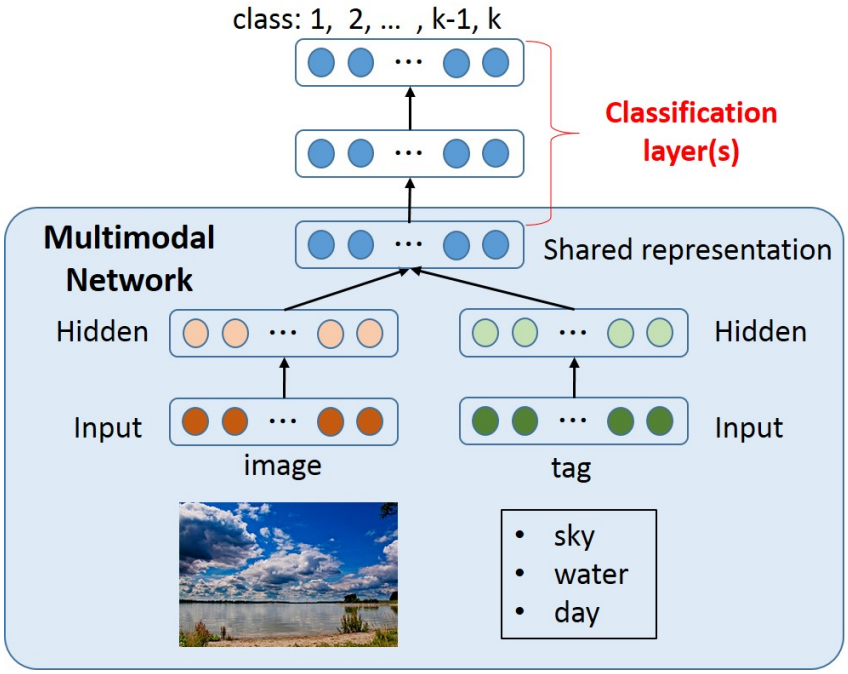
Fonte: Retirado de https://gilscvblog.com/2013/08/23/bag-of-words-models-for-visual-categorization/

A aplicação combinada destes modelos por meio de uma "representação distribuída multimodal" tem inspiração na maneira como o ser humano aprende: utilizando múltiplas fontes de informação de maneira suplementar simultaneamente (GARDNER, 1985). O objetivo de uma

[1] Bag-of-Words é uma representação de dados onde um texto é transformado em um vetor numérico do tamanho do vocabulário do corpus onde está inserido, com seus valores sendo a presença de cada palavra no documento

representação multimodal é complementar as informações obtidas por diferentes meios para capturar o conceito que se deseja representar (ATREY, 2010) (por exemplo, utilizando imagem e texto em conjunto, como na Figura 1.3).

Figura 1.3: Exemplo de representação multimodal.



Fonte: Retirado de http://www.nlab.ci.i.u-tokyo.ac.jp/projects-e.html

Para encontrar essas representações multimodais, precisamos correlacionar palavras em um texto com as propriedades visuais encontradas em imagens para criar densos vetores numéricos que podem ser analisados por meio de métodos estatísticos. Ao aprender o conceito "poltrona", por exemplo, um modelo poderia tratar as diferenças semânticas entre conceitos similares ("cadeira" ou "assento", por exemplo) relacionados a uma mesma imagem e atribuir uma representação única a eles, assim como encontrar diferentes imagens de poltronas de acordo com uma única palavra relacionada a elas. Assim, podemos encontrar características que são invariantes à modalidade, intrínsecas ao conceito. Este tipo de representação poderia ser usada em qualquer tipo de tarefa de aprendizado de máquina que envolva múltiplas fontes de dados previamente pareados. Um exemplo de aplicação prática desta abordagem é a classificação e a comparação de produtos encontrados em lojas eletrônicas, por meio da associação de imagens às descrições de funcionalidades e características do produto.

1.1 Objetivos e Hipóteses

Este projeto de mestrado tem o objetivo de investigar como diferentes métodos de representação de informação textual e visual podem ser combinados para gerar um vetor numérico que represente características encontradas em modalidades distintas.

Duas hipóteses norteiam este trabalho. A primeira delas é a de que a geração de representações multimodais é diretamente influenciada pela escolha dos métodos de representação unimodais usados em seu pré-processamento. Para testar tal hipótese, experimentos com diferentes métodos de representação de informação textual e visual foram realizados. A segunda hipótese está amparada em trabalhos anteriores que já utilizaram representações multimodais para reportar melhora em tarefas de classificação de instâncias com múltiplas fontes de dados (NGIAM, 2011). Aqui, queremos verificar se as informações semânticas encontradas nos dados são retratadas nas representações unimodais existentes, e se há correlação entre os vetores numéricos obtidos por representações em modalidades diferentes. Para testar tal hipótese, propomos uma nova arquitetura para adaptar representações de duas modalidades em uma representação multimodal composta, utilizando uma rede neural para codificá-la.

1.2 Organização do texto

Este texto está organizado como segue. Os três próximos capítulos são reservados para a descrição dos modelos de representação de texto, de imagem e multimodal, respectivamente. Nos Capítulos 2 e 3, descrevemos alguns dos métodos mais populares de representação distribuída de texto e imagem, respectivamente. No Capítulo 4, descrevemos alguns modelos propostos para fundir as informações de duas modalidades diferentes, bem como as arquiteturas aplicadas para isso.

No Capítulo 5 é apresentada a arquitetura proposta para a geração de uma representação multimodal de texto e de imagem. O Capítulo 6 descreve os experimentos realizados com a arquitetura neural proposta para a representação multimodal avaliada em uma aplicação de classificação. Este documento é finalizado com algumas considerações finais (Capítulo 7) sobre o trabalho e suas futuras extensões.

Capítulo 2

REPRESENTAÇÃO DISTRIBUÍDA DE TEXTOS

Para processar e analisar material textual, os algoritmos de aprendizado de máquina precisam traduzir sentenças em linguagem natural para uma representação que tais algoritmos possam entender. Tradicionalmente, essas representações são conhecidas na área de Processamento de Linguagem Natural (PLN) como representação semântica distribuída (BENGIO, 2003), vetor de palavras ou, como referenciado na literatura mais recente sobre o assunto, *word embeddings*. Os modelos de representação distribuída têm como objetivo mapear palavras ou sentenças de um texto para um espaço matemático vetorial de dimensionalidade reduzida.

O embasamento para esse tipo de representação está no fato de que o contexto de uma palavra em uma sentença pode ser considerado por meio de uma representação matemática contínua e multidimensional, e que a posição relativa das palavras dentro deste espaço representa o relacionamento semântico entre elas (HARRIS, 1954): palavras similares compartilham contextos similares. Para gerar automaticamente tal mapeamento, diversos algoritmos e técnicas estatísticas podem ser utilizados: redes neurais e estimadores probabilísticos, por exemplo. Representações matemáticas de palavras e textos são utilizadas em muitas tarefas em PLN, como classificação de texto (JOACHIMS, 1998), similaridade de textos e palavras (KUSNER, 2015), desambiguação de termos (NAVIGLI, 2009), agrupamento de documentos (STEINBACH; KARYPIS; KUMAR, 2000) e análise de sentimentos (PANG; LEE, 2008).

Embora as pesquisas em representação distribuída tenham se popularizado nos últimos anos graças ao uso concorrente com redes neurais para produção automática de representações, existem trabalhos teóricos desde 1950 sobre a correlação de palavras e como usá-las para representação textual. Uma das primeiras técnicas que surgiram para medir a conotação de termos utilizando tais conceitos é o diferencial semântico, originado de uma obra de Charles E. Osgood (HILL, 1958). Esse método estabelece que podemos medir o "sentimento" de uma pa-

lavra utilizando pares de adjetivos polares (bom-ruim, bonito-feio) e assinalando a sua posição relativa entre cada adjetivo, codificando-a pelos pares fornecidos.

Hoje, existem diversas maneiras de codificar textos e palavras e extrair diferentes tipos de informação de cada um deles: podemos utilizar a frequência das palavras em técnicas como *Bag of Words* (BoW), Análise Semântica Latente (LSA) e Alocação Latente de Dirichlet (LDA); ou modelos preditivos usando inteligência artificial para gerar representações automaticamente, como um dos métodos mais utilizados na atualidade que faz isso usando redes neurais: o Word2Vec (W2V). Neste capítulo, apresentamos o conceito de modelagem matemática de linguagens (seção 2.1) e a teoria por trás de algumas das representações disponíveis na literatura: estatísticas (seção 2.2) e neurais (seção 2.3).

2.1 Modelagem matemática de linguagens

Um modelo matemático de uma linguagem é uma distribuição estatística de probabilidades gerado a partir de um conjunto de sentenças e palavras. Ele consiste de duas partes:

1. Um conjunto finito V, que representa o vocabulário do corpus analisado;
2. Uma função de probabilidade tal que:
    - Para toda sentença/palavra W pertencente a V, $p(W) >= 0$;
    - O somatório das probabilidades é igual a 1.

A probabilidade de uma sentença é calculada a partir das probabilidades das palavras que a compõe. Com um modelo que calcula esta probabilidade, é possível encontrar as frases mais frequentes e quais sentenças têm maior similaridade com o conteúdo de um corpus. Pode-se, também, computar a probabilidade de uma palavra dadas as palavras anteriores a ela. O tamanho da frase anterior define uma janela de contexto (n-grama) para o cálculo de cada probabilidade, ou seja, em vez de usar uma sentença inteira para este cálculo, reduz-se o número de palavras a serem contadas e, com isso, aumenta-se a performance do modelo. A equação 2.1 mostra este cálculo.

$$p(w_1, \cdots, w_T) = \prod_i p(w_i \,|\, w_{i-1}, \cdots, w_{i-n+1}) \tag{2.1}$$

Esta probabilidade condicional pode ser calculada de acordo com a frequência de cada n-grama (equação 2.2).

$$p(w_t \mid w_{t-1}, \cdots, w_{t-n+1}) = \frac{count(w_{t-n+1}, \cdots, w_{t-1}, w_t)}{count(w_{t-n+1}, \cdots, w_{t-1})}. \tag{2.2}$$

Por exemplo, para calcular a probabilidade da palavra "nascente" dada a palavra anterior "sol" em um texto qualquer, teríamos que calcular a seguinte equação:

$$p(nascente \mid sol) = \frac{count(sol\ nascente)}{count(sol)}. \tag{2.3}$$

A probabilidade de "sol" e "nascente" ocorrerem em conjunto dependem da quantidade de ocorrências de "sol nascente" e da quantidade total de ocorrências da palavra "sol". Podemos aumentar a acurácia desses cálculos utilizando as ocorrências dos termos em outros textos em um determinado corpus ou utilizando a frequência relativa do termo em vez de sua contagem numérica.

Embora as pesquisas em representação distribuída tenham se popularizado nos últimos anos graças ao uso concorrente com redes neurais para produção automática de representações, existem trabalhos teóricos desde 1950 sobre a correlação de palavras e como usá-las para representação textual. Uma das primeiras técnicas que surgiram para medir a conotação de termos utilizando tais conceitos é o diferencial semântico, originado de uma obra de Charles E. Osgood (HILL, 1958). Este método estabelece que podemos medir o "sentimento" de uma palavra utilizando pares de adjetivos polares (bom-ruim, bonito-feio) e assinalando a sua posição relativa entre cada adjetivo, codificando-a pelos pares fornecidos.

Cada um dos métodos de representação textual utiliza um tipo de cálculo de probabilidade ou frequência em alguma parte de sua formulação. Alterando a distribuição, pode-se incluir informações no modelo úteis para generalização, para detecção de palavras e sentenças com frequências anormais, para lidar com relacionamentos semânticos de longa distância, etc. A seguir, são apresentadas algumas abordagens para geração de modelos de linguagem.

2.2 Modelagem de linguagem usando técnicas estatísticas

Cada texto tem características únicas, obtidas a partir de observação dos dados, e que podem ser representadas matematicamente para encontrar padrões e similaridades entre sentenças e documentos. A seguir são apresentados modelos de representação textual que utilizam características advindas de pesquisas linguísticas e estatísticas para mapear um espaço de palavras para um espaço numérico distribuído no qual cada palavra é representada por um vetor, permitindo operações matemáticas sobre estes vetores.

2.2.1 Bag of Words

O modelo *bag-of-words* (BoW) é uma representação textual simplificada, onde cada sentença (ou documento) é representada pelas palavras que estão presentes nela e em sua frequência, desconsiderando a ordem de ocorrência.

Nesse modelo, cada sentença é traduzida em um vetor, onde cada índice representa uma palavra do vocabulário, e o seu valor representa o número de vezes que esta palavra aparece na sentença. Esse valor pode ser o número total de ocorrências, ou a sua frequência relativa aos outros termos. Basicamente, esse modelo é um histograma de ocorrências de palavras.

Palavras muito comuns (preposições e artigos, por exemplo) podem ser desconsideradas, uma vez que suas altas frequências não trazem dados relevantes para esse tipo de representação e não definem uma sentença por sua presença. O processo de remoção de palavras que não trazem conteúdo relevante para a aplicação pretendida é denominado remoção de *stopwords*. Por exemplo, em um corpus com duas sentenças:

1. *hoje é um bom dia*
2. *bom dia para caminhar*

Após a remoção de *stopwords* (nesse exemplo, “um”, “é” e “para”), tem-se o vocabulário: [hoje, bom, dia, caminhar]. As representações das duas sentenças, usando BoW, são apresentadas na Tabela 2.1.

Tabela 2.1: Exemplo de vetores gerados usando a estratégia BoW

	hoje	bom	dia	caminhar
Sentença 1	1	1	1	0
Sentença 2	0	1	1	1

N-gramas também podem ser utilizados para adicionar informação semântica: nas sentenças acima, “bom” e “dia” estão em ambas as sentenças e sua junção poderia ser adicionada como um termo do vocabulário. Como esses termos também aparecem próximos um do outro, “bom dia” pode ser considerado como apenas uma palavra.

Entretanto, a contagem de palavras pode não ser a melhor maneira de se representar um texto, uma vez que uma sentença grande pode tratar do mesmo assunto que uma pequena, mas terá maior contagem de palavras pelo fato de ser maior: um texto de dez mil palavras e uma frase com cem palavras podem estar falando do mesmo assunto, mas a contagem de palavras será muito maior no texto do que na frase.

Para contornar esse problema, pode-se normalizar a frequência de cada palavra considerando sua frequência em todos os documentos. A maneira mais comum de se fazer isso é utilizando *term frequency – inverse document frequency* (TF-IDF) (SALTON; BUCKLEY, 1988). Essa medida estatística representa a importância de uma palavra (também chamada de "termo") em um documento, que aumenta quando essa palavra aparece muitas nesse documento, e diminui caso ela seja frequente em muitos outros documentos no corpus. Seu cálculo consiste na multiplicação da frequência de uma palavra (TF) com o inverso de sua frequência nos documentos (IDF), tendo diversas variações no cálculo de cada um desses fatores.

A TF de uma palavra dentro de um documento é usualmente calculada como o número de vezes que ela aparece nesse documento dividido pelo número total de palavras, ou seja, é a frequência relativa da palavra no documento. A IDF, representando a importância de uma palavra, tem um valor menor para as palavras frequentes em todos o documentos e valor maior para as palavras mais raras. Assim, os artigos e as preposições, que geralmente ocupam a maior parcela de uma sentença, tem valores reduzidos.

A forma mais usual de representar essa medida é usando a equação

$$\text{tf-idf}_{t,d} = (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t} \tag{2.4}$$

onde t é o termo, d é o documento, $tf_{t,d}$ é a frequência do termo t dentro do documento d, N é o número total de documentos e df_t, o número de documentos que contêm o termo t.

Em nosso exemplo anterior, podemos medir as pontuações de cada palavra utilizando o TF-IDF. Nesse caso, considerando que cada sentença é um "documento", calculamos o TF-IDF como ilustrado na equação 2.4 (a aplicação dessa equação no nosso exemplo é ilustrada nas Equações 2.5 e 2.6) e colocamos os resultados[1] na Tabela 2.2.

$$\text{tf-idf}_{hoje,sentenca1} = (1 + \log \text{tf}_{hoje,sentenca1}) \cdot \log \frac{2}{\text{df}_{hoje}} = (1 + \log 1) \cdot \log \frac{2}{1} \tag{2.5}$$

$$\text{tf-idf}_{dia,sentenca1} = (1 + \log \text{tf}_{dia,sentenca1}) \cdot \log \frac{2}{\text{df}_{dia}} = (1 + \log 1) \cdot \log \frac{2}{2} \tag{2.6}$$

A simplicidade e a robustez dessa medida permite que representações rápidas e esclarecedoras sejam obtidas em relação ao conteúdo e ao teor de cada documento, mas ela falha em passar informações sintáticas e semânticas. Tarefas que não precisam desses níveis de informação podem usar o TF-IDF para alcançar alta acurácia, como ocorre na recuperação de informação

[1] Resultados obtidos a partir da execução do algoritmo TF-IDF com o módulo Scikit-Learn (PEDREGOSA, 2012)

Tabela 2.2: Vetores BoW/TF-IDF de exemplo

	hoje	bom	dia	caminhar
Sentença 1	0,70490949	0,50154891	0,50154891	0,0
Sentença 2	0,0	0,50154891	0,50154891	0,70490949

com base no cálculo de similaridade de palavras e na análise de sentimentos. Entretanto, tarefas como a desambiguação semântica, que analisam a estrutura de uma sentença, precisam de mais do que frequências de termos para obter uma boa performance. Outro problema dessa medida é que a base de dados vetorizada pode ocupar um espaço muito grande: uma matriz de frequência de termo por documento pode chegar facilmente a dezenas de milhares de colunas e linhas, mesmo em um corpus pequeno.

2.2.2 Análise Semântica de Fatores Latentes (LSA)

Um modo de simplificar a representação matricial apresentada anteriormente é utilizando a decomposição em valores singulares (SVD) (ECKART; YOUNG, 1936), uma fatoração de matriz usada para obter uma representação de menor dimensionalidade para cada documento.

A motivação por trás dessa técnica é a redução do grande tamanho e redundância da matriz termo-documento gerada pelo modelo BoW: muitas das palavras têm uma sinonímia que ocupa espaço desnecessário (por exemplo, “paladino” e “herói”) ou uma polissemia que não é computada somente pela frequência de uma única palavra (por exemplo, “moda” pode ser a métrica estatística ou o conceito social de estilo, mas ambas as possibilidades contam como um único significado na contagem).

Para modelar uma linguagem levando em consideração essas diferenças, é preciso levar em conta a relação de co-ocorrência de palavras para encontrar as relações entre todas as palavras e remover informação irrelevante altamente correlacionada. A seguir, na seção 2.2.2.1, são revisados os fundamentos da álgebra linear usados nesse método antes de entrar na explicação e análise do algoritmo propriamente dito (seção 2.2.2.2).

2.2.2.1 Decomposição de Matrizes

Seja uma matriz A composta por números reais, com tamanho $M*N$. Em uma matriz termo-documento, todos os valores são não-negativos, já que não existe contagem negativa de palavras. O posto (*rank*) de uma matriz é determinado pelo seu número de colunas ou linhas linearmente independentes ($posto(A) <= min(M,N)$), representando a dimensão do espaço vetorial gerado pelas suas colunas. Uma matriz $r*r$ quadrada que tenha valores nulos em todas as posições

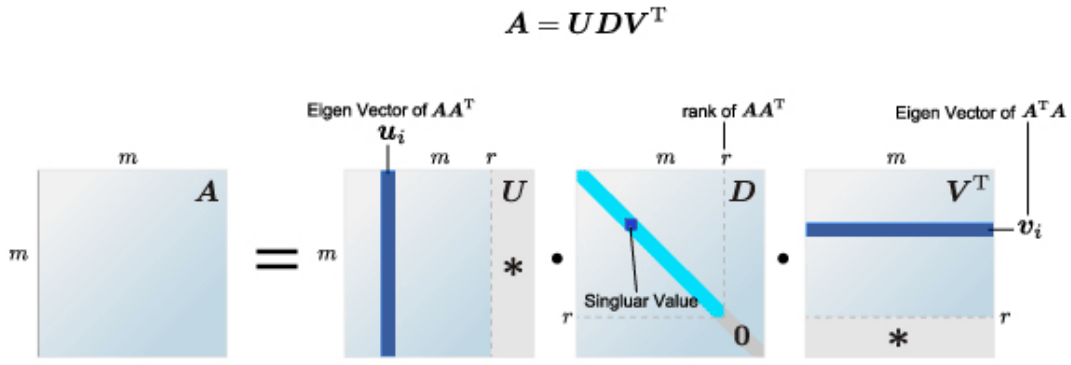
não-diagonais é uma matriz diagonal, com posto igual ao número de valores não-nulos. Os autovalores e autovetores de uma matriz A_{r*r} são determinados por λ e x tais que $Ax = \lambda x$, sendo x não-nulo.

Essa matriz A pode ser decomposta como $A = U\Sigma V^t$ (STEWART, 1993), onde

- U é uma matriz ortogonal[2] de tamanho $N * N$, com colunas formadas por autovetores de AA^t;
- V é uma matriz ortogonal de tamanho $M * M$, com colunas formadas por autovetores de A^tA;
- Σ é uma matriz diagonal $M * N$, com cada entrada diagonal sendo um valor singular (autovalor) de A.

Essa decomposição está ilustrada na Figura 2.1. Os autovalores e autovetores de uma matriz podem dizer muito sobre sua estrutura e sua distribuição numérica. Os maiores autovalores representam as colunas que mais adicionam à variância dos dados dentro da matriz original, ou seja, que têm menos correlação com as outras colunas (HERVÉ, 2007).

Figura 2.1: Ilustração da decomposição de uma matriz em seus valores singulares. A matriz D corresponde a matriz de



Fonte: https://www.numtech.com/Content/images/solutions/singular-value-decomposition.png

Para exemplificar esse conceito considere a matriz termo-documento descrita na Tabela 2.3, como a matriz base A (MANNING; RAGHAVAN, 2009). Cada coluna dessa matriz representa um documento, e os valores em suas linhas representam o número de vezes que cada palavra aparece no referido documento. No documento d_1, por exemplo, temos a presença das palavras "*ship*", "*ocean*" e "*voyage*".

[2]Matriz que tem a sua inversa sendo igual a sua transposta

Tabela 2.3: Matriz termo-documento A

	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
voyage	1	0	0	1	1	0
trip	0	0	0	1	0	1

Fonte: Retirado de (MANNING; RAGHAVAN, 2009).

A partir dessa matriz base, fazemos a equação $A = U\Sigma V^t$ e obtemos as três matrizes (MANNING; RAGHAVAN, 2009):

$$U = \begin{bmatrix} -0,44 & -0,30 & 0,57 & 0,58 & 0,25 \\ -0,13 & -0,33 & -0,59 & 0,00 & 0,73 \\ -0,48 & -0,51 & -0,37 & 0,00 & -0,61 \\ -0,70 & 0,35 & 0,15 & -0,58 & 0,16 \\ -0,26 & 0,65 & -0,41 & 0,58 & -0,09 \end{bmatrix}, \tag{2.7}$$

$$\Sigma = \begin{bmatrix} 2,16 & 0,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & 1,59 & 0,00 & 0,00 & 0,00 \\ 0,00 & 0,00 & 1,28 & 0,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 1,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 & 0,39 \end{bmatrix}, \tag{2.8}$$

$$V^t = \begin{bmatrix} \text{-}0,75 & \text{-}0,28 & \text{-}0,20 & \text{-}0,45 & \text{-}0,33 & \text{-}0,12 \\ \text{-}0,29 & \text{-}0,53 & \text{-}0,19 & 0,63 & 0,22 & 0,41 \\ 0,28 & \text{-}0,75 & 0,45 & \text{-}0,20 & 0,12 & \text{-}0,33 \\ 0,00 & 0,00 & 0,58 & 0,00 & \text{-}0,58 & 0,58 \\ \text{-}0,53 & 0,29 & 0,63 & 0,19 & 0,41 & \text{-}0,22 \end{bmatrix}. \tag{2.9}$$

2.2.2.2 Análise/Indexação Semântica Latente (LSI)

Para aproximar uma matriz termo-documento para uma representação de menor *rank* sem perder informações, utilizamos a decomposição por valores singulares discutida anteriormente. Criamos uma transformação linear que levará um vetor de frequência de palavras para um espaço de dimensionalidade reduzida, diminuindo o número de colunas da matriz e criando uma correlação entre os termos e conceitos do documento. Além de tratar dos problemas anterior-

mente discutidos, sinonímia e polissemia, diminuímos drasticamente a computação necessária para cálculos envolvendo esses vetores.

Para tal, a partir da matriz Σ da equação $A = U\Sigma V^t$ discutida na seção anterior, truncamos esta matriz utilizando somente os k maiores valores para produzir uma matriz C_k, com k dimensões e posto k. Essa matriz é projetada em um novo espaço dimensional, definido usando os autovetores relacionados a esses autovalores de CC^t e C^tC escolhidos, com a mínima perda de informação possível. Esse método é chamado de *Latent Semantic Indexing/Analysis* (LSI/LSA) (DEERWESTER; DUMAIS; HARSHMAN, 1990), muito usado para a comparação e recuperação de documentos.

Novos documentos são passados para esse novo espaço dimensional por meio da transformação

$$\vec{q}_k = \Sigma_k^{-1} U_k^T \vec{q},$$

multiplicando o inverso da matriz truncada de autovalores pela transposta da matriz U truncada e a representação original do documento. Podemos adicionar novos documentos ao espaço LSI dessa maneira, mas as novas relações de co-ocorrência não serão captadas dessa maneira, danificando sua qualidade: uma nova representação LSI deve ser re-computada para garantir a integridade da representação (DEERWESTER; DUMAIS; HARSHMAN, 1990).

Seguindo o exemplo da seção 2.2.2.1, temos a seguinte matriz de autovalores:

$$\Sigma = \begin{bmatrix} 2,16 & 0,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & 1,59 & 0,00 & 0,00 & 0,00 \\ 0,00 & 0,00 & 1,28 & 0,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 1,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 & 0,39 \end{bmatrix} \tag{2.10}$$

Escolhemos os dois maiores autovalores: $2,16$ e $1,59$. Os outros são zerados, obtendo a matriz truncada Σ_t:

$$\Sigma_t = \begin{bmatrix} 2,16 & 0,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & 1,59 & 0,00 & 0,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 & 0,00 \\ 0,00 & 0,00 & 0,00 & 0,00 & 0,00 \end{bmatrix} \tag{2.11}$$

Agora, refazemos a matriz termo-documento com a matriz truncada de autovalores, obtendo A_t (na Tabela 2.4), que é uma representação de menor dimensionalidade dos documentos representados em A.

Tabela 2.4: Matriz termo-documento A_t, com duas dimensões

	d_1	d_2	d_3	d_4	d_5	d_6
1	-1,62	-0,60	-0,44	-0,97	-0,70	-0,26
2	-0,46	-0,84	-0,30	1,00	0,35	0,65

Para encontrar a representação de um novo documento q, usamos a matriz de autovalores e a matriz U, ambas truncadas. A equação para obter esta representação é

$$q_k = q^t U_k \Sigma_k \tag{2.12}$$

Para nosso exemplo, uma representação de um novo documento q com os termos "*ship*", "*boat*" e "*ocean*" terá uma nova representação q_k:

$$\begin{aligned} q_k &= \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} -0,44 & -0,30 \\ -0,13 & -0,33 \\ -0,48 & -0,51 \\ -0,70 & 0,35 \\ -0,26 & 0,65 \end{bmatrix} \begin{bmatrix} 2,16 & 0,00 \\ 0,00 & 1,59 \end{bmatrix} \\ &= \begin{bmatrix} -2,268 & -1,812 \end{bmatrix} \end{aligned} \tag{2.13}$$

O método LSI se comporta como um algoritmo de compactação, reduzindo o tamanho do espaço vetorial gerado pelos dados e removendo informação desnecessária (ruído) no processo. Este foi um dos métodos de representação textual investigados neste trabalho, como explicado no Capítulo 6.

2.2.3 Alocação Latente Dirichlet (LDA)

Latent Dirichlet Allocation (LDA) (BLEI; NG; JORDAN, 2003) é um método que cria um modelo estatístico generativo descrevendo documentos partindo da premissa de que qualquer texto é uma mistura de tópicos, e de que cada palavra pode ser atribuída a um desses tópicos. Como no método LSA descrito anteriormente, mapeamos uma matriz termo-documento para um espaço de menor dimensionalidade definido por diferentes tópicos. Mas, ao invés de usar as características intrínsecas da matriz para sua decomposição em conceitos, nesse método são usadas as distribuições probabilísticas de cada documento do corpus.

Usando o conceito visto anteriormente na Análise Semântica Latente, podemos modelar probabilisticamente a ocorrência de uma palavra em um documento usando a equação

$$\mathsf{P}(w \mid d) = \mathsf{P}(d) \sum_{c} \mathsf{P}(c \mid d)\mathsf{P}(w \mid c), \tag{2.14}$$

onde d representa um documento, c é um tópico e w é uma palavra. A probabilidade de uma palavra ocorrer dado um documento, $\mathsf{P}(w \mid d)$, é calculada a partir de duas distribuições multinomiais[3]:

- $\mathsf{P}(c \mid d)$: probabilidade de um tópico c pertencer a um documento d;
- $\mathsf{P}(w \mid c)$: probabilidade de uma palavra w ser gerada por um tópico c.

Essa equação representa o método probabilístico *Latent Semantic Indexing/Analysis* (pLSA) (HOFMANN, 1999), desenvolvido a partir do LSA com bases matemáticas e estatísticas. O problema, aqui, é estimar as probabilidades de tópicos em documentos que não pertencem ao corpus: como a escolha de tópicos para cada documento está ligada aos documentos previamente vistos, novos conceitos não são passados para a representação.

Partindo dessa equação, o método LDA busca solucionar esse problema criando um modelo que gera a distribuição de tópicos para cada documento na coleção, e a estrutura desses tópicos é inferida de acordo com o processo inverso. Cada uma das distribuições utilizadas tem seus parâmetros estimados de acordo com a base de dados que ela irá projetar. Para cada documento no corpus:

1. Uma distribuição de tópicos Θ é escolhida aleatoriamente a partir de uma distribuição Dirichlet[4].
    - Em um modelo com três tópicos, uma distribuição possível para um dado documento é 30% para o primeiro, 60% para o segundo e 10% para o terceiro. Estas probabilidades são utilizadas como parâmetros para a distribuição Dirichlet.
2. Para cada tópico, uma distribuição de palavras por tópicos é escolhida aleatoriamente a partir de outra distribuição Dirichlet.
    - Dado um tópico T e palavras W_1, W_2, W_3, uma distribuição possível para as probabilidades de W_n pertencerem ao tópico T pode ser de 70% para W_1, 20% para W_2 e 10% para W_3.

[3] generalização da probabilidade binomial, onde cada evento pode ter n diferentes resultados.
[4] distribuição parametrizada utilizando números reais para inicialização

3. Para cada palavra no documento:

   (a) Um tópico c é escolhido a partir da distribuição Θ escolhida anteriormente;

   (b) Uma palavra w é escolhida a partir da distribuição do tópico c escolhido.

O modelo é, então, treinado com a estimação dos parâmetros das distribuições, usando o corpus como base de dados inicial. O vetor de probabilidades de tópicos gerado a partir desse modelo pode ser usado posteriormente como uma representação conceitual do texto, e é muito usado para calcular a similaridade entre documentos e em recuperação de informação.

Para ilustrar a aplicação desse método, considere as cinco diferentes sentenças em um *corpus*:

- d_1: Hoje o clima estará instável pela manhã, com risco de pancadas de chuva.
- d_2: A umidade do ar estará com níveis muito baixos durante a noite.
- d_3: Existe uma correlação entre o nível de óleo de um carro e acidentes em estradas.
- d_4: Os freios de um carro devem passar por manutenção preventiva todo ano.
- d_5: A neve e a chuva prejudicam a visibilidade dos carros na rodovia, cuidado é necessário nas próximas semanas ao viajar.

Digamos que existem dois tópicos, “clima” e “carros”. As distribuições estatísticas das palavras que têm o mesmo tópico tendem a ser similares: como “carro” e “freios” ocorrem na mesma frase e “carro” e “estradas” em outra, as palavras “freios” e “estradas” podem ser relacionadas semanticamente. Nós, humanos, sabemos que existem dois tópicos separáveis dentro desse corpus.

Inicializamos o algoritmo com o número de tópicos ($K = 2$). Um tópico escolhido aleatoriamente é atribuído a cada palavra, dando as distribuições iniciais de palavras sobre tópicos, e de tópicos sobre documentos. Na Tabela 2.5, temos um exemplo de como essa escolha inicial dos tópicos pode ocorrer.

Agora, treinamos as distribuições. Para cada palavra w em documento d:

1. Calcular a probabilidade da palavra w pertencer a cada tópico existente, e a probabilidade de cada tópico aparecer no documento d.

Tabela 2.5: Exemplo de uma distribuição inicial de tópicos nas sentenças d_1 e d_5

	d_1		d_5
T_1	clima	T_2	neve
T_2	manhã	T_1	chuva
T_1	pancadas	T_2	carros
T_2	chuva	T_2	rodovia

2. Rever a atribuição da palavra ao tópico com base nas probabilidades adquiridas anteriormente: no quanto a palavra é frequente em um tópico e em como os tópicos estão distribuídos no documento.

Considerando o exemplo, se escolhermos a palavra "chuva" no documento d_5:

- Procuramos a probabilidade da palavra pertencer a um tópico. Em d_1, ela está associada ao tópico 2 (T_2). Como estamos computando a atribuição da palavra "chuva" no documento d_5, não utilizamos a sua atribuição anterior ao tópico 1. Então, essa palavra tem 100% de chance de pertencer ao tópico 2.
- Também vemos a prevalência de tópicos no documento d_5. Há três palavras atribuídas ao tópico 2, e uma ao tópico 1. Então, há 25% de probabilidade de "chuva" pertencer ao tópico 1, e 75% de pertencer ao tópico 2.

Simplificando os cálculos, consideremos apenas essas probabilidades sem contar com parâmetros relevantes às distribuições. A multiplicação das duas probabilidades (uma palavra pertencer a um tópico, $\mathsf{P}(w \mid c)$, e um tópico pertencer a um documento, $\mathsf{P}(c \mid d)$) nos revela qual tópico é o mais indicado para atualizar a escolha. No caso da palavra "chuva" do documento d_5, o tópico escolhido é o 2, alterando sua escolha inicial. O processo é repetido até a distribuição de tópicos se estabilizar, ou seja, até que nenhuma alteração de atribuição seja realizada. Um novo documento pode analisado com base nas distribuições de cada palavra que o compõe, e sua representação calculada com base na pertinência a um tópico ou a outro.

O LDA foi um dos métodos de representação textual investigados neste trabalho, como explicado no Capítulo 6.

2.2.4 GloVe

Um modelo estatístico mais recente foi proposto por Pennington, Socher e Manning (2014) utilizando as matrizes de co-ocorrência de palavras para definir um mapeamento de palavras em um espaço vetorial de menor dimensionalidade.

Primeiro, define-se uma matriz X de co-ocorrência de palavras: $X_{i,j}$ representa a frequência com que a palavra i aparece próxima à palavra j. Para tal, é preciso uma leitura do corpus inteiro. Então, constrói-se uma função para criar os vetores $\vec{w}_i$ e $\vec{w}_j$ com base nos valores de $X_{i,j}$:

$$\vec{w}_i^T \vec{w}_j + b_i + b_j = \log X_{i,j} \tag{2.15}$$

onde b_i e b_j representam o viés de cada palavra no corpus, sua frequência no corpus. Agora, podemos criar uma função-objetivo a ser minimizada, utilizando a equação anterior:

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f\left(X_{ij}\right) \left(\vec{w}_i^T \vec{w}_j + b_i + b_j - \log X_{ij}\right)^2 \tag{2.16}$$

onde f é uma função para normalizar co-ocorrências muito frequentes. Minimizando esse valor J, conseguimos os vetores $\vec{w}$ para todas as palavras da base de dados.

Para exemplificar como funciona esse método, vamos analisar uma tabela de probabilidades (ilustrada na Figura 2.2) de algumas palavras em um corpus de aproximadamente seis bilhões de palavras. Esse exemplo foi retirado do artigo original de proposição do modelo (PENNINGTON; SOCHER; MANNING, 2014).

Figura 2.2: Tabela de probabilidades com palavras selecionadas de um corpus com 6 bilhões de termos.

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Fonte: Retirado de (PENNINGTON; SOCHER; MANNING, 2014)

Os valores representados nessa tabela indicam que "*ice*" (gelo) e "*solid*" (sólido) ocorrem no mesmo contexto com maior frequência do que "*ice*" e "gas" (gás), já que o valor da probabilidade de $\mathrm{P}(solid \mid ice)$ é maior do que $\mathrm{P}(gas \mid ice)$. As duas palavras são igualmente frequentes com a palavra "*water*" (água), e tem menor correlação com "*fashion*" (moda).

Agora, ao observar a razão entre duas probabilidades (ilustrada na última linha da Tabela 2.2), podemos notar propriedades interessantes:

- Valores maiores que 1 são relacionadas à palavra que está no numerador da razão. No exemplo, "*ice*" e "*solid*" são mais relacionados do que "*steam*" e "*solid*", uma vez que a razão entre as probabilidades tem um valor alto.
- Valores menores que 1 são muito relacionadas à palavra que está no denominador da razão. No exemplo, "*steam*" e "*gas*" estão mais relacionados entre si do que "*ice*" e "*gas*", uma vez que a razão entre as probabilidades tem um valor baixo.
- A magnitude destes valores obtidos representa a intensidade da relação entre o termo no numerador/denominador e o termo sendo comparado.

O objetivo de treinamento do método GloVe (retratado na equação 2.16) é aprender vetores que tenham um produto escalar igual ao logaritmo das probabilidade de co-ocorrência das palavras, associando as razões das probabilidades de co-ocorrência com a distância entre vetores no novo espaço vetorial mapeado. Essa função-objetivo pode ser minimizada com qualquer algoritmo, seja neural ou linear (PENNINGTON; SOCHER; MANNING, 2014).

O GloVe foi um dos métodos de representação textual investigados neste trabalho, como explicado no Capítulo 6.

2.3 Modelagem de linguagem usando modelos neurais

Os métodos estatísticos, mesmo sendo robustos, simples e funcionais na maioria dos casos, não representam totalmente as nuances de um texto e sofrem com problemas como: a dimensionalidade elevada do vocabulário e a quantidade de dados a serem processados. Seguindo a abordagem estatística, nem sempre um vetor terá maior representatividade aumentando o número de características utilizadas para lhe compor.

Os modelos preditivos tentam prever uma palavra dado o seu contexto, codificando-a de acordo com seus vizinhos. Utilizando algoritmos da área de Aprendizado de Máquina, esses modelos têm como objetivo aprender as minúcias de um texto automaticamente. Assim, pode-se encontrar representações de baixa dimensionalidade que ainda possuem representatividade alta sem a necessidade de interferência humana (BARONI; DINU; KRUSZEWSKI, 2014).

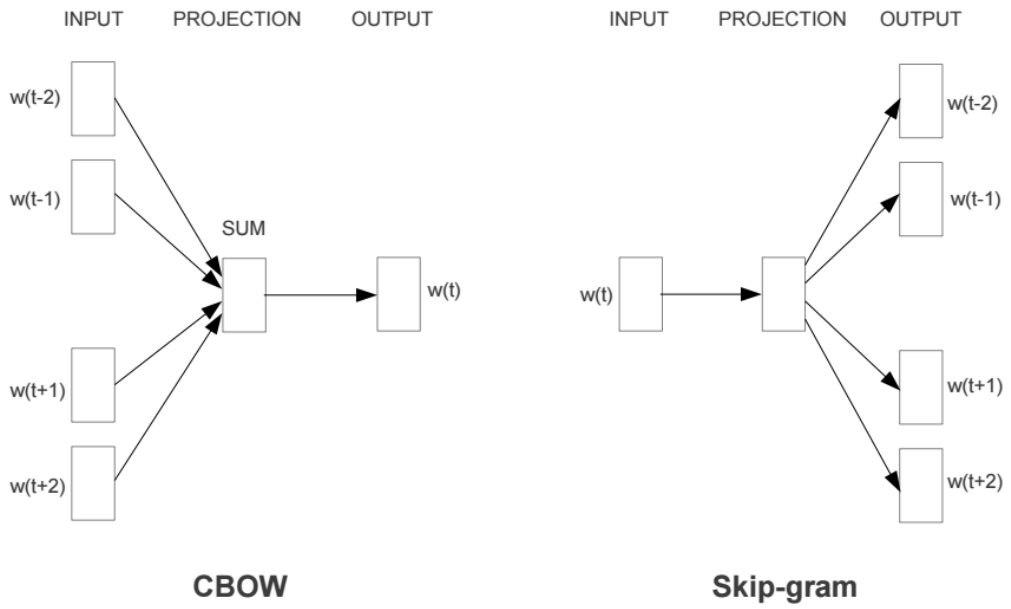
Com os avanços recentes na performance geral dos computadores e o uso de placas gráficas para otimizar cálculos matriciais, métodos mais complexos envolvendo redes neurais se popularizaram no meio acadêmico, obtendo resultados promissores em diversas áreas da Inteligência Artificial. Representações vetoriais de alta qualidade podem ser obtidas a partir de métodos não-supervisionados usando redes neurais, encontrando relações semânticas entre palavras.

Dois métodos recentes para representação textual foram utilizados neste trabalho: o Word2Vec (apresentado a seguir na seção 2.3.1) e o GloVe (já apresentado na seção 2.2.4). O GloVe, mesmo não sendo uma abordagem que utiliza diretamente redes neurais, tem uma função-objetivo similar à do Word2Vec. Na seção 2.3.1.1, apresentamos uma breve discussão sobre a comparação entre os dois métodos e quais suas semelhanças e diferenças.

2.3.1 Word2Vec

O modelo Word2Vec foi proposto por Mikolov (2013) e se popularizou como um dos meios mais rápidos e simples de se obter uma representação textual de alta qualidade. Derivado do modelo neural de linguagem inicial de Bengio (2003), para mapear uma palavra em um vetor, o Word2Vec utiliza uma camada intermediária de uma rede neural treinada para identificar uma palavra dado o seu contexto (*Continuous Bag of Words*, CBoW) ou um contexto dada uma palavra (*Skip-Gram*). Esses dois modelos estão ilustrados na Figura 2.3

Figura 2.3: Modelo Skip-Gram e CBoW lado a lado. O primeiro prevê o contexto dada uma palavra, o segundo faz o inverso.

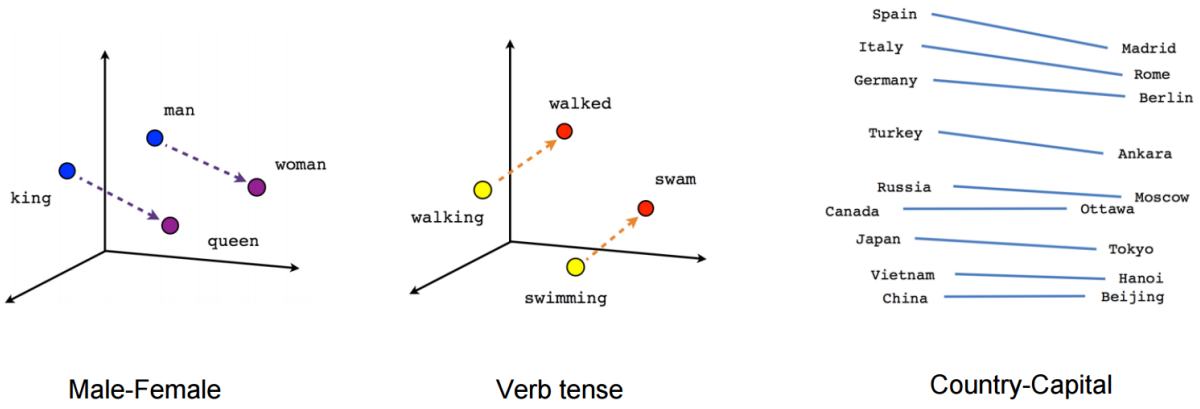


Fonte: Retirado de (MIKOLOV, 2013)

Ao passar uma palavra por essa rede neural, o resultado da camada intermediária será a projeção dela em um espaço vetorial. Sua posição será condicionada ao seu contexto, com palavras semanticamente relacionadas tendo menor distância vetorial. Por exemplo, em um documento sobre música, o vetor representando "canto" estará próximo de "música", "gregoriano" e "tom"; enquanto que em um documento sobre *design* de interiores, o mesmo vetor

estará próximo de “mesa”, “cômoda” e “poltrona”. Na Figura 2.4, é possível observar alguns relacionamentos interessantes em um modelo treinado pelo Word2Vec.

Figura 2.4: Relações entre palavras no espaço vetorial



Fonte: https://www.tensorflow.org/versions/r0.11/tutorials/word2vec/index.html

O contexto de uma palavra é dado por uma janela de tamanho N contendo as N palavras que a cercam. Por exemplo, na sentença “hoje é um bom dia”, o contexto da palavra “um” pode ser dado por “é” e “bom” quando $N = 1$, e por “hoje”, “é”, “bom” e “dia” quando $N = 2$.

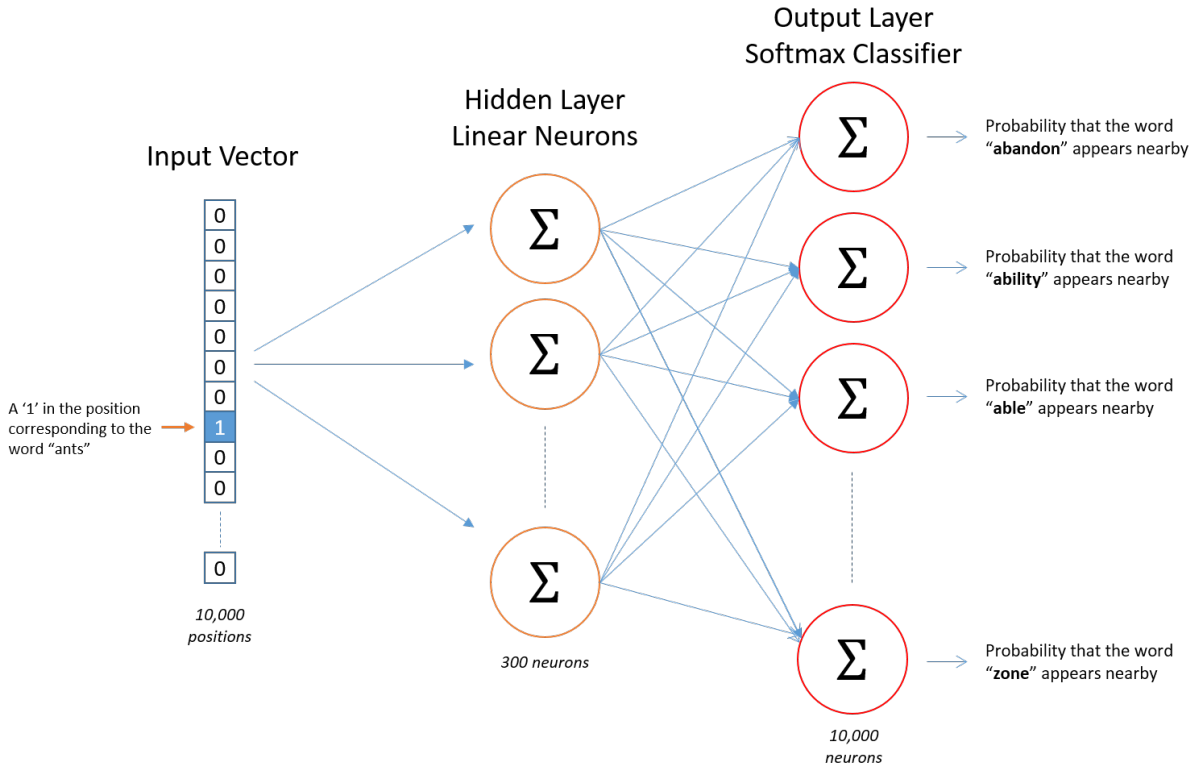
Para treinar essa rede, cada palavra é traduzida em um vetor com o tamanho do vocabulário de seu corpus (simbolizado por V). Deve-se escolher também o modelo. No CBoW, cada palavra no contexto de um termo (w(t-n), ..., w(t-1), w(t+1), ..., w(t+n)) é utilizada para lhe predizer (w(t)), com a entrada sendo os $N-1$ vetores contextuais e o objetivo sendo um vetor de tamanho V com a probabilidade de cada palavra ser a escolhida, representando a distribuição estatística do termo central. No Skip-Gram (ilustrado na Figura 2.5), cada palavra (w(t)) tenta prever o seu contexto (w(t-n), ..., w(t-1), w(t+1), ..., w(t+n)), com a entrada sendo o vetor da palavra e a saída tendo múltiplos vetores de probabilidade, representando a distribuição estatística de cada palavra ocorrer próxima a w(t).

A diferença na tarefa proposta pela rede neural interfere diretamente nas informações codificadas pela sua camada intermediária. Assim:

- Com o CBoW, cada contexto é uma única instância de dados, suavizando a distribuição de palavras e diminuindo a quantidade de instâncias disponíveis para treinamento. Palavras raras tendem a ser ignoradas pelo modelo CBoW, pois essa suavização é alcançada fazendo a média dos vetores do contexto de uma palavra para prevê-la: as palavras raras se misturam com as palavras de seu contexto e não têm tanta influência no resultado final.

- Com o Skip-Gram, múltiplas instâncias de treinamento podem ser geradas aumentando a janela que define o contexto de uma palavra e trocando as possíveis escolhas de palavras. Mais dados podem passar mais informação para a representação, e melhores vetores podem ser obtidos ao custo de performance e tempo de treinamento.

Figura 2.5: Representação de uma rede neural com o modelo Skip-Gram.



Fonte: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/

O Word2Vec (W2V) Skip-Gram foi um dos métodos de representação textual investigados neste trabalho, como explicado no Capítulo 6.

2.3.1.1 Comparando Word2Vec e GloVe

Os dois métodos comparados nesta seção, Word2Vec e GloVe, geram representações de alta qualidade utilizando diferentes maneiras para chegar ao seu objetivo: o Word2Vec usa uma rede neural rasa para codificar automaticamente palavras baseadas em seu contexto; o GloVe usa as probabilidades de co-ocorrência de palavras para modelar um espaço vetorial.

Berardi, Esuli e Marcheggiani (2015) comparam os dois métodos para a tarefa de analogia entre palavras utilizando um corpus italiano, com o modelo Word2Vec sendo o melhor na maioria dos casos. Os autores teorizam que a complexidade semântica e sintática da língua italiana

contribuiu para esse resultado, aliada à má escolha de hiper-parâmetros para o treinamento do GloVe. Como a tarefa é simples, muitos experimentos e diferentes tarefas de analogia foram utilizadas: encontrar capitais relacionadas a cidades, moedas relacionadas a países, palavras e seus plurais, etc. Mas, pela tarefa ser simples, não podemos utilizar esses resultados para comparações a tarefas práticas, como classificação e cálculo de similaridade de textos.

O próprio artigo do GloVe (PENNINGTON; SOCHER; MANNING, 2014) traz algumas comparações com outros tipos de representações, incluindo o modelo Word2Vec, e mostra resultados superiores na maioria dos casos. O artigo cita também que uma comparação rigorosa é difícil de se obter, pois há muitos hiper-parâmetros a serem ajustados: os dois modelos podem obter acurácia semelhante por meio de sucessivos ajustes. Os experimentos realizados usaram os parâmetros padrões do Word2Vec para a representação do corpus de teste, o que pode ter influenciado negativamente no resultado.

Levy, Goldberg e Dagan (2015) dá mais importância à escolha de hiper-parâmetros do que ao algoritmo escolhido, apontando que os diversos métodos disponíveis para a representação distribuída de palavras recentes têm uma dependência muito grande em relação aos parâmetros de seu treinamento: o tamanho da janela de contexto, a utilização de palavras com poucas ocorrências, as medidas para avaliar a relação de uma palavra com seu contexto e a normalização dos vetores de pesos em redes neurais são exemplos de decisões que podem alterar a representação final gerada por qualquer modelo.

Em termos de desempenho computacional, o tempo de processamento do GloVe é menor do que o do Word2Vec: o treinamento neural demanda tempo e não pode ser paralelizado, enquanto o cálculo da matriz de co-ocorrência pode ser feito de maneira paralela. No método GloVe, podemos rapidamente criar novas representações de diferentes dimensionalidades utilizando a matriz de co-ocorrências previamente obtida, enquanto que no método Word2Vec o processo de treinamento precisa ser repetido para alterar o número de dimensões da representação. Como o método GloVe requer a matriz total de co-ocorrência em memória, ele demanda mais espaço de armazenamento quando comparado com o Word2Vec.

Capítulo 3

REPRESENTAÇÃO DISTRIBUÍDA DE IMAGENS

3.1 Introdução

Imagens são modos de informação ricos em detalhes, expressando dados sobre o mundo físico por meio de uma representação visual virtual, uma foto ou desenho que foi criado/copiado e, no contexto deste trabalho, armazenado em forma eletrônica. Elas podem ser descritas de maneira vetorial ou rasterizada. A versão rasterizada produz um *bitmap*, uma matriz de tuplas de números representando as intensidades das cores primárias que compõem cada um dos pontos (*pixels*). Essa matriz pode ser considerada a representação distribuída de uma imagem por si só, com cada *pixel* sendo uma dimensão de um vetor com tamanho $w * h$, onde w representa a largura (*width*) e h a altura (*height*) da imagem.

Os problemas da vida real que a Visão Computacional tenta resolver exigem grande quantidade de detalhes para soluções satisfatórias e têm alta complexidade. Alinhamento e classificação de cenas, segmentação de imagens, reconstrução tridimensional e reconhecimento de objetos são exemplos de tarefas onde a quantidade e a qualidade da informação extraída de uma única imagem se torna essencial para um resultado aceitável.

Muitos componentes existem em uma única imagem e devem ser analisados individualmente. Formas geométricas podem ser comuns entre imagens, mas a sua posição, cor e tamanho influeciam na sua representação. *Pixels* vizinhos estão fortemente correlacionados, devido à regularidade de cores em objetos reais. O tamanho da matriz de *pixels* faz com que utilizar o dado puro seja inviável em termos de performance de processamento e da redundância de dados.

As pesquisas sobre representação de imagens e seu uso tiveram início na década de 1960, em universidades que já tinham polos de pesquisa em inteligência artificial. Em 1966, no *Mas-*

sachusetts Institute of Technology (MIT), um projeto de verão buscava acoplar uma câmera a um computador para a descrição textual de imagens (PAPERT; TECHNOLOGY. ARTIFICIAL INTELLIGENCE LABORATORY, 1966). Na década de 1970, as técnicas mais comuns da área foram delineadas: as bases para os algoritmos de extração de bordas, identificação de linhas, modelagem poliédrica e detecção de movimento foram definidas nesse período (SZELISKI, 2010). Na década de 1980, estudos em áreas teóricas e matemáticas tiveram mais destaque. Um exemplo é o conceito de espaço escalar (*scale-space*), onde uma imagem é representada por múltiplas cópias suavizadas por diferentes *kernels* (WITKIN, 1983), que nasceu nessa época e é utilizado até hoje por diversos algoritmos. Na década de 1990, estudos em reconstruções tridimensionais de imagens por múltiplas fotos (HOPPE, 1992), segmentação de objetos (NIKHIL; SCANSAR, 1993) e reconhecimento de faces (TURK; PENTLAND, 1991) foram lançados. Logo em seguida as conexões com a área de Computação Gráfica foram reforçadas com estudos em renderização e transformação de imagens, interpolação e junção panorâmica de imagens (SZELISKI, 2010).

Para analisar e descrever uma imagem, temos que trabalhar a partir da matriz de *pixels* que a representa e encontrar características que a definam unicamente por meio de vetores de menor dimensionalidade[1]. Esse processo é chamado de extração de características e pode ser realizado seguindo duas abordagens: supervisionada ou não-supervisionada. Na abordagem supervisionada, as representações são criadas com base em atributos e descritores construídos por humanos. Na maioria dessas técnicas a diferença entre gradientes de intensidade de *pixels* vizinhos é utilizada para delinear regiões de interesse na imagem, encontrando áreas que contêm informação relevante. Já as técnicas não-supervisionadas buscam encontrar as características mais relevantes autonomamente para cada banco de imagens. Para tanto, podem utilizar, por exemplo, redes neurais e suas camadas intermediárias para gerar representações de acordo com os atributos inerentes de cada imagem.

Neste capítulo, descrevemos o que define uma característica relevante em uma imagem, além de mostrar os métodos mais comuns de extração e enriquecimento de informação existentes na área.

3.2 Técnicas de Extração de Características

Geralmente, dados não se apresentam estruturados, prontos para serem analisados. Sua redundância, complexidade e volume dificultam tentativas de compreender os sutis padrões intrínsecos de cada banco de dados. Para tal, precisamos encontrar, quantificar e discriminar

[1] Assim, representações distribuídas de texto (como visto no capítulo anterior) e de imagem têm o mesmo objetivo.

as numerosas características de cada instância de dado. Podemos relacionar tal processo à redução de dimensionalidade, também citada no capítulo 2: reduzimos a complexidade de um dado ao extrair os seus dados essenciais. Transforma-se um dado em um vetor de dimensionalidade reduzida (um vetor de características, ou *feature vector*), para que seja usado no lugar da informação original com perda tolerável de significado. A extração de características é um termo genérico que se refere aos inúmeros métodos de construção, combinação e discretização de *feature vectors* que sejam suficientemente representativos dos dados originais.

A aplicação deste tipo de processo é vital na área de processamento de imagem, extraindo as informações mais importantes de matrizes com muita correlação entre *pixels* e de alta dimensionalidade. Podemos extrair informações de tamanho, forma, textura, cor e posição de objetos em uma cena para descrevê-la com base em suas partes.

Nas seções seguintes, cobriremos os métodos de extração mais comuns e interessantes da área no contexto deste trabalho. Começamos com características de baixo nível, como identificação de linhas, bordas e regiões de interesse (seção 3.2.1), e nos aprofundamos em maneiras de descrever tais características em função de seus atributos: descritores manuais como o SIFT (seção 3.2.2), SURF (seção 3.2.3) e ORB (seção 3.2.4) e automáticos, como os produzidos por redes neurais profundas (seção 3.2.6) e, em especial, *autoencoders* (seção 3.2.5).

3.2.1 Características de uma Imagem

Para encontrar regiões de interesse dentro de uma imagem, primeiro devemos delinear quais são as características mais importantes dentro desta. Por exemplo, na figura 3.1, quais são os atributos que podem definir unicamente esta foto dentro de um conjunto de dados? Podemos extrair a silhueta do Everest, que tem um formato singular dentro da cadeia de montanhas onde ele reside. Também podemos utilizar o formato que esta cadeia de montanhas deixa no horizonte, suas bordas em relação ao céu azul. A composição das cores também pode ser usada para indicar local, clima e tipo de cena. Podemos também encontrar informações de perspectiva por meio de linhas e pontos de fuga.

Para uma análise mais profunda, o primeiro passo é encontrar pontos de referência que nos tragam regiões com informação relevante: cantos de uma forma geométrica ou, como na imagem 3.1, o *pixel* no pico do Everest. Um ponto de interesse de uma imagem tem as seguintes propriedades:

- Está bem definido matematicamente, focado e visível;

Figura 3.1: Imagem do pico do Everest.

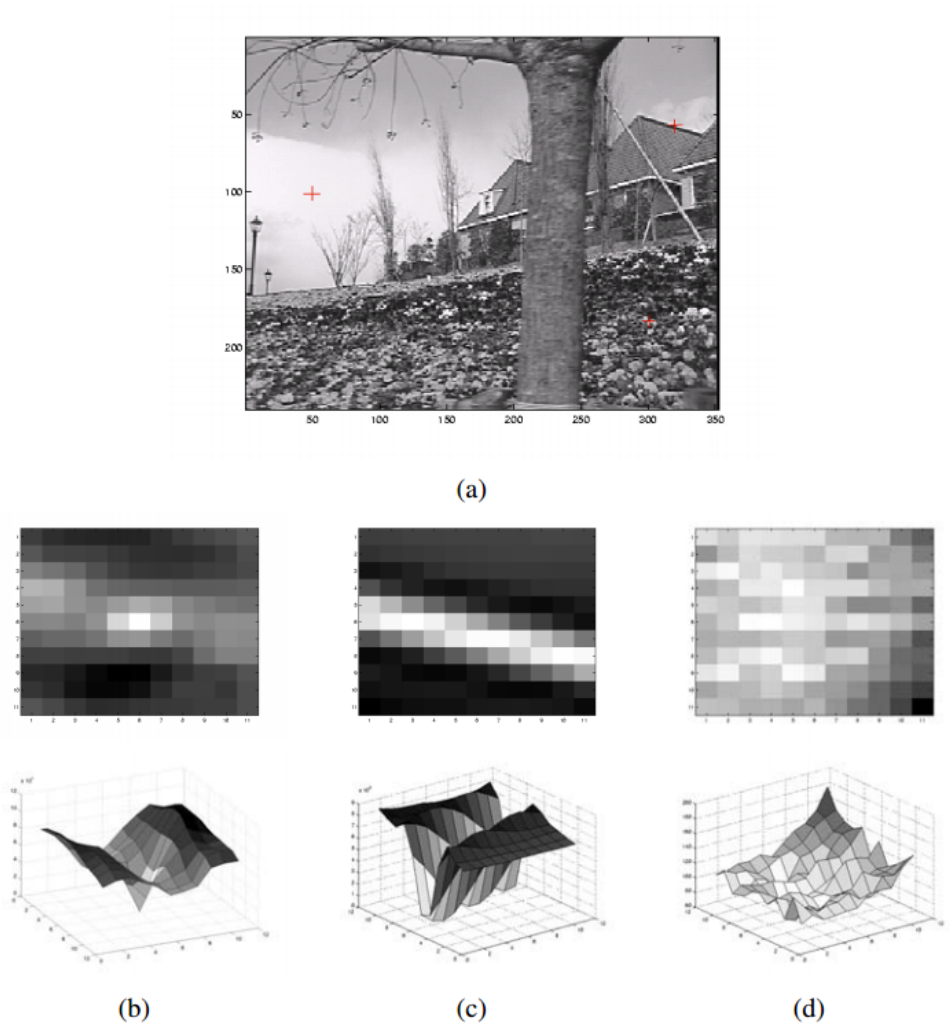
Fonte: Retirado de https://en.wikipedia.org/wiki/Mount˙Everest

- Está em uma posição bem definida na imagem;
- Os *pixels* na vizinhança deste ponto são ricos em termos de informação;
- Pode ser encontrado em diversos lugares, tanto localmente (na própria imagem) como globalmente (nas outras imagens do seu domínio).

Pontos em regiões sem textura (como, por exemplo, o céu) não trazem informação interessante: a parte superior da imagem pode ser descrita inteiramente por uma única tonalidade de cor azul. Mesmo que muitas outras fotos tenham o céu ao fundo, não há informação que identifique pequenos pedaços do céu. Em termos matemáticos, pontos interessantes em uma imagem têm alta diferença de intensidade entre os *pixels* de sua vizinhança. Na figura 3.2 podemos ver gráficos representando as intensidades de cor de cada *pixel* ao redor de pontos selecionados na imagem (a), onde grandes variações indicam quais pontos têm alta representatividade de informação. Para encontrar todos os pontos de interesse de uma imagem, podemos simplesmente passar por todos os pedaços (*patches*) possíveis da imagem e coletar todos os pontos que apresentam mudança abrupta de intensidade de cor. Os gráficos (b) e (c) ilustram as intensidades dos pontos inferior direito e superior direito na imagem, respectivamente, onde podemos notar diferença razoável entre os picos da vizinhança e do ponto central. O gráfico (d) não apre-

senta picos, mínimos ou mudanças bruscas interessantes no seu gráfico de intensidades. Assim, os pontos denotados por (b) e (c) são pontos de interesse, e (d) não é.

Figura 3.2: (a): Foto com pontos de interesse denotados por cruzes vermelhas. (b): Ponto inferior direito. (c): Ponto superior direito. (d): Ponto superior esquerdo.



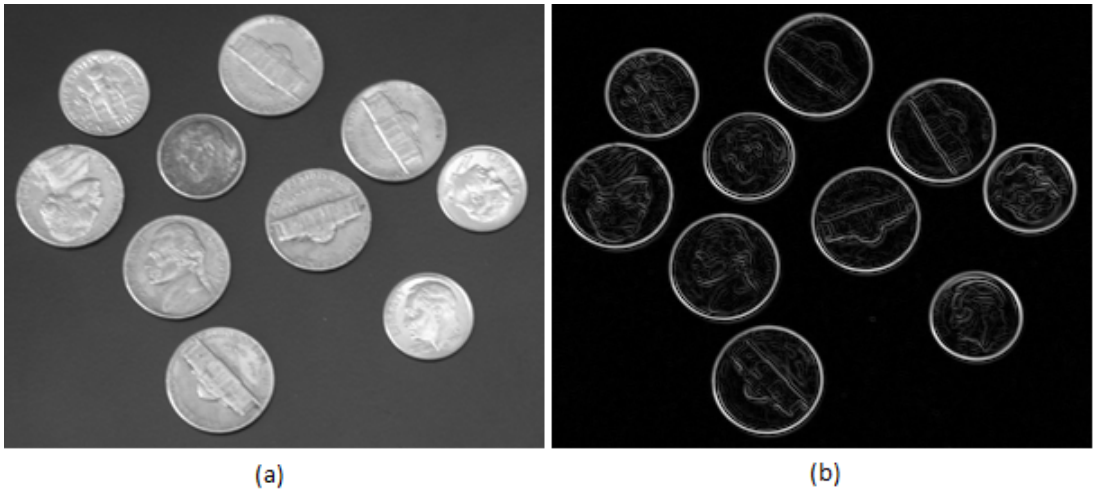
Fonte: Retirado de Szeliski (2010).

Podemos inferir a mudança direcional na cor ou intensidade de uma imagem através de cálculos sobre uma matriz de *pixels* qualquer. Esta mudança pode ser mensurada e é chamada de *gradiente*. Matematicamente, o gradiente de uma função de duas variáveis é um vetor de duas posições, preenchido pelas suas derivadas verticais e horizontais. Em termos de imagem, este vetor de gradiente aponta para a direção de maior variação de intensidade de cor, e o seu módulo representa a força da mudança.

Na figura 3.3, podemos ver a intensidade dos gradientes em cada *pixel* de uma imagem retratando moedas em uma superfície. As bordas das moedas são os locais onde os gradientes têm módulo elevado, tendo destaque na imagem à direita. Os detalhes das moedas também

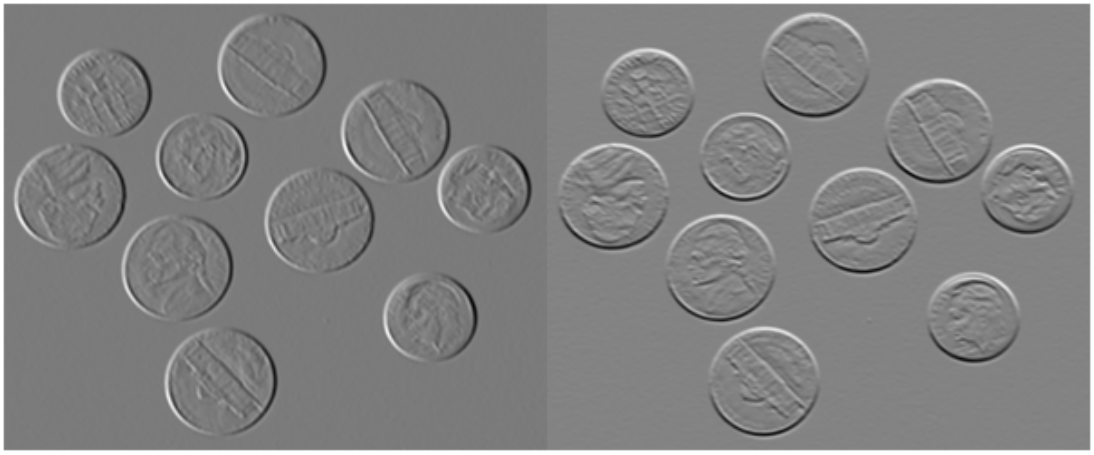
podem ser vistos, com menor força. Quanto maior o gradiente, mais branco é o *pixel* nesta visualização. Na figura 3.4, os componentes vertical e horizontal do vetor de gradientes são retratados: somados, criam o mapa de intensidade de gradiente visto na figura 3.3. Na figura 3.4, tons claros indicam valores positivos, enquanto tons escuros, valores negativos.

Figura 3.3: (a): Imagem original. (b): Mapa de intensidade de gradientes da imagem (a).



Fonte: Retirado de https://www.mathworks.com/help/images/ref/imgradient.html em 07/01/2017

Figura 3.4: Mapa de gradientes direcionais, horizontal (esquerda) e vertical (direita).



Fonte: Retirado de https://www.mathworks.com/help/images/ref/imgradient.html em 07/01/2017

Os gradientes de uma imagem podem revelar outras informações sobre uma imagem, criando formatos bem delineados para posterior análise e computação. Podemos obter tais mapas de gradiente fazendo a convolução de uma imagem com um filtro pré-estabelecido (por exemplo, o filtro Sobel (SOBEL; FELDMAN, 1973))[2]. Com estes mapas, podemos melhorar nosso

[2]Convolução é o processo de adição de cada elemento de uma imagem aos seus vizinhos multiplicados por um filtro (*kernel*)

algoritmo para encontrar pontos de interesse citados anteriormente: pontos com vizinhança de grande intensidade tendem a ter mais informação.

Este é um dos métodos mais usados para encontrar regiões e pontos de interesse em uma imagem. De posse destes pontos, precisamos de uma maneira de descrevê-los juntamente com sua vizinhança. Para isso, empregamos *descritores de características*. A seguir, mostramos alguns exemplos de descritores e suas vantagens.

3.2.2 *Scale Invariant Feature Transform* (SIFT)

Lowe (1999) publicou um algoritmo para encontrar e descrever, de maneira invariante, a escala e o posicionamento como características de uma imagem. Este método é chamado de *Scale Invariant Feature Transform* (SIFT) (LOWE, 1999) e engloba as tarefas de encontrar pontos de interesse de uma imagem juntamente com sua descrição. O texto aqui apresentado é baseado no artigo original.

Este algoritmo transforma uma imagem em uma coleção de vetores descrevendo características locais, sendo que estes vetores são invariantes à translação (posição relativa na imagem), escala (tamanho relativo da característica) e rotação (ângulo da característica). Também são parcialmente invariantes em relação a mudanças na iluminação da cena e mudanças de perspectiva. Assim, as características obtidas desta maneira podem ser comparadas com as de outras imagens muito diferentes e, mesmo assim, encontrar pontos semelhantes. Esta maneira de descrição de *features* é similar ao processo de reconhecimento de objetos pelo córtex visual primata, invariante a mudanças em escala, localização e iluminação (ITO, 1995).

O método SIFT tem os seguintes passos:

1. Encontrar pontos de interesse da imagem por meio de diferença de gaussianas;
2. Filtrar pontos de interesse irrelevantes;
3. Assinalar orientação de cada ponto de interesse;
4. Criar descritor baseado no histograma de gradientes da vizinhança deste ponto.

O primeiro passo é detectar pontos de interesse na imagem. Para tal, passamos filtros gaussianos em diferentes escalas da imagem (figura 3.5, onde cada coluna é uma escala, e cada fileira um degrau na escala de suavização), e computamos a diferença entre as imagens obtidas para destacar suas características (figura 3.6). Os pontos com intensidade máxima ou

mínima nesta diferença de gaussianas correspondem aos pontos de interesse com maior quantidade de informação (figura 3.7), onde cada pixel é comparado com os seus vizinhos locais, e a vizinhança das diferenças de gaussianas obtidas de escalas próximas: se ele possui o valor mínimo/máximo dentre todos, ele é um ponto de interesse e é armazenado. Esta abordagem utiliza o conceito de espaço escalar citado anteriormente na seção 3.1 para encontrar atributos invariantes a escala dentro de uma imagem, fazendo amostragens em múltiplas escalas (oitavas) (WITKIN, 1983).

Em termos matemáticos, a diferença de gaussianas é representada por $D(x,y,\sigma)$, e computada pela equação

$$D(x,y,\sigma) = L(x,y,k_i\sigma) - L\left(x,y,k_j\sigma\right), \tag{3.1}$$

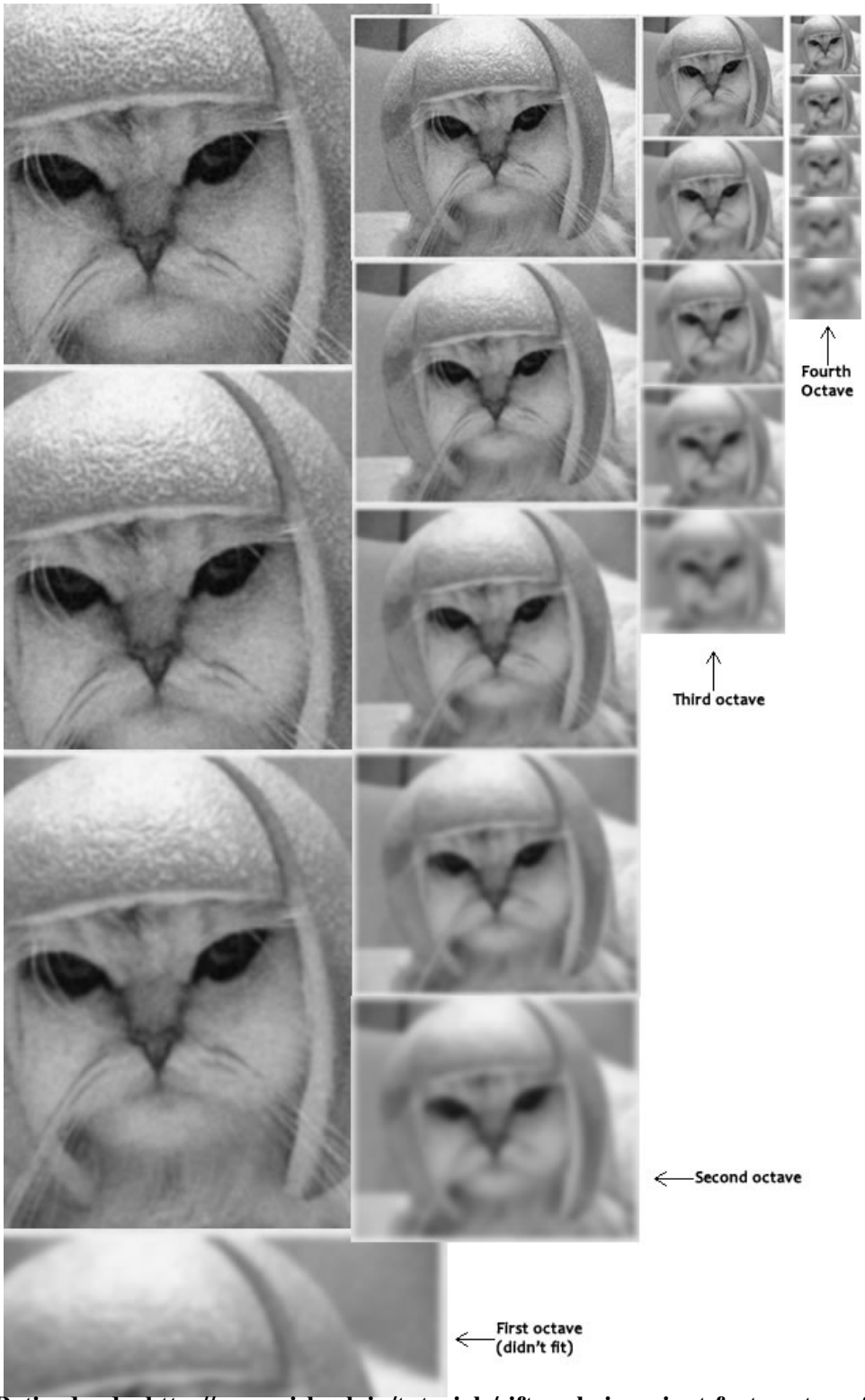
onde $L(x,y,k_i\sigma)$ é a imagem após convolução por um filtro gaussiano de escala $k_i\sigma$. O termo k marca a força da suavização aplicada pelo filtro gaussiano dentro de uma oitava, e o termo σ assinala a força da suavização inerente a cada oitava. Depois de adquiridas estas diferenças de gaussianas, procuramos pelos pontos máximos e mínimos da imagem, indo *pixel* a *pixel* e comparando seu valor com seus oito vizinhos imediatos e a vizinhança da mesma posição em diferentes escalas de suavização (figura 3.8: a matriz central corresponde à diferença de gaussianas sendo analisada, com o *pixel* central sendo o foco. Ele é comparado com seus oito vizinhos imediatos, os nove vizinhos da diferença de gaussianas obtida com menor suavização e os nove vizinhos da diferença de gaussianas obtida com maior suavização.).

Estes pontos são os máximos e mínimos em cada *pixel*, mas não são os picos de suas respectivas funções interpoladas: é necessário um cálculo posterior para encontrar estes pontos virtuais entre *pixels*. Para tal, utilizamos uma interpolação dos dados próximos aos pontos de interesse encontrados, para criar uma função contínua e encontrar a exata magnitude dos picos. Esta interpolação é feita utilizando a expansão quadrática de Taylor na função da diferença de gaussianas.

Após encontrar a posição dos mínimos e máximos, podemos filtrar os pontos de interesse por sua estabilidade em relação a sua informação relativa. Nem todos eles são estáveis: podem estar em bordas retas (com muitas correspondências globais e locais, não sendo bem definidos) ou ter baixo contraste (passam pouca informação).

Para retirar pontos de interesse em linhas, calculamos dois gradientes perpendiculares em cada um. Se os dois forem baixos, então é uma região de pouca alteração em cor/intensidade, e não é interessante. Se apenas um gradiente for baixo, é sinal de que o ponto está em uma linha

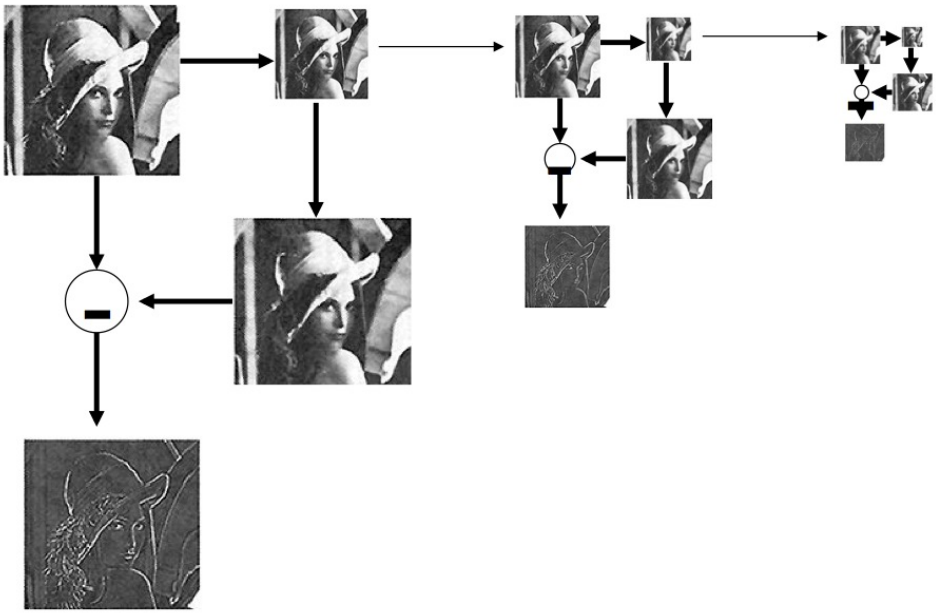
Figura 3.5: Ilustração da obtenção das imagens para o cálculo das diferenças de gaussianas.



Fonte: Retirado de http://www.aishack.in/tutorials/sift-scale-invariant-feature-transform-scale-space/ em 08/01/2017

como na figura 3.2 (b), e não é representativo. Se os dois gradientes forem altos, o ponto está em um "canto" e deve ser armazenado. Para retirar pontos de interesse com baixo contraste,

Figura 3.6: Diferenças de gaussianas obtidas em uma oitava.



Fonte: Retirado de http://eric-yuan.me/wp-content/uploads/2013/10/lap.jpg em 08/01/2017

comparamos o valor da magnitude da intensidade de cada *pixel* com um alto limiar: quanto maior, mais pontos serão filtrados.

Com pontos de interesse estáveis, assinalamos suas orientações. Para isso, usamos um histograma de gradientes com a orientação de cada *pixel* em uma janela de tamanho variável ao redor do ponto, na imagem original. A magnitude da direção ($m(x,y)$) e seu ângulo ($\theta(x,y)$) são calculados pelas seguintes equações:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}, \tag{3.2}$$

$$\theta(x,y) = \text{atan2}(L(x,y+1) - L(x,y-1), L(x+1,y) - L(x-1,y)). \tag{3.3}$$

Após calculados estes valores, passamos um filtro gaussiano pela matriz de magnitudes, com um σ igual à escala do ponto de interesse (obtida a partir de qual oitava ele pertence) vezes 1,5. Este filtro determina também o tamanho da janela de coleta de orientações. Então, criamos um histograma de orientações, dividindo o gráfico em trinta e seis parcelas, onde cada parte corresponde a um intervalo de dez graus: se um *pixel* tem o seu ângulo em 45^o, o valor de sua intensidade é adicionado à quinta parcela (correspondente ao intervalo de 40^o a 50^o). Todos os *pixels* ao redor do ponto de interesse que estão dentro deste círculo gaussiano são computados neste histograma. Na figura 3.9, podemos ver uma ilustração deste processo: a janela gaussiana

Figura 3.7: Pontos de interesse de uma imagem.

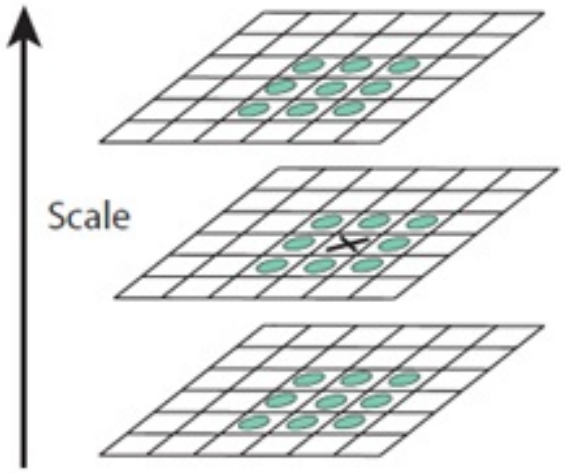
Fonte: Retirado de https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/50319/versions/10/screenshot.jpg em 08/01/2017

é representada pelo círculo azul que tem como centro o ponto de interesse, e o histograma de gradientes está representado à sua direita.

A orientação que possui mais intensidade neste histograma é escolhida para representar o ponto de interesse. Mas, se outra orientação tiver uma intensidade maior ou igual a 80% da escolhida, um novo descritor deve ser criado com ela no mesmo local. Na imagem 3.10, um histograma de orientação com múltiplos picos é representado: no caso, a orientação 20^o-29^o é a que tem mais força, mas a orientação 300^o-309^o tem mais de 80% da intensidade da orientação 20^o-29^o, o que significa que dois descritores SIFT serão criados no mesmo local.

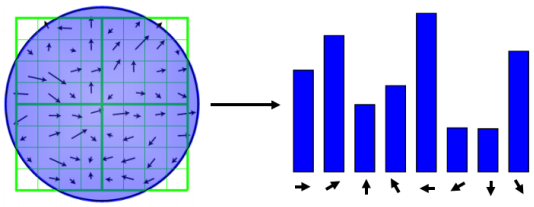
Para finalizar, criamos os vetores de descrição dos pontos de interesse. Uma janela de 16x16 *pixels* é criada ao redor do ponto, e separada em 16 divisões de 4x4 (figura 3.11). Em cada janela, todos os *pixels* tem sua magnitude da direção e seu ângulo calculados, e um histograma com 8 parcelas é criado: se o *pixel* tem ângulo entre 0^o e 44^o sua magnitude (multiplicada por

Figura 3.8: Ilustração da obtenção dos pontos máximos/mínimos.



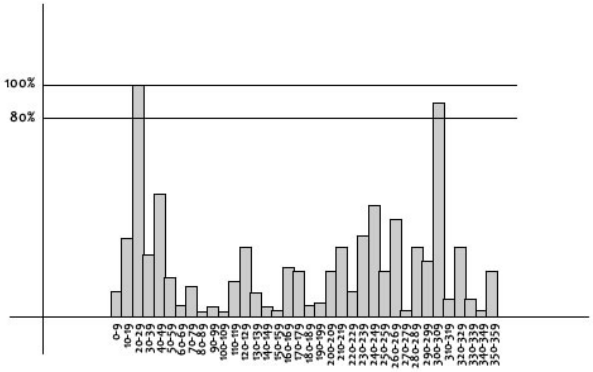
Fonte: Retirado de http://www.aishack.in/tutorials/sift-scale-invariant-feature-transform-scale-space/ em 08/01/2017

Figura 3.9: Ilustração do método de descoberta da orientação de um ponto de interesse.



Fonte: Retirado de http://www.vlfeat.org/api/sift.html/ em 08/01/2017

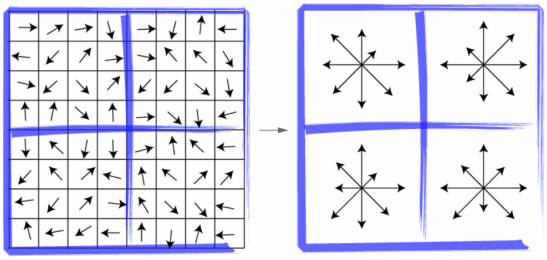
Figura 3.10: Histograma de orientação com múltiplos picos.



Fonte: Retirado de http://www.aishack.in/tutorials/sift-scale-invariant-feature-transform-keypoint-orientation/ em 08/01/2017

um fator gaussiano de distância do ponto de interesse) é adicionada à primeira parcela, se o ângulo está entre 45º e 88º adicionada à segunda, e assim por diante.

Figura 3.11: Processo de descrição de um ponto de interesse.



Fonte: Retirado de http://www.maxwell.vrac.puc-rio.br/9142/9142˙4.PDF em 08/01/2017

Para obter invariância à rotação, a orientação do ponto de interesse é subtraída de todas as orientações dos *pixels* em sua vizinhança. Para obter invariância à iluminação, todas as magnitudes que têm valor maior do que 0.2 são alteradas para 0.2, e todos os vetores são normalizados em seguida.

Os valores de magnitude não estão exatamente em cima da posição dos *pixels* na imagem: encontramos estes valores anteriormente por meio de interpolação. Todos os valores computados acima dependem desta interpolação para um cálculo exato.

Com isso, temos oito valores por divisão e 16 divisões, resultando em um vetor de 128 posições que descreve o ponto de interesse de maneira invariante à escala, rotação e iluminação, além de ser robusto com relação a mudanças em perspectiva. Este descritor é um dos mais utilizados na literatura, por sua praticidade, vasta variedade de implementações e robustez frente a diferentes domínios de imagem.

3.2.3 *Speeded Up Robust Features* (SURF)

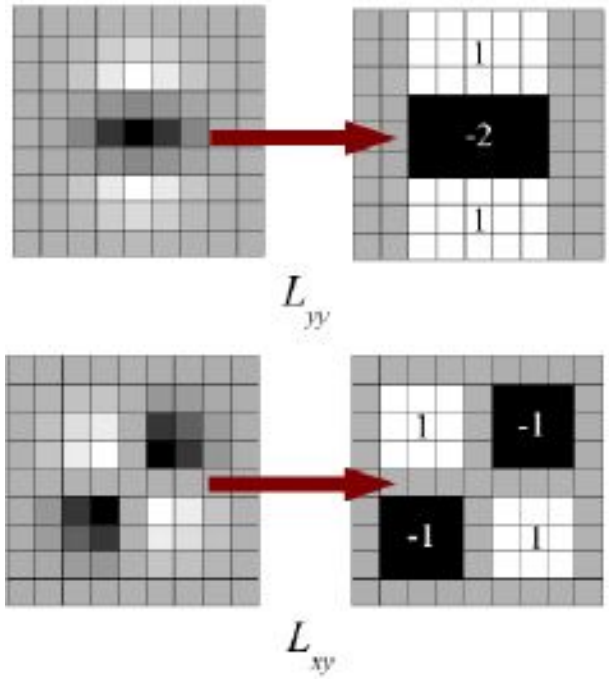
Uma alternativa ao SIFT é apresentada por Leonardis (2006) e revisada por Bay (2008): um detector e descritor de características invariante à escala e à rotação denominado SURF (*Speeded-up Robust Features*). Este método utiliza os mesmos passos do algoritmo SIFT, otimizando cálculos por diferentes maneiras de encontrar, filtrar e definir pontos de interesse. Relembrando, os quatro passos do método SIFT são detecção de pontos de interesse, filtragem, obtenção de orientação do ponto e sua descrição distribuída.

Para a detecção de pontos de interesse, o método SIFT utiliza os pontos máximos e mínimos das diferenças de gaussianas em escalas sucessivas. No método SURF, o mesmo efeito é alcançado analisando o determinante de uma matriz Hessiana, dada pela matriz $H(p,\sigma)$ computada pela equação 3.4.

$$H(p,\sigma) = \begin{pmatrix} L_{xx}(p,\sigma) & L_{xy}(p,\sigma) \\ L_{yx}(p,\sigma) & L_{yy}(p,\sigma) \end{pmatrix} \tag{3.4}$$

Os termos $L_{xx}(p,\sigma)$, $L_{xy}(p,\sigma)$, $L_{yx}(p,\sigma)$ e $L_{yy}(p,\sigma)$ correspondem às convoluções das derivadas gaussianas de segunda ordem centradas no ponto $p = (x,y)$, cada uma em uma direção diferente. Estas derivadas são obtidas no método SIFT por sucessivas suavizações na imagem, e aqui, são aproximadas por filtros quadrados com pesos inteiros baseados na teoria de imagens integrais. Os cálculos das matrizes $L(p,\sigma)$ são computacionalmente intensivos e lidam com números reais. Na Figura 3.12, temos as matrizes que gerarão os termos originais da matriz Hessiana na esquerda, e na direita, temos as aproximações feitas por filtros utilizando pesos inteiros. Os filtros na primeira linha são correspondentes ao cálculo de $L_{yy}(p,\sigma)$, e, na segunda linha, $L_{xy}(p,\sigma)$. O termo σ tem valor de 1.2, sendo a menor escala de *blur* utilizada para encontrar *blobs* em uma imagem. Para aumentar a performance deste custoso processo, utilizam-se filtros quadrados integrais (LEWIS, 1995) para obter cada uma das orientações desejadas.

Figura 3.12: Cálculo das derivadas gaussianas de segunda ordem.



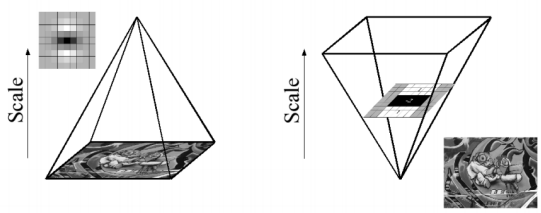
Fonte: Retirado de http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html em 09/01/2017.

Isto cobre a escala original da imagem. Mas, para encontrar características invariantes à escala, precisamos utilizar a teoria de espaço escalar e criar novas representações da imagem. No método SURF, isto é feito por uma pirâmide de imagens criada por sucessivas convoluções de filtros com escala crescente (figura 3.13). Dividimos o espaço escalar em oitavas, assim

como no algoritmo SIFT, mas ao invés de alterar diretamente a imagem, alteramos o tamanho inicial do filtro gaussiano.

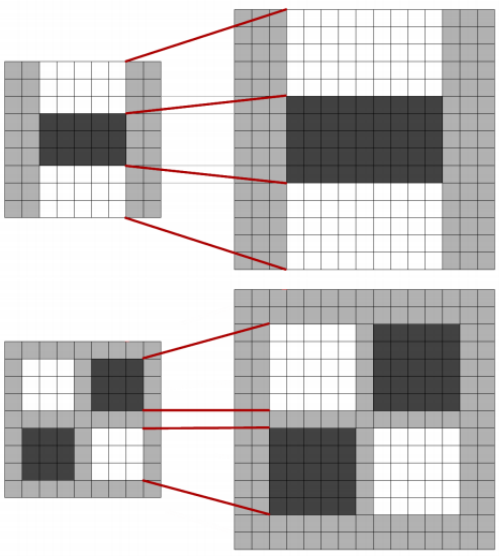
No nível inicial da pirâmide, temos os filtros de 9×9 *pixels* apresentados na figura 3.12. No segundo nível, aumentamos este filtro para 15×15 *pixels* (processo ilustrado na figura 3.14). No terceiro, 21×21 *pixels*. A cada nível, aumentamos o tamanho do filtro em 6 *pixels* em cada dimensão. Na primeira oitava, os filtros começam com 9×9 *pixels*, e as próximas oitavas tem acréscimos iguais ao dobro da oitava anterior: a segunda oitava tem o primeiro nível em 15×15 *pixels* e o segundo em 27×27, por exemplo.

Figura 3.13: Pirâmide de imagens. Na esquerda, o processo SIFT original. Na direita, o processo SURF.



Fonte: Retirado de (BAY, 2008).

Figura 3.14: Processo de aumento escalar dos filtros gaussianos aproximados. Na imagem, um filtro de 9x9 é passado para um nível superior, com tamanho 15x15.

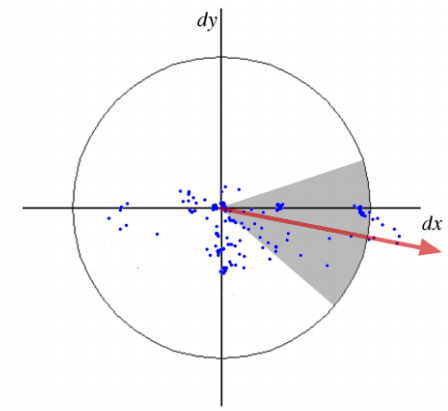


Fonte: Retirado de (BAY, 2008).

Pontos de interesse são obtidos procurando os valores máximos em vizinhanças de 3x3x3, nos seus vizinhos próximos e entre escalas. Os pontos resultantes são interpolados em espaço escalar e de imagem para encontrar suas posições exatas na imagem original.

Para encontrar a orientação deste ponto de interesse, são calculadas as respostas das transformadas de Haar (HAAR, 1910) nas direções verticais e horizontais em um raio de $6s$, onde s representa a escala do ponto. As respostas obtidas são multiplicadas por uma função gaussiana centrada no ponto de interesse e desenhadas como pontos em um espaço bidimensional. A orientação dominante do ponto de interesse é dada por um vetor com sua magnitude definida pela soma de todas as respostas em uma janela deslizante de tamanho $\frac{\pi}{3}$, e sua direção definida pelo ângulo para o qual a janela está correntemente apontando. Esta janela verifica todos os ângulos possíveis e gera um vetor local de orientação a cada passo. O maior vetor gerado dentre todos é o escolhido para ser o dominante. Na figura 3.15, a orientação dominante pode ser vista como o vetor vermelho.

Figura 3.15: Extração da orientação de um ponto de interesse pelo método SURF. O vetor indicado em vermelho é o maior dentre todos os possíveis vetores encontrados pela janela deslizante em cinza.

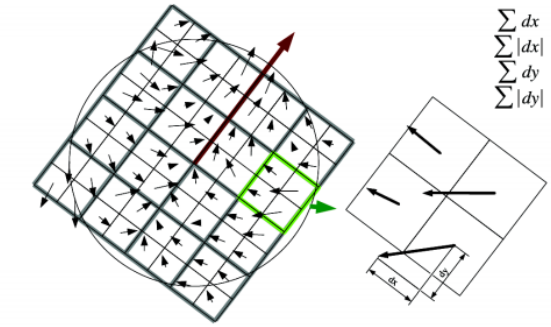


Fonte: Retirado de (BAY, 2008).

Com a orientação definida, prosseguimos para a etapa de descrição do ponto de interesse. Uma janela quadrática de tamanho $20s$ é gerada direcionada para a orientação dominante do ponto. Esta janela é dividida em quatro sub-regiões de 4x4. Em cada setor destas sub-regiões, 25 pontos de amostragem equidistantes são retirados e suas respostas à transformada de Haar são computadas e rotacionadas de acordo com a orientação dominante (figura 3.16). Elas são, então, somadas para representar a orientação e magnitude daquele setor, e cada sub-região armazena quatro números: a soma das derivadas nas direções verticais e horizontais, e a soma dos absolutos destas derivadas.

As quatro regiões tem quatro somas cada, resultando em um vetor concatenado de 64 posições. Este vetor descreve o ponto de interesse com menor tamanho comparado com o método SIFT, otimizado e com acurácia comparável.

Figura 3.16: Criação do vetor de descrição em um ponto de interesse. Na esquerda, a janela quadrática orientada completa; na direita, uma sub-região em destaque.



Fonte: Retirado de (BAY, 2008).

3.2.4 *Oriented FAST and Rotated BRIEF* (ORB)

Uma fusão expandida de dois métodos, *Features from Accelerated Segment Test* (FAST) para detecção rápida de pontos de interesse (ROSTEN; PORTER; DRUMMOND, 2010) e *Binary Robust Independent Elementary Features* (BRIEF) (CALONDER, 2010) para descrição de pontos de interesse, o algoritmo *Oriented Fast and Rotated BRIEF* (ORB) (RUBLEE, 2011) encontra e descreve imagens com vetores menores que os utilizados nos algoritmos SIFT e SURF, além de ter uma performance e custo computacional reduzidos com acurácia comparável.

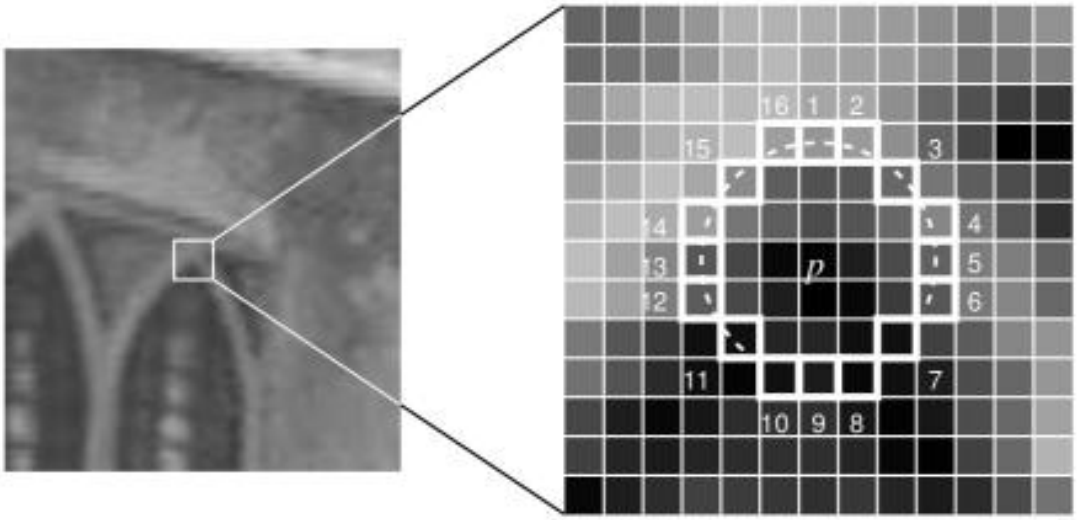
O método ORB tem quatro fases de execução:

1. Encontrar pontos de interesse em múltiplos espaços escalares (para garantir invariância escalar) utilizando o algoritmo FAST;
2. Filtrar pontos de interesse utilizando medida de *cornerness*;
3. Encontrar a orientação de cada ponto de interesse utilizando a centróide de intensidade de sua vizinhança (para garantir invariância de rotação);
4. Criar vetores binários BRIEF rotacionados de acordo com o ângulo do ponto de interesse.

O primeiro passo é utilizar o algoritmo FAST em cada ponto da imagem para encontrar pontos de interesse. Em cada ponto, definimos um círculo de raio $\rho = 9$ e comparamos os valores de intensidade dos *pixels* na borda do círculo com o *pixel* central: se *n pixels* contíguos neste círculo forem maiores ou menores do que o valor de intensidade do *pixel central* por um valor t limite pré-definido, o centro define um ponto de interesse. Na Imagem 3.17, uma janela de $\rho = 3$ é definida em volta de um ponto p. Este processo ocorre em diferentes oitavas da imagem, para encontrar pontos de interesse de diferentes tamanhos.

Os pontos encontrados são filtrados pelo seu valor de *cornerness* (HARRIS; STEPHENS, 1988), definindo sua qualidade, e sua direção é calculada utilizando a centróide de intensidades (ROSIN, 1999) da janela previamente definida pelo algoritmo FAST. Esta centróide define o ”centro de massa“ desta janela, e o ângulo definido entre a localização desta centróide e o ponto de interesse é a orientação dele.

Figura 3.17: Análise de *pixel* utilizando FAST com $\rho = 3$.



Fonte: Retirado de http://opencv-python-tutroals.readthedocs.io/en/latest/˙images/fast˙speedtest.jpg em 18/02/18.

Com o ponto de interesse e a sua orientação, utiliza-se o método BRIEF rotacionado para geração do vetor de características: m pares fixos de *pixels* ($[(x_i, y_j), ..., (x_m, y_m)]$) são definidos ao redor de um dado ponto de interesse, com localizações retiradas de uma distribuição Gaussiana isotrópica, formando uma matriz de tamanho $2m$. Esta matriz de *pixels* é rotacionada de acordo com a orientação do ponto de interesse, dando invariância ao descritor. Um vetor binário de m posições ($[v_i, v_{i+1}, ..., v_m]$) é definido a partir desta matriz, utilizando uma simples comparação entre valores de intensidade de cada par de *pixels*: dado um par (x_i, y_j), se a intensidade de x_i for maior do que a de y_j o valor de v_i será 1; caso contrário, será 0.

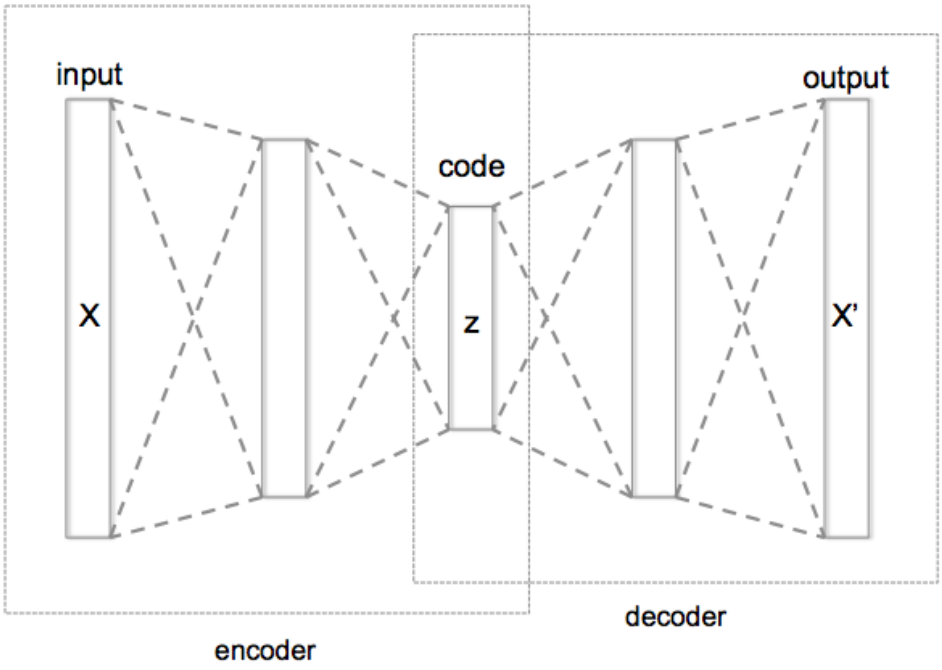
Estes vetores de descrição tem como principal vantagem o seu tempo de processamento: é muito mais rápido do que os seus principais concorrentes, SIFT e SURF, sendo comparavelmente acurado.

3.2.5 Autoencoder

Um *autoencoder* (KRAMER, 1991) é uma rede neural profunda que tem como função-objetivo aprender sua própria entrada, criando uma representação distribuída de dimensionali-

dade reduzida em suas camadas interiores (BENGIO; COURVILLE; VINCENT, 2013). O algoritmo Word2Vec apresentado na seção 2.3 tem suas raízes neste conceito: criar uma representação intermediária focando em um objetivo que precise de informações refinadas.

Figura 3.18: Ilustração da arquitetura de um *autoencoder*



Fonte: Retirado de https://en.wikipedia.org/wiki/Autoencoder em 18/01/17.

A arquitetura de um *autoencoder* consiste de duas partes principais: o setor de codificação, que recebe uma entrada e a codifica em um vetor de menor dimensionalidade, e o setor de decodificação, que recebe o código e reconstrói a entrada que o criou (arquitetura ilustrada na figura 3.18). Cada setor pode ter múltiplas camadas de neurônios, dependendo da complexidade da tarefa e da performance desejada. Denominamos o primeiro setor por ϕ e o segundo por ψ, com funções dadas por

$$\phi : \mathscr{X} \to \mathscr{F},$$

$$\psi : \mathscr{F} \to \mathscr{X},$$

$$\phi, \psi = \arg\min_{\phi,\psi} \|X - (\psi \circ \phi)X\|^2.$$

O processo de codificação e decodificação leva uma entrada de $\{x\} \in \mathbb{R}^d = \mathscr{X}$ e cria um código $\{z\} \in \mathbb{R}^p = \mathscr{F}$ que é reconstruído em $\{x'\} \in \mathbb{R}^d = \mathscr{X}$. As equações que definem tal processo em uma rede de uma única camada escondida (ou seja, entrada $\rightarrow$ código $\rightarrow$ reconstrução) são mostradas nas equações a seguir.

$$\mathbf{z} = \sigma_1(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{3.5}$$

$$\mathbf{x}' = \sigma_2(\mathbf{W}'\mathbf{z} + \mathbf{b}'). \tag{3.6}$$

Nas equações 3.5 e 3.6, os termos σ_1 e σ_2 são funções de ativação (funções sigmóides ou *rectified linear units* (ReLUs)) (BENVENUTO; PIAZZA, 1992), os termos W, b, W' e b' são os pesos e vieses das camadas de codificação e decodificação, respectivamente. O objetivo do treinamento desta rede é minimizar a diferença entre a entrada x e a saída x'. Ao terminar o treinamento desta rede neural, retiramos a camada de decodificação para trabalhar diretamente com o código, que repassa as informações mais importantes da entrada com menor dimensionalidade.

Há diversas variantes deste método: pode-se adicionar ruído a entrada para reconstruir a partir de versões corrompidas dos dados (*denoising autoencoders*) (VINCENT, 2010) ou utilizar múltiplos *autoencoders* para pré-treinamento de redes neurais (BENGIO, 2007), por exemplo. No campo de visão computacional, podemos passar imagens como a entrada de um *autoencoder* e utilizar a representação intermediária para reduzir o tempo de processamento e análise de dados.

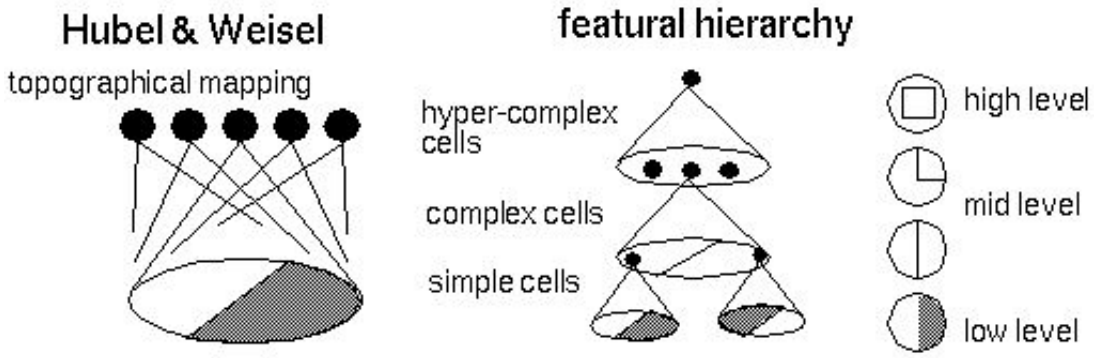
3.2.6 Síntese de Características por Redes Neurais Profundas

Um método que utiliza a abordagem neural combinada com o conhecimento da estrutura do córtex animal (HUBEL; WIESEL, 1968) é a rede neural convolucional (*convolutional neural network*, CNN). Neurônios contribuem localmente para a ativação de camadas superiores, não sendo totalmente conectados como em uma rede neural comum. As respostas de cada neurônio aos estímulos das camadas anteriores pode ser aproximado por uma operação de convolução, onde os pesos de cada camada são utilizados como filtros sobre a imagem. A função-objetivo usada pela rede neural convolucional pode ser qualquer tarefa desejada, como, por exemplo, classificação de imagens.

O córtex visual animal é formado por redes neuronais de complexidade crescente. Cada neurônio utiliza uma sub-região da imagem (campo receptivo) para computar sua resposta,

criando ativações locais e agindo como um filtro. Estas ativações são passadas para outros neurônios localmente conectados, passando a informação relevante da imagem de maneira hierárquica. Na figura 3.19, a estrutura definida por Hubel e Wiesel (1968) é ilustrada: células na primeira camada detectam características de baixo nível, passam para células de complexidade superior que compreendem toda a informação das sub-regiões adjacentes, e assim por diante.

Figura 3.19: Ilustração do córtex visual proposto por Hubel e Wiesel (1968).

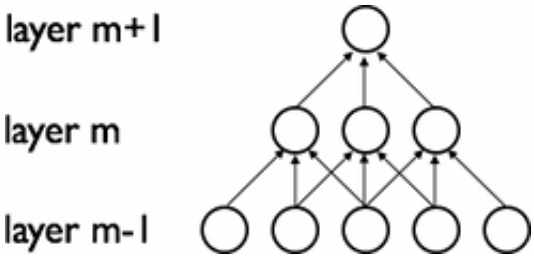


Fonte: Retirado de (HUBEL; WIESEL, 1968).

As principais características de uma rede neural convolucional são:

- Conectividade esparsa. Neurônios utilizam informações locais para sua ativação e gradualmente adicionam informações de regiões espacialmente contíguas. Na figura 3.20, os neurônios da camada m utilizam as informações dos três neurônios mais próximos da camada $m-1$. A camada $m+1$ compreende três neurônios da camada m, efetivamente utilizando as informações combinadas dos cinco neurônios da camada $m-1$.

Figura 3.20: Ilustração da arquitetura de uma rede neural convolucional. Cada neurônio se conecta apenas com os neurônios localmente próximos.

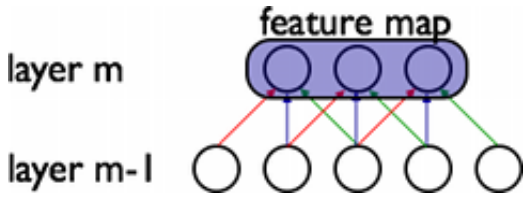


Fonte: Retirado de http://deeplearning.net/tutorial/lenet.html em 22/01/17.

- Pesos compartilhados. Os pesos de cada camada são aplicados a todos os neurônios, e não são específicos a cada ligação. Na figura 3.21, todas as conexões dos neurônios na

camada $m-1$ são multiplicadas por um único vetor de pesos para computar a ativação da camada m. Cores iguais representam fatores de peso iguais. A matriz resultante desta aplicação é chamada de mapa de características (*feature map*, no original).

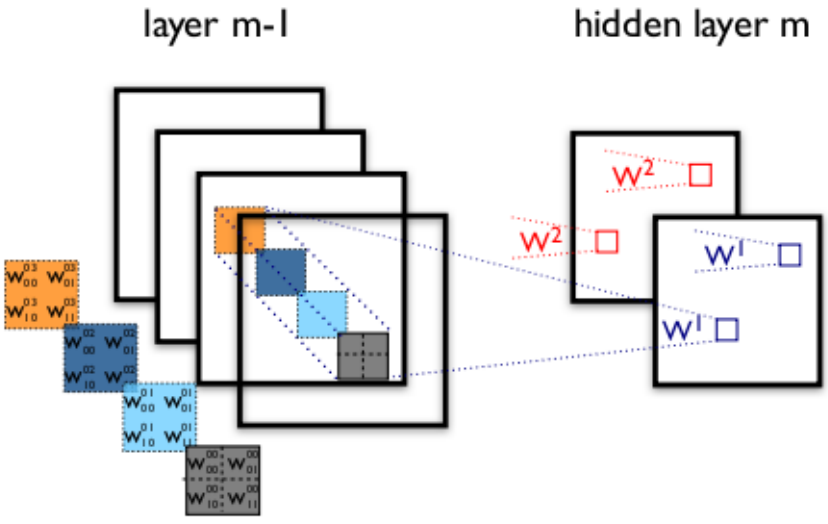
Figura 3.21: Ilustração da arquitetura de uma parte da camada convolucional de uma rede. Cada camada tem um vetor de pesos compartilhado por todos os neurônios.



Fonte: Retirado de http://deeplearning.net/tutorial/lenet.html em 22/01/17.

- Múltiplos pesos codificando características diferentes. Cada camada é composta por múltiplos filtros, que criam mapas de características a partir de seus dados de entrada. Na figura 3.22, dois mapas de características são gerados a partir de quatro mapas anteriores: cada *pixel* dentro desses mapas é resultado de uma convolução do peso correspondente à característica. Estes pesos tem três dimensões, por utilizarem informações da vizinhança de múltiplos mapas ao mesmo tempo.

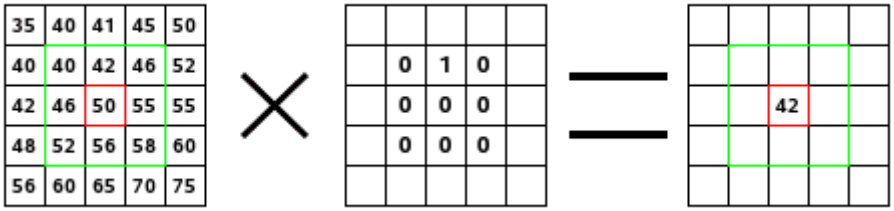
Figura 3.22: Ilustração da criação de mapas de características em uma rede neural convolucional.



Fonte: Retirado de http://deeplearning.net/tutorial/lenet.html em 22/01/17.

- A operação de convolução. Esta operação (DUMOULIN; VISIN, 2016) é o resultado da adição de todos os elementos em uma matriz multiplicados por pesos fornecidos por uma segunda matriz (figura 3.23).

Figura 3.23: Ilustração da convolução de uma matriz. A matriz original (esquerda), *kernel* (centro) e resultado (direita).

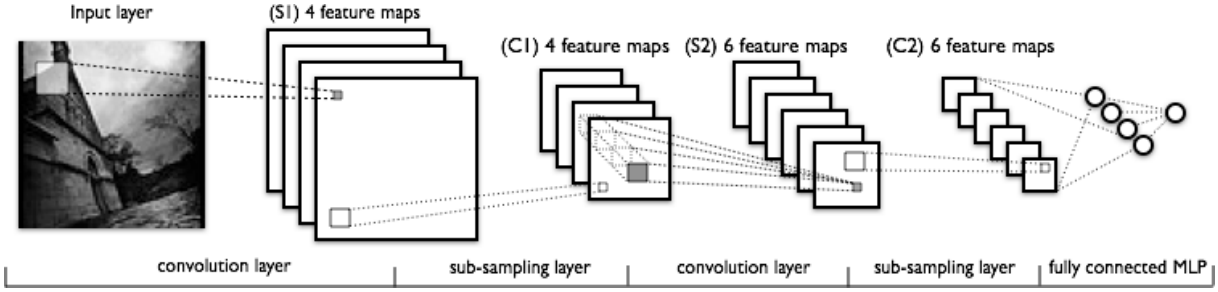


Fonte: Retirado de https://docs.gimp.org/en/plug-in-convmatrix.html em 22/01/17.

- Camadas de *pooling*, que diminuem a dimensionalidade dos dados. Setores de tamanho quadrado (2x2, 3x3, etc) são simplificados em apenas um valor, geralmente o máximo local (*max-pooling*). Assim, reduz-se o tempo de computação e aumenta-se a robustez em relação a mudanças na localização de características encontradas na imagem.

O modelo completo de uma rede neural convolucional utiliza todos estes componentes para aproximar o aprendizado dela aos métodos encontrados na biologia animal. Esse modelo foi introduzido pela primeira vez por LeCun (1998) com o nome LeNet (LECUN, 1998). A figura 3.24 ilustra o processo de extração de características até o seu uso final, em camadas totalmente conectadas para classificação. A primeira camada consiste em quatro filtros diferentes, gerando quatro mapas de características distintos a partir de uma imagem original. Estes mapas tem dimensionalidade reduzida por meio de *pooling*, e passam por nova etapa de convolução, criando seis mapas. Depois de nova redução de dimensionalidade, os mapas são concatenados em um único vetor, que é passado a uma rede totalmente conectada para classificação.

Figura 3.24: Ilustração de uma rede neural convolucional em funcionamento.



Fonte: Retirado de http://deeplearning.net/tutorial/lenet.html em 22/01/17.

Após o treinamento dessa rede, podemos retirar o classificador final (a camada totalmente conectada) para ter uma representação distribuída que encontra características interessantes

à tarefa que a rede se prestou a resolver (RADFORD; METZ; CHINTALA, 2015). Também podemos combinar as características de uma rede convolucional ao *autoencoder*, criando uma representação que utiliza as correlações locais de cada *patch* (MASCI, 2011).

Em nosso trabalho, utilizamos uma rede pré-treinada para criar representações de dados: a VGG19 (SIMONYAN; ZISSERMAN, 2015), arquitetura de rede com 19 camadas alternando convoluções e *paddings*. Esta rede foi pré-treinada com os dados da ImageNet (Jia Deng, 2009), base de dados com 1000 classes de objetos comuns. Utilizamos as probabilidades destas classes como representações das imagens, pois muitos dos objetos descritos nesta base de dados estão correlacionadas com os produtos de e-commerce de nosso domínio.

Assim, a VGG19 foi um dos métodos de representação visual investigados neste trabalho, como explicado no Capítulo 6.

Capítulo 4

REPRESENTAÇÃO DISTRIBUÍDA MULTIMODAL

O aprendizado de representações distribuídas tem diversas aplicações e algoritmos para cada modalidade de dados existente. Imagem, texto e áudio podem ser mapeados para espaços vetoriais de baixa dimensionalidade que representam bem as informações expressas em cada um deles. No entanto, considerar separadamente cada uma das modalidades pode não ser suficiente para representar informações semânticas compartilhadas pelos diversos tipos de mídias. Essa afirmação é amparada no modo como a informação é processada no mundo real, onde as mídias se complementam para formalizar um dado conceito na mente humana. Por exemplo, imagens são relacionadas a legendas e palavras-chave, notícias são complementadas com imagens, áudio e imagem formam vídeos, etc. Em nosso contexto, cada um desses tipos de dados apresenta características diferentes, dificultando a correlação de seus atributos. Assim, novas abordagens são necessárias para melhorar a representação matemática de dados provenientes de múltiplas fontes.

O aprendizado multimodal é aquele que considera informações de múltiplas fontes para realizar uma determinada tarefa. Segundo Gardner (1985), o termo vem da teoria de múltiplas inteligências, que separa o conceito de inteligência em diferentes campos: cinestésica (relativa ao corpo e seu movimento), lógico-linguística (relativa ao pensamento lógico e compreensão textual/auditiva) e espacial (relativa à percepção do ambiente e recriação), por exemplo. Computacionalmente, podemos separar as informações específicas de cada modalidade da mesma maneira, considerando as regularidades estatísticas inerentes a cada tipo de dado.

Aplicando o conceito de aprendizado multimodal aos métodos de representação distribuída atuais, assumimos que a representação conjunta de um determinado dado expressa mais informação semântica do que as representações separadas. Assim, podemos relacionar ações descritas em um texto a imagens específicas ou associar emoção e movimento de um objeto em

uma imagem a um trecho de texto, por exemplo, e assim complementar algoritmos de aprendizado de máquina.

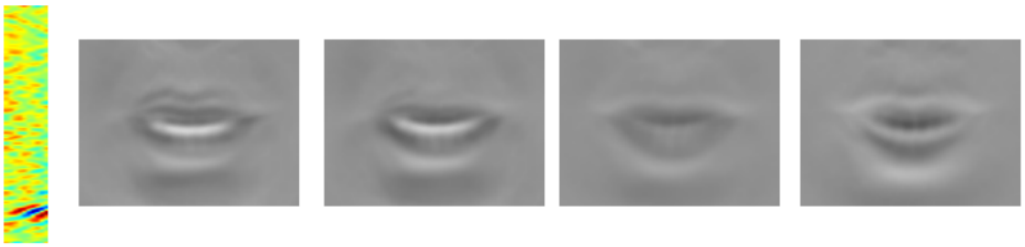
Neste capítulo, cobriremos alguns dos trabalhos que abordam o tema e desenvolvem técnicas para utilizar múltiplas fontes de dados em representações conjuntas.

4.1 *Multimodal Deep Learning*

Um dos primeiros trabalhos na área aplicando os conceitos de aprendizado multimodal a aprendizado de máquina é o de Ngiam (2011), aplicando máquinas restritas de Boltzmann (SALAKHUTDINOV; HINTON, 2009) para gerar representações conjuntas de vídeo e áudio. Esse processo é similar ao de um *autoencoder* (visto na seção 3.2.5), reconstruindo uma entrada a partir de um código intermediário de dimensionalidade reduzida.

O problema abordado por Ngiam (2011) é o de correlacionar e identificar características de fala utilizando áudio e vídeo de pessoas falando certos fonemas. Para tal, aprendem-se representações para o áudio da fala que são correlacionadas com vídeos de lábios falando tal fonema. Na Figura 4.1, representações separadas de áudio e vídeo são apresentadas, pareadas pelo modelo proposto.

Figura 4.1: Imagem das representações de áudio e vídeo para um determinado fonema.

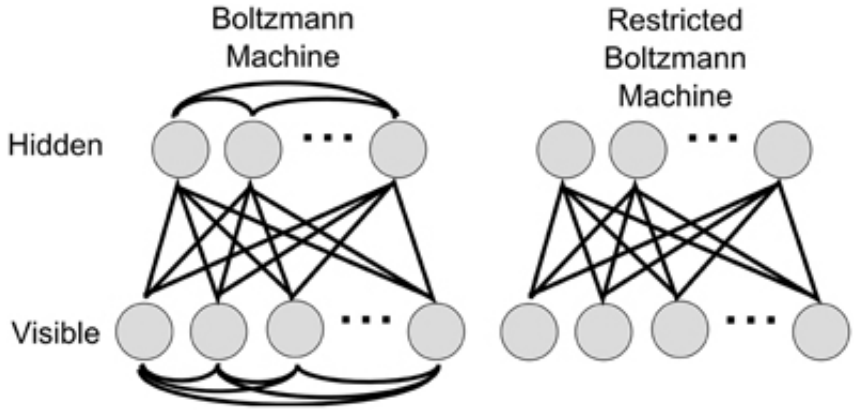


Fonte: Retirado de (NGIAM, 2011)

Uma *máquina de Boltzmann* é uma rede neural estocástica não-supervisionada que aprende uma distribuição estatística sobre um conjunto de entradas (ACKLEY; HINTON; SEJNOWSKI, 1985). Podemos considerá-la como uma máquina de estados, que altera suas ativações e seus parâmetros até estabilizar. Essas ativações são dadas por ligações entre os neurônios e são ponderadas de acordo com cada aresta para que atinjam valores altos adaptando-se a padrões estatísticos nos dados. O treinamento dessa rede consiste em sucessivas passagens de dados até a estabilização da energia residual do sistema. Sua arquitetura é composta por duas camadas: neurônios visíveis (entrada de dados) e neurônios escondidos (representação distribuída da entrada).

Para otimizar o treinamento de uma máquina de Boltzmann, podemos retirar as ligações entre neurônios da mesma camada, criando um grafo bipartido entre neurônios visíveis e escondidos. Essa arquitetura foi proposta por Salakhutdinov e Hinton (2009) com o nome de *máquinas restritas de Boltzmann*, e tem propriedades únicas que permitem o uso de diferentes algoritmos de otimização e aprendizado. As duas máquinas estão ilustradas na Figura 4.2.

Figura 4.2: Ilustração de uma máquina de Boltzmann e uma máquina restrita de Boltzmann.



Fonte: Retirado de (O'CONNOR, 2013)

O processo de treinamento de uma máquina de Boltzmann tem os seguintes passos:

1. Para cada exemplo de treinamento:

   (a) Computar a energia de ativação dada por $a_i = \sum_j w_{i,j} x_j$ para cada neurônio i da camada escondida. Essa energia de ativação é a soma de todos os valores passados pela camada visível (x_j) ponderados por um peso $W_{i,j}$.

   (b) Calcular a ativação de todos os neurônios da camada escondida utilizando uma função de probabilidade dada por $\sigma(a_j)$, onde σ é uma função de ativação previamente escolhida.

   (c) Para cada aresta $e_{i,j}$, calcular a multiplicação entre os estados dos dois neurônios conectados dada por $Pos(e_{i,j}) = x_i * x_j$.

   (d) Computar a energia de ativação a_j no sentido contrário, reconstruindo os neurônios visíveis de acordo com os neurônios escondidos e ativando-os de acordo com a função de probabilidade utilizada anteriormente.

   (e) Novamente, calcular a multiplicação entre os estados dos dois neurônios conectados, e armazenar em uma outra variável, $Neg(e_{i,j}) = x_i * x_j$.

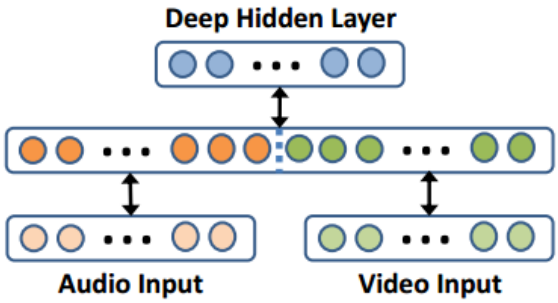
   (f) Atualizar o peso de cada aresta $e_{i,j}$ com a equação $w_{i,j} = w_{i,j} + L * (Pos(e_{i,j}) - Neg(e+i,j))$.

2. Repetir o processo até a estabilização do erro de reconstrução ou o fim das épocas pré-estabelecidas.

Esse processo também é utilizado para criação de representações, com aplicações em diferentes modos de informação, como áudio (JAITLY; HINTON, 2011), dados numéricos (SALAKHUTDINOV; MNIH; HINTON, 2007) e imagem (Van Tulder; De Bruijne, 2016).

A arquitetura do método multimodal proposto em (NGIAM, 2011) consiste em camadas sucessivas de máquinas restritas de Boltzmann. O primeiro passo é construir duas máquinas separadas, uma para cada modalidade, e utilizar os fatores latentes encontrados para gerar uma representação única, como ilustrado na Figura 4.3.

Figura 4.3: Ilustração do modelo inicial para obtenção da representação unificada.



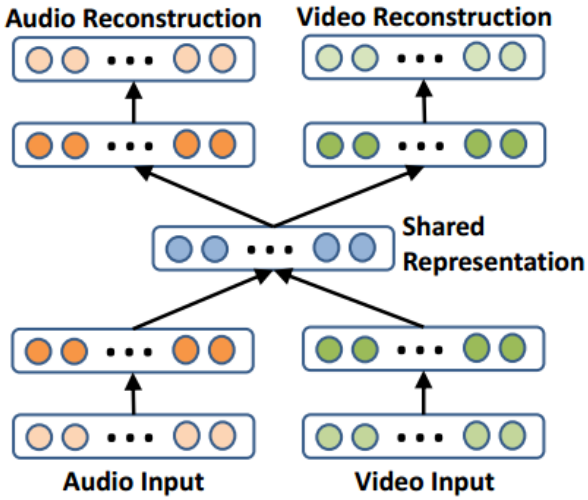
Fonte: Retirado de (NGIAM, 2011)

O segundo passo é estender essa rede com mais camadas de máquinas restritas de Boltzmann para que, além de aprender fatores latentes sobre múltiplas modalidades de dado, seja capaz de reconstruí-las posteriormente (arquitetura apresentada na Figura 4.4). Assim, as similaridades entre instâncias de dados semelhantes em modos diferentes terão influência sobre a representação final. O resultado é um *autoencoder* profundo multimodal que pode ser estendido para diversas outras áreas e modos de conhecimento.

Para o treinamento desse *autoencoder*, utilizam-se dados com faixas de vídeo e áudio, e dados com somente uma das faixas, mas que precisam reconstruir a outra da mesma maneira: assim, adiciona-se uma robustez à eventual falta de dados em uma faixa ou outra.

Em (NGIAM, 2011), foram considerados três tipos de aprendizado para testar esse modelo: Multimodal, *Cross-modality* e *Shared Representation Learning*. No Multimodal, foram utilizadas faixas de áudio e vídeo para treinamento da rede e seu teste. No caso, a acurácia da reconstrução de áudio parcialmente corrompido melhorou com a introdução da informação de vídeo na representação conjunta. No aprendizado *Cross-modality*, foram usados dados de uma modalidade para melhorar a outra, para verificar o ganho de informação com múltiplas fontes

Figura 4.4: Ilustração do modelo final para obtenção da representação unificada.



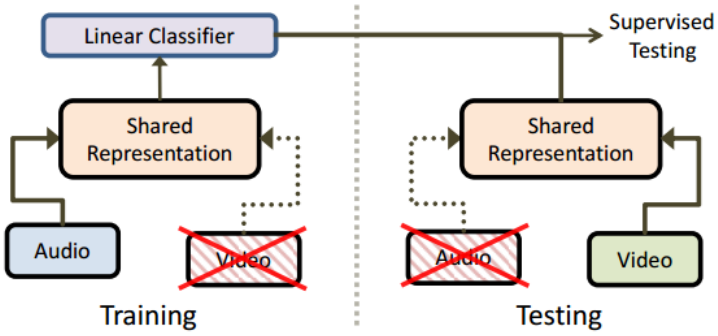
Fonte: Retirado de (NGIAM, 2011)

de dado. Para tal, utilizou-se a tarefa de classificação de vídeos no banco de dados CUAVE (PATTERSON, 2002). Como resultado, verificou-se que a introdução de informação de áudio ao calcular as representações de vídeo diminuiu os erros encontrados no processo de classificação.

Por fim, no aprendizado compartilhado de representação (*Shared Representation Learning*) testou-se uma rede treinada para criar representações de uma modalidade com dados da outra: uma rede treinada para criar representações de áudio é usada para representar vídeo e vice-versa. O objetivo, aqui, era verificar se as características obtidas são invariantes à modalidade. Nesse caso, o treinamento multimodal foi feito com uma única modalidade (somente vídeo ou somente áudio) reconstruindo áudio e vídeo e sendo testada para criar representações do tipo oposto (se treinado com áudio, cria representações de vídeo, e vice-versa). Mais especificamente, o classificador mostra o formato de um número e o modelo representa-o como áudio. Essa arquitetura é ilustrada na Figura 4.5. Na fase de treinamento, a faixa de áudio é passada para a representação compartilhada e o classificador é treinado. O teste é feito com a faixa de vídeo e sua classificação correspondente. Os resultados obtidos mostraram uma melhora considerável de acerto comparado com o *baseline* relacionado à "pura chance": dentre 10 dígitos possíveis, a chance de se relacionar uma imagem a seu áudio é de 10%, e o classificador atingiu aproximadamente 30% nesse teste. Os autores do trabalho concluem, então, que existem características invariantes à modalidade que são capturadas pelo classificador multimodal que eles propuseram.

Em suma, o modelo se mostrou capaz de capturar características compartilhadas entre modalidades de dados e melhorou a performance de classificadores utilizando tal representação.

Figura 4.5: Ilustração do modelo para treinamento de representação compartilhada.



Fonte: Retirado de (SALAKHUTDINOV; HINTON, 2009)

4.2 *Deep Canonical Correlation Analysis*

A junção neural de representações para criar vetores conjuntos não é a única maneira de se obter características compartilhadas entre modos. Um método proposto por Andrew (2013) utiliza o conceito de Análise de Correlação Canônica (*Canonical Correlation Analysis*) em conjunto com redes neurais profundas para obter vetores de representação de múltiplas modalidades de dado com máxima correlação entre si.

A análise de correlação canônica é um método que utiliza dois conjuntos de variáveis relacionadas a uma única instância de dados e cria uma projeção que maximiza a correlação entre eles (HOTELLING, 1936). Essa projeção pode ter menor dimensionalidade do que os dados originais, e representa os dois conjuntos de variáveis com um único vetor.

Dados dois vetores, $X = (x_1, \ldots, x_n)'$ e $Y = (y_1, \ldots, y_m)'$, temos uma matriz de co-covariância dada por $\Sigma_{XY} = \text{cov}(X, Y)$, onde cada posição (i, j) é a covariância de $\text{cov}(x_i, y_j)$. A análise de correlação canônica procura vetores a e b tais que $a'X$ e $b'Y$ maximizem a correlação $\rho = \text{corr}(a'X, b'Y)$. As variáveis obtidas pela multiplicação dos vetores a e b pelas variáveis iniciais são chamadas de U e V, o primeiro par de variáveis canônicas. Outros pares podem ser encontrados, desde que tenham alta correlação entre eles e baixa correlação com os demais pares já encontrados.

O método descrito em (ANDREW, 2013), denominado *Deep Canonical Correlation Analysis* (DCCA) , utiliza redes neurais profundas para calcular representações de níveis intermediários, atualizando os pesos por meio de *backpropagation* da diferença nos gradientes de correlação encontrados pelas saídas. Ou seja, a rede é treinada em função da correlação das suas saídas.

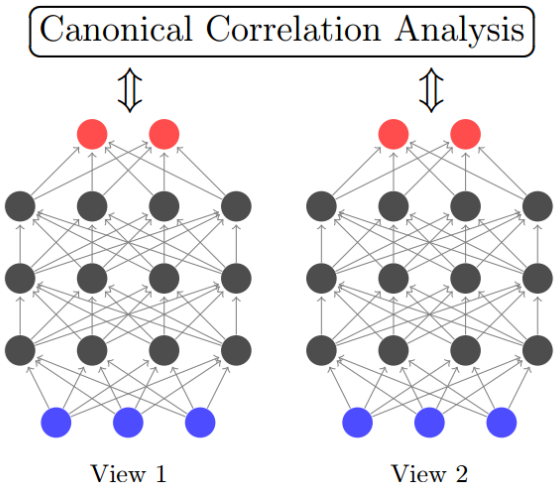
Duas redes são criadas, uma para cada entrada. Essas redes têm um resultado $f_1(X)$ e $f_2(Y)$, representando a sua saída e representação dos dados originais. Cada rede tem seus próprios

parâmetros, $\theta_1 = (W_1, b_1)$ e $\theta_2 = (W_2, b_2)$. O objetivo da rede é encontrar o par (θ_1^*, θ_2^*) que maximize a correlação entre as duas funções, representado pela equação

$$(\theta_1^*, \theta_2^*) = \underset{(\theta_1, \theta_2)}{argmax}\ corr(f_1(X; \theta_1), f_2(Y; \theta_2)). \tag{4.1}$$

Para tal, utiliza-se o gradiente da correlação calculada com base nos resultados de cada uma das redes (equação em detalhes em (ANDREW, 2013)). Assim, o objetivo da rede não é se reconstruir como em um *autoencoder*, mas sim criar uma transformação linear que maximize as semelhanças entre os dados e a variância entre as dimensões encontradas. A arquitetura desse modelo está ilustrada na Figura 4.6. Duas redes são treinadas para duas modalidades diferentes de dado, e são treinadas com o objetivo de maximizar a correlação das características representadas pelas variáveis canônicas.

Figura 4.6: Ilustração do modelo proposto por Andrew (2013).



Fonte: Retirado de (ANDREW, 2013)

Esse modelo assume que a projeção de um espaço em outro é suficiente para retratar as características compartilhadas por duas modalidades diferentes de dados. A reconstrução total de um modo pelo outro, como feito pelo método descrito na seção 4.1, pode ter mais poder de representatividade mas com um custo muito alto de processamento. O método DCCA é mais rápido, e considera também a correlação entre dimensões canônicas para que passem informações complementares entre si.

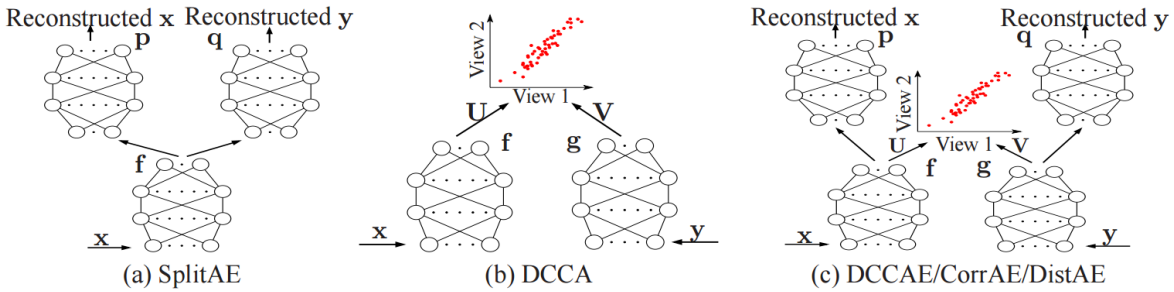
Dois experimentos foram realizados para demonstrar o funcionamento do método. O primeiro utiliza o banco de dados MNIST (LECUN; CORTES, 2010) para criar representações de imagens, onde o aprendizado da representação multimodal se dá por meio da separação das imagens em duas partes, direita e esquerda, correspondendo a dois modos diferentes de informação.

O segundo utiliza a *Wisconsin X-ray Microbeam Database* (WESTBURY, 1994), base composta por trechos de fala humana e localização espacial da língua pareados para representação multimodal de áudio e de dados numéricos. Em ambos os experimentos, a medida para determinar a representatividade do método foi o percentual de correlação total entre as dimensões das representações multimodais encontradas. Como resultado, o autor relata que esses percentuais foram superiores nas representações multimodais criadas pelo método proposto quando comparadas a outras abordagens semelhantes (*Canonical Correlation Analysis* e *Kernel Canonical Correlation Analysis*). Entretanto, isto somente prova a validade do algoritmo DCCA, e não sua eficácia em aplicações práticas.

4.3 *On deep multi-view representation learning*

As duas abordagens mostradas nas seções 4.1 e 4.2 utilizam diferentes pontos de vista para análise multimodal. O primeiro utiliza *autoencoders* para tentar reconstruir duas modalidades de dados a partir de uma; enquanto o segundo utiliza análise de correlação canônica para tentar encontrar uma projeção de máxima correlação entre dois tipos de dados. Para combinar os dois métodos, Wang (2016) propôs uma junção, fazendo com que a função-objetivo minimize o erro de reconstrução de cada uma das projeções enquanto maximiza a correlação entre elas. Na Figura 4.7, os três métodos citados são ilustrados.

Figura 4.7: Ilustração dos três métodos de representação multimodal citados: (a) *Autoencoder* multimodal, (b) *Deep Canonical Correlation Analysis* e (c) Arquitetura híbrida proposta por Wang (2016).



Fonte: Retirado de (WANG, 2016).

À esquerda, ilustra-se o método retratado na seção 4.1 sobre *autoencoders* multimodais (NGIAM, 2011), onde uma entrada de uma única modalidade x é codificada por $f(x)$ e reconstrói duas faixas simultaneamente, pelas funções $p(f(x))$ e $q(f(x))$. Ao centro, tem-se o método descrito na seção 4.2, que utiliza duas redes neurais nas entradas x e y para encontrar projeções de dados altamente correlacionadas, codificadas pelas variáveis canônicas U e V (ANDREW,

2013). À direita, é apresentado o método híbrido proposto por Wang (2016), onde o objetivo é maximizar a correlação entre projeções (U e V) e minimizar o erro das reconstruções destas projeções ($p(f(x))$ e $q(g(x))$).

Três diferentes funções-objetivo para o treinamento da arquitetura proposta são estudadas por Wang (2016): *Deep canonically correlated autoencoders*, *Correlated autoencoders* e *Minimum-distance autoencoders*.

O *Deep canonically correlated autoencoder* (DCCAE) otimiza simultaneamente a reconstrução dos dados de entrada dos *autoencoders* e a correlação canônica entre as representações de baixa dimensionalidade pela seguinte função-objetivo:

$$\begin{aligned} \min_{W_f,W_g,W_p,W_q,U,V} & -\frac{1}{N}tr(\mathbf{U}^T\mathbf{f}(\mathbf{X})\mathbf{g}(\mathbf{Y})^T\mathbf{V}) + \frac{\lambda}{N}\sum_{i=1}^{N}(||\mathbf{x_i}-\mathbf{p}(\mathbf{f}(\mathbf{x_i}))||^2 + ||\mathbf{y_i}-\mathbf{q}(\mathbf{g}(\mathbf{y_i}))||^2), \\ & \text{onde } \mathbf{U}^T(\frac{1}{N}\mathbf{f}(\mathbf{X})\mathbf{f}(\mathbf{X})^T + r_x\mathbf{I})\mathbf{U} = \mathbf{I}, \\ & \mathbf{V}^T(\frac{1}{N}\mathbf{g}(\mathbf{Y})\mathbf{g}(\mathbf{Y})^T + r_y\mathbf{I})\mathbf{V} = \mathbf{I}, \\ & \mathbf{u}_i^T\mathbf{f}(\mathbf{X})\mathbf{g}(\mathbf{y})^T\mathbf{v}_j = 0, \ \textit{para } i \neq j. \end{aligned} \tag{4.2}$$

Na equação 4.2, $\mathbf{U} = [\mathbf{u}_1, ..., \mathbf{u}_L]$ e $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_L]$ são as direções obtidas pelo método de Análise de Correlação Canônica que projetam as saídas dos primeiros *autoencoders* (**f(X)** e **g(Y)**); $r_x, r_y > 0$ são parâmetros de regularização para estimação de covariância (HARDOON; SZEDMAK; SHAWE-TAYLOR, 2004); N representa o número de amostras no banco de dados; $\mathbf{W}_f, \mathbf{W}_g, \mathbf{W}_p$ e $\mathbf{W}_q$ são os pesos dos *autoencoders* no sistema; e $\lambda > 0$ é um parâmetro para controlar quanto cada objetivo (reconstrução ou correlação) interfere no resultado. Tal objetivo é aprendido via otimização estocástica, e garante representações que levam em conta a informação da entrada original e das entradas correspondentes em diferentes modalidades.

No *Correlated autoencoders*, o termo relacionado à Análise de Correlação Canônica é trocado pela soma das correlações escalares entre pares de dimensões das representações aprendidas pelos *autoencoders*. Ou seja, a correlação entre atributos é contada sem restringir os pares de variáveis canônicas a serem não-correlacionadas entre si. Esse objetivo é dado pela equação 4.3.

$$\min_{W_f,W_g,W_p,W_q,U,V} -\frac{1}{N}tr(\mathbf{U}^T\mathbf{f}(\mathbf{X})\mathbf{g}(\mathbf{Y})^T\mathbf{V}) + \frac{\lambda}{N}\sum_{i=1}^{N}(||\mathbf{x_i}-\mathbf{p}(\mathbf{f}(\mathbf{x_i}))||^2 + ||\mathbf{y_i}-\mathbf{q}(\mathbf{g}(\mathbf{y_i}))||^2),$$
$$\text{onde} \tag{4.3}$$
$$\mathbf{U}^T(\frac{1}{N}\mathbf{f}(\mathbf{X})\mathbf{f}(\mathbf{X})^T + r_x\mathbf{I})\mathbf{U} = \mathbf{V}^T(\frac{1}{N}\mathbf{g}(\mathbf{Y})\mathbf{g}(\mathbf{Y})^T + r_y\mathbf{I})\mathbf{V} = N,$$
$$\textit{para } 1 \leq i \leq L.$$

No *Minimum-distance autoencoders*, o objetivo é maximizar a correlação entre as representações por meio da aproximação à função de minimizar a distância entre as projeções das duas modalidades. Assim, a função-objetivo tem menos restrições e sua computação é otimizada pela equação 4.4.

$$\min_{W_f,W_g,W_p,W_q,U,V} -\frac{1}{N}\frac{\|f(x_i)-g(y_i)\|^2}{\|f(x_i)\|^2+\|g(y_i)\|^2} + \frac{\lambda}{N}\sum_{i=1}^{N}(||\mathbf{x_i}-\mathbf{p}(\mathbf{f}(\mathbf{x_i}))||^2 + ||\mathbf{y_i}-\mathbf{q}(\mathbf{g}(\mathbf{y_i}))||^2). \tag{4.4}$$

Essa equação representa uma combinação dos erros de reconstrução dos dois *autoencoders* e a discrepância média entre os pares de variáveis da representação. Não há restrições e a otimização por descida de gradiente pode ser utilizada normalmente (RUDER, 2016).[1]

Testes com as representações obtidas considerando as diferentes funções-objetivo estudadas por (WANG, 2016) e em trabalhos relacionados apontam que a maior quantidade de informação absorvida pelas funções de correlação/reconstrução afeta positivamente a representatividade de dados multimodais. Diferentes modalidades foram usadas em diferentes tarefas, como a classificação de imagens originais e corrompidas (banco de dados MNIST) (LECUN; CORTES, 2010), classificação de vídeo e áudio de fonemas falados por humanos (MUNHALL; VATIKIOTIS-BATESON; TOHKURA, 1995) e pareamentos substantivo-adjetivo e verbo-objeto em representação textual multilíngue (MITCHELL; LAPATA, 2010). A abordagem de encontrar características correlacionadas entre si funciona perfeitamente em situações onde duas modalidades de dados não são correlacionadas mas possuem classes definidas e podem passar por treinamento supervisionado, como na base de dados MNIST original/corrompida. Nas outras tarefas, não há vantagem significativa em adicionar mais complexidade aos modelos anteriores (*autoencoders* multimodais e DCCA).

[1] Método numérico iterativo para otimização de funções, utilizando a direção negativa do gradiente para encontrar um valor mínimo de resposta.

4.4 *Bidirectional Joint Representation Learning with Symmetrical Deep Neural Networks*

As abordagens anteriores utilizam a fusão de camadas de diferentes *autoencoders* para reconstruir dados de múltiplas modalidades a partir de representações de menor dimensionalidade. Essas redes têm uma alta complexidade, tanto para sua criação quanto para seu treinamento. Uma alternativa foi proposta por Vukotić, Raymond e Gravier (2016), onde dois *autoencoders* são treinados simultaneamente com pesos iguais, reconstruindo dados em modalidades diferentes e criando representações de dados utilizando informação multimodal.

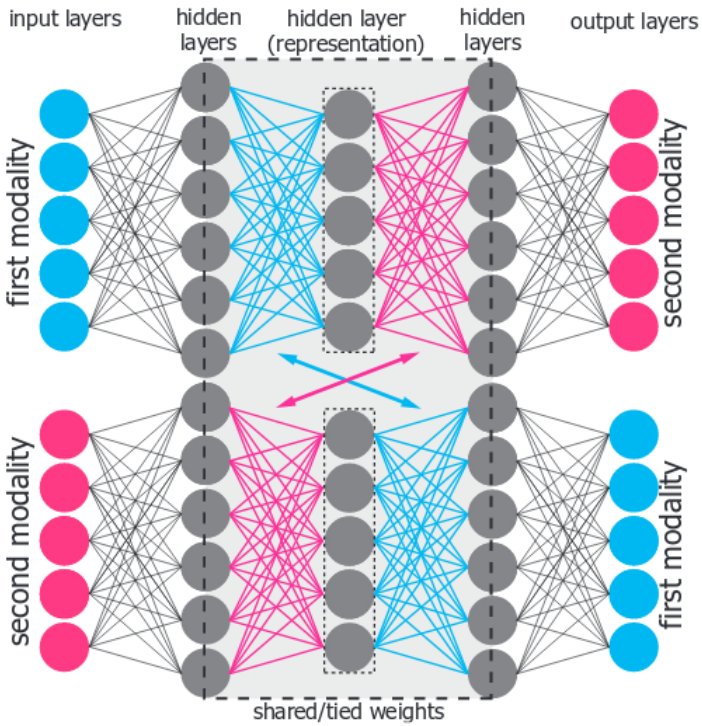
Esse modelo, diferentemente dos outros já apresentados anteriormente, não funde diretamente as informações de duas fontes diferentes. Representações multimodais são geradas aqui utilizando o produto do treinamento de duas redes diferentes, criando indiretamente uma conexão entre modalidades. Assim, essas redes podem ser utilizadas de forma independente, com menos processamento e melhor distinção entre representações.

Esse modelo de redes neurais ligadas (abreviada neste trabalho por BiDNN) mapeia um modo diretamente em outro. Sua arquitetura, ilustrada na Figura 4.8, é semelhante à de um *autoencoder* normal, onde cada rede representa a passagem de um modo a outro. A diferença aqui está no treinamento: os pesos internos das redes são iguais, e são treinados ao mesmo tempo para as duas modalidades. Assim, o espaço vetorial definido pela camada no meio das duas redes é compartilhado entre as duas redes, e codifica uma representação multimodal. Podemos passar uma modalidade para a outra utilizando a arquitetura completa da rede, ou podemos codificar representações conjuntas utilizando os resultados das camadas escondidas centrais, concatenando-as no caso onde as duas modalidades de um dado estejam disponíveis.

Em Vukotić, Raymond e Gravier (2016), a abordagem descrita na Seção 4.1 é comparada por meio da tarefa de criação de Hyperlinks em vídeos utilizando a base de dados MediaEval 2014 (ESKEVICH, 2014). Neste teste, o *autoencoder* multimodal teve performance muito inferior ao modelo bidirecional (59.60% obtido pelo *autoencoder* multimodal contra 73.74% obtido pelo *autoencoder* bidirecional, medida representando precisão utilizando as top-10 predições).

Por sua simplicidade em não envolver objetivos diferentes, seu resultado superior às demais arquiteturas apresentadas anteriormente (como apresentado no próprio artigo que derivou essa arquitetura, na tarefa de criação de *hyperlinks* (VUKOTIĆ; RAYMOND; GRAVIER, 2016)) e disponibilidade do código original, essa arquitetura foi a escolhida para ser a geradora de representações multimodais neste trabalho.

Figura 4.8: Ilustração original da rede proposta por Vukotić, Raymond e Gravier (2016).



Fonte: Retirado de (VUKOTIĆ; RAYMOND; GRAVIER, 2016).

Capítulo 5

CLASSIFICAÇÃO MULTIMODAL

Os métodos de representação distribuída de modalidade única têm apresentado excelentes resultados individualmente, em diversas aplicações e com diferentes objetivos, como pôde ser verificado nas descrições de alguns desses métodos apresentadas nos Capítulos 2 e 3. A representação multimodal está se popularizando como tema de trabalhos científicos e conferências como a *International Conference on Multimodal Interaction* (ICMI..., 2016), por exemplo. Os trabalhos explorados no Capítulo 4 indicam que há um ganho de informação com a utilização de múltiplas fontes de informação. Nesses trabalhos, utiliza-se o dado bruto (sem pré-processamento) para capturar o máximo de informações possível de cada instância.

Neste trabalho, queremos explorar as representações individuais de informação e verificar se a junção de dados previamente mapeados por dois métodos de representação diferentes (como, por exemplo, SIFT, SURF e Word2Vec) pode ter efeitos benéficos em aplicações práticas como classificação e comparação de informações. Sabemos que existe correlação entre diferentes tipos de dados brutos, e que as informações codificadas por dois tipos de dados são complementares, então, a questão de pesquisa que naturalmente emerge e que é explorada neste trabalho é: os métodos de representação distribuída de texto e de imagem já existentes são capazes de codificar e agregar tais informações ao utilizá-las em conjunto? Em consequência dessa questão surge outra: a qualidade de uma representação multimodal depende diretamente das representações unimodais que a compõe?

Outra característica importante desta pesquisa é que, embora os modelos propostos sejam independentes de língua, seu objeto de estudo é a língua portuguesa. A maioria dos trabalhos na literatura utiliza textos em inglês para comparações e testes. Compilando uma base de dados composta de textos em português e suas imagens correspondentes, esperamos fomentar aplicações brasileiras dos algoritmos e modelos propostos neste trabalho e, assim, avançar o

estado-da-arte em pesquisas com o idioma português do Brasil. A base de dados compilada neste trabalho é apresentada na seção 6.1.

Para criar as representações multimodais, neste trabalho foram explorados dois modelos: (i) um *autoencoder* multimodal simplificado (MMAE, *autoencoder multimodal*) e (ii) as Redes Neurais Bidirecionais com Pesos Ligados (*Bidirectional Deep Neural Networks with tied weights*, BiDNN) propostas em (VUKOTIĆ; RAYMOND; GRAVIER, 2016).

A seguir, nas seções 5.1 e 5.2 são apresentados os dois modelos implementados neste trabalho. Os experimentos para avaliá-los são descritos no Capítulo 6.

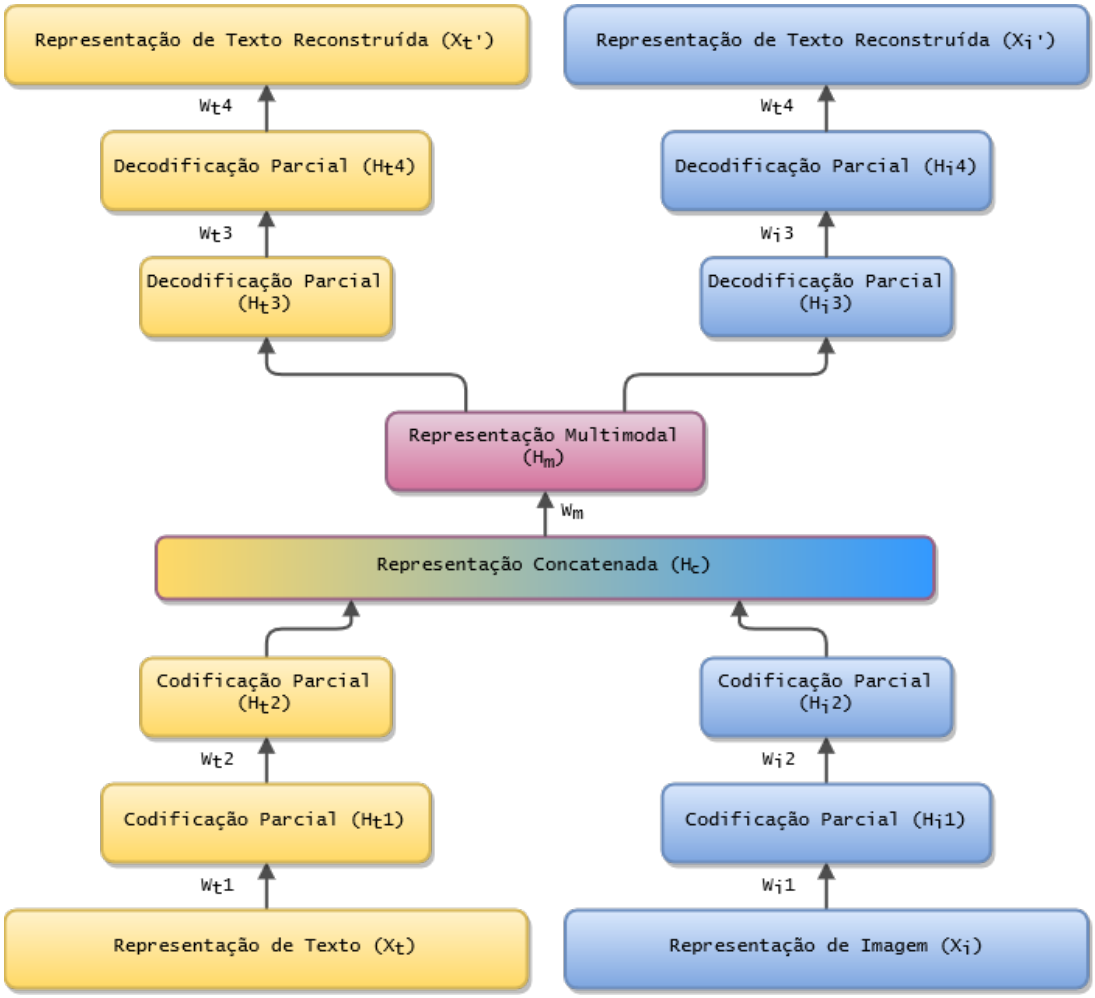
5.1 Modelo MMAE

O primeiro modelo implementado neste trabalho serve como *baseline* para a comparação direta entre representações multimodais geradas a partir dele. Mais especificamente, ao mudarmos as representações unimodais que ele utiliza para formar o espaço vetorial, esperamos que seus resultados mudem também.

A arquitetura do *autoencoder* simplificado, ilustrada na Figura 5.1, utiliza dois modos de informação com seus respectivos métodos para a obtenção de representações distribuídas (os quais podem ser, por exemplo, o Word2Vec para texto e o SIFT para imagem). Após mapear os dados para suas representações, essas são concatenadas e o vetor resultante é fornecido como entrada para um *autoencoder*. Essa rede neural, após treinada para reconhecer sua própria entrada, criará uma codificação em sua camada escondida de menor dimensionalidade, comprimindo os dois vetores para obter a representação multimodal. Se informações complementares existirem entre os mapeamentos individuais, este vetor terá mais representatividade do que os seus componentes originais.

Na Figura 5.1, duas representações são fornecidas como entrada para a rede (X_t e X_i). As camadas escondidas, situadas antes da camada de codificação da representação multimodal, chamadas aqui de "Codificação Parcial" (H_t1, H_t2, H_i1, H_i2), utilizam somente a informação das representações unimodais. O resultado das duas camadas ocultas unimodais é concatenado em uma representação única (H_c), que é utilizada para criar a representação multimodal (H_m). A partir dessa representação multimodal, a rede tentará reconstruir as representações que lhe formaram inicialmente por meio de outras camadas escondidas, chamadas aqui de "Decodificação Parcial" (H_t3, H_t4, H_i3, H_i4) gerando como saída vetores que devem se assemelhar às representações de entrada (X_t' e X_i').

Figura 5.1: Arquitetura do *autoencoder* multimodal simplificado.



Fonte: Próprio autor.

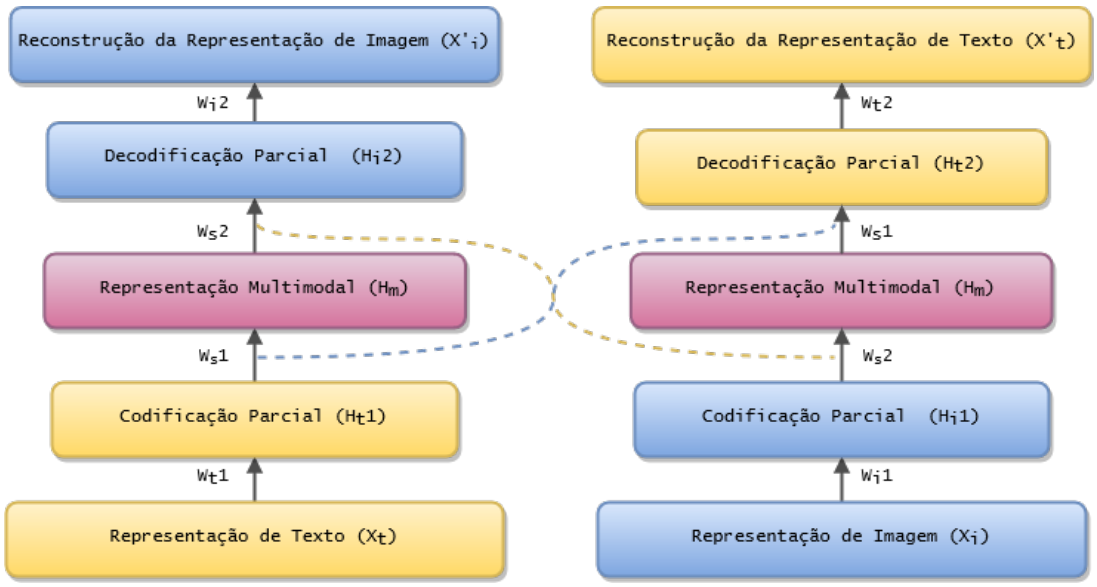
5.2 Modelo BiDNN

O segundo modelo implementado neste trabalho replica a estrutura de teste apresentada por Vukotić, Raymond e Gravier (2016), e também serve para medir a performance de uma representação multimodal bem fundamentada e robusta comparada com os meios unimodais já conhecidos.

A arquitetura da rede neural bidirecional, descrita na seção 4.4, ilustrada na Figura 5.2, utiliza dois modos de informação com seus respectivos métodos para a obtenção de representações distribuídas (os quais podem ser, por exemplo, o Word2Vec para texto e o SIFT para imagem). Após mapear os dados para suas representações, duas redes neurais são treinadas em conjunto para mapear uma representação na outra, de maneira que os pesos internos são compartilhados para criar um espaço vetorial único na sua camada mais interna (na Figura 5.2, tais pesos estão conectados por linhas tracejadas).

Na Figura 5.2, duas representações são dadas para as redes (X_t e X_i). As camadas escondidas, situadas antes da camada de codificação da representação multimodal, chamadas aqui de "Codificação Parcial" (H_t1 e H_i1), utilizam somente a informação das representações unimodais. Para criar uma representação multimodal, cada rede utiliza pesos compartilhados (W_s1 e W_s2) mapeando os dados em um espaço vetorial único. Esta representação multimodal tentará reconstruir a representação oposta (X_t reconstrói X_i' e vice-versa).

Figura 5.2: Arquitetura da BiDNN.



Fonte: Próprio autor.

A maioria dos métodos já propostos anteriormente utiliza representações fixas, sem comparar o impacto que diferentes abordagens de representação parcial podem ter na representação final. O diferencial deste trabalho é a comparação de múltiplas representações unimodais diferentes na formação das representações multimodais.

Capítulo 6

EXPERIMENTOS

Os experimentos descritos neste capítulo foram realizados com o objetivo de responder as duas questões de pesquisa (hipóteses) definidas neste trabalho: (1) se os mapeamentos de métodos de representação distribuída de texto e de imagem já existentes são capazes de codificar a correlação ou complementação das informações presentes quando utilizados em conjunto e (2) se as combinações diferentes de representações unimodais alteram a qualidade das representações multimodais.

A aplicação selecionada para avaliar extrinsecamente a qualidade das representações multimodais em comparação com as unimodais foi a de classificação automática aplicada em dois domínios: (1) produtos de *e-commerce* e (2) notícias de jornal.

Duas bases de dados foram, então, construídas para esses domínios: (1) uma contendo produtos (descrição textual do produto e imagem associada) do site de *e-commerce* Balão da Informática[1] e (2) outra contendo notícias (texto da notícia e imagem associada) da Folha de São Paulo Internacional[2], conforme descrito na seção 6.1. Para a geração das diferentes representações unimodais (de texto e de imagem) foram utilizadas as ferramentas descritas na seção 6.2.

Os experimentos, então, foram realizados considerando-se a similaridade das *embeddings* obtidas pelos diferentes métodos de representação unimodal e multimodal, além da utilização desses vetores para classificação. Como medida final de avaliação, utilizou-se a acurácia dos classificadores baseados em suas entradas; ou seja, o foco não foi, em nenhum momento, otimizar cada classificador e seus hiper-parâmetros para cada caso. Portanto, valores padrão foram utilizados para a maioria dos parâmetros, em cada modelo de classificação.

[1] http://www.balaodainformatica.com.br
[2] http://www1.folha.uol.com.br/internacional/en/

6.1 Bases de dados

Para a realização dos experimentos descritos neste documento, utilizamos duas bases de dados: (1) coleção de descrições de produtos e imagens relacionadas, retiradas do site de comércio eletrônico (*e-commerce*) Balão da Informática, e (2) coleção de notícias em português e inglês com imagens relacionadas, retiradas da Folha de São Paulo.

Nas descrições de produtos em sites de *e-commerce*, cada produto é acompanhado de uma descrição textual e uma ou mais imagens que ilustram aquele produto, para que o consumidor saiba o que está comprando. A seguir, apresenta-se um exemplo de uma descrição de um produto (Figura 6.1) e uma imagem associada (Figura 6.2). De modo similar, em uma notícia, imagens sobre o evento ou sobre uma pessoa da qual trata a notícia são apresentadas para ilustrar o artigo (exemplo na Figura 6.3), facilitando o entendimento do leitor relacionando informação visual com textual (Beatriz Ribas, 2004). Algumas notícias não possuem imagem associada, mas sempre possuem os dois textos em inglês e português.

Figura 6.1: Descrição de um Mouse vendido em um *e-commerce*.

Caracteristicas:
- Marca: Logitech
- Modelo: G600

Especificações:
- 8200 DPI
- Cor: Preto
- Optico

Informações adicionais:

Torne-o pessoal:
- Deseja personalizar as configurações padrão? Deseja acessar capacidades de macro e script mais complexas do que as que as macros incorporadas oferecem? Faça o download do Logitech Gaming Software opcional e chegue a configurações perfeitas

Memória incorporada:
- Os três perfis de memória estão armazenados para você poder ter acesso a todas informações de botões, rastreamento e cores de luz em qualquer computador sem a necessidade de software

Duplique com G-Shift:
- A atribuição da função G-Shift a um botão no G600 dobra a funcionalidade dos outros botões no mouse, proporcionando grande poder e flexibilidade de configuração

Alterne DPI imediatamente:
- Para obter precisão de alvo ou disparo, atribua um botão de alternância de DPI para mudar rapidamente o valor de DPI. Ao soltá-lo, o valor padrão será restaurado. Defina até cinco níveis de DPI por perfil

Posto à prova, construído para durar:
- Os três botões grandes principais são regulados para 20 milhões de cliques cada um. O teste de durabilidade opera teclas três vezes por segundo, o dia inteiro, sete dias por semana, durante quase três meses - e por isso os cliques nunca irão falha

Cabo USB trançado:
- O flexível cabo USB de 1,90 m é revestido com tecido para proporcionar durabilidade adicional e uma aparência de alta qualidade. E é entregue com uma correia com colchete para ajudar a manter tudo organizado és de baixa fricção
- Pés de ultrabaixa fricção reduzem significativamente a quantidade de esforço necessária para mover o mouse, resultando em menor cansaço em longas sessões de jogo

Pés de baixa fricção:
- Pés de ultrabaixa fricção reduzem significativamente a quantidade de esforço necessária para mover o mouse, resultando em menor cansaço em longas sessões de jogo

Fonte: Retirado de http://www.kabum.com.br/produto/36686/mouse-gamer-mmo-logitech-optico-g600-8200-dpi-pret- em 06/03/2017

Figura 6.2: Fotografia de um Mouse vendido em um *e-commerce*.

Fonte: Retirado de http://www.kabum.com.br/produto/36686/mouse-gamer-mmo-logitech-optico-g600-8200-dpi-pret- em 06/03/2017

Figura 6.3: Trecho de notícia em jornal eletrônico.

O ministro defendeu também que o fim das coligações seja encaminhado e passe a valer já nas eleições de 2018, considerando que a emenda constitucional sobre o tema foi aprovada no Senado e precisa passar pela Câmara. Para ele, a mudança já um "grande ganho".

Gilmar acrescentou que espera em breve uma decisão sobre o fim do sigilo das delações da Odebrecht, dizendo que "na semana que vem ou daqui a pouco" o relator do processo na Suprema Corte, Edson Fachin, pode deliberar sobre a questão.

Foto: André Dusek|Estadão

Gilmar Mendes, ministro do STF

Fonte: Retirado de http://politica.estadao.com.br/noticias/geral,gilmar-mendes-sugere-crowdfunding-para-financiar-campanha-eleitoral,70001688565 em 06/03/2017

A Tabela 6.1 resume os tamanhos das bases de dados utilizadas neste trabalho em termos de quantidade de conjuntos de texto-imagem.

Tabela 6.1: Quantidade de conjuntos de textos e imagens das bases de dados construídas neste trabalho.

Base de dados	Quantidade de conjuntos textos-imagens
Produtos	6.390
Notícias	935

Como mencionado anteriormente, esse é um dos primeiros trabalhos que realizam a coleta e criação de uma base multimodal para o idioma português do Brasil. As bases de dados aqui descrias, assim como todo o código produzido neste trabalho, serão disponibilizados no site do LALIC: www.lalic.dc.ufscar.br.

Para a tarefa de classificação automática, foram consideradas as classes pré-definidas em cada uma dessas bases de dados, as quais foram coletadas automaticamente por meio de *crawlers*. As descrições detalhadas das classes que compõem cada uma dessas bases de dados são apresentadas nas subseções a seguir.

6.1.1 Base de dados de Produtos

A base de dados do Balão da Informática consiste de 6.390 produtos eletrônicos com descrições, imagens, classes e sub-classes. A coleção de descrições e nomes dos produtos tem 13.348 palavras únicas (*types*), com 10 classes e 78 sub-classes. Cada produto tem uma descrição textual do produto acompanhada, opcionalmente, de uma ou mais imagens. Neste trabalho foram utilizadas as 9 seguintes classes (descartando uma classe "papelaria", pois a mesma era composta por apenas 9 produtos):

1. Automotivo (244 produtos) – com acessórios automotivos;
2. Casa e Eletrônicos (607 produtos) – com eletrodomésticos e acessórios elétricos para cozinha;
3. Games (122 produtos) – com *consoles* e jogos;
4. Hardware (1.144 produtos)– com peças de computador;
5. Informática (901 produtos) – com produtos relacionados a computadores (roteadores e *no-breaks*, por exemplo);
6. Periféricos (1.891 produtos)– com produtos utilizados em conjunto com computadores (*mouse* e teclado, por exemplo);
7. Usados (703) – com um amálgama de todas as outras categorias, com descrições feitas por usuários do site;
8. *Smartphones* (708 produtos) – com celulares e acessórios;
9. Telefonia (60 produtos) – com telefones fixos, cabos e rádios.

Um exemplo de produto é apresentado na Figura 6.4: um cartucho de tinta para impressoras HP®, item da classe "periféricos" e sub-classe "suprimentos para impressão".

Figura 6.4: Produto da base de dados Balão da Informática

Imagem	**Texto** (Cartucho HP 82 (CH566A) (Ref.46891))
	Descrição do Produto: HP 82 de 28 ml foi concebido com a impressora e suportes HP de grandes formatos para produzir uma qualidade de linha nítida e precisa e impressão sem problemas. Imprima internamente com confiança usando tinteiros originais HP. Imprima internamente com confiança. As tintas originais HP foram concebidas com a impressora HP Designjet e suportes HP de grandes formatos para produzir imagens vibrantes e de alto impacto, uma qualidade de linha nítida e precisa e resultados profissionais em todas as impressões. Com a impressão fiável e sem problemas dos consumíveis Originais HP evita as experiências e os erros dispendiosos. A inteligência incorporada nos tinteiros originais HP e a impressora optimizam de forma contínua a qualidade e fiabilidade para oferecer resultados profissionais em todas as impressões. Compatibilidade de hardware: Impressora HP Designjet 500/500 Plus/500PS, HP Designjet 510 Series

Fonte: Retirado de http://www.balaodainformatica.com.br/cartucho-hp-82-ch566a-ref46891/p em 24/03/2018.

6.1.2 Base de dados de Notícias

A base de dados da Folha de São Paulo Internacional consiste de 935 notícias com texto em português e inglês, imagens e classes. A coleção de descrições e nomes das notícias tem 21.704 palavras únicas (*types*), sendo que 8.721 advém dos textos em inglês e 12.984 dos textos em português, dividida em 10 classes. Toda notícia nesta base tem um texto em português e um texto em inglês acompanhados, opcionalmente, de uma ou mais imagens. Neste trabalho, foram utilizadas as 10 seguintes classes:

1. *Brazil* (338 artigos) – com notícias gerais sobre o dia-a-dia no Brasil;
2. *Business* (124 artigos) – com notícias sobre negócios nacionais e internacionais;
3. *Culture* (59 artigos) – com notícias sobre cultura brasileira, arte e música;
4. *Ombudsman* (66 artigos) – com artigos de opinião produzidos com base nas reclamações e pedidos dos leitores;
5. *Opinion* (42 artigos) – com artigos de opinião produzidos por outros jornalistas;

6. *São Paulo* (75 artigos) – com artigos sobre a cidade de São Paulo;
7. *Science and Health* (32 artigos) – com artigos sobre avanços da ciência;
8. *Sports* (77 artigos) – com artigos sobre esportes;
9. *Travel* (57 artigos) – com artigos sobre turismo;
10. *World* (65 artigos) – com artigos sobre notícias internacionais.

Este corpus tem uma distribuição de classes desigual: a maior parte das notícias está em *Brazil* ou *Business*, com cada classe tendo o dobro de instâncias comparadas as outras. Um exemplo de notícia está descrito na Tabela 6.2, artigo sobre o uso de *doping* na categoria *Sports*. A tradução da notícia é quase literal, podendo ser pareada sentença por sentença.

Tabela 6.2: Trecho de uma notícia da base de dados Folha Internacional sem imagem associada

Texto em Inglês	Texto em Português
This week, ABCD (Brazilian Doping Control Agency), the federal entity for combating doping attached to the Ministry of Sports, will undergo an audit from WADA (World Anti-Doping Agency). Three officials involved in the organization that regulates and controls worldwide anti-doping activities will verify the utilization of international technical standards called for in the Worldwide Anti-Doping Code. This posture also reveals dissatisfaction on the part of the international agency with the performance presented by the Brazilian entity, which was created in November 2011...	A ABCD (Autoridade Brasileira de Controle de Dopagem), orgao federal atrelado ao Ministerio do Esporte, esta passando nesta semana por uma auditoria da Wada (Agencia Mundial Antidoping). Tres oficiais enviados pela organizacao que regula o controle antidoping em nivel mundial vao verificar a aplicacao de padroes tecnicos internacionais previstos no Codigo Mundial Antidoping. A atitude tambem evidencia uma insatisfacao da agencia mundial com a conduta da entidade brasileira, que foi criada em novembro de 2011...

Fonte: Retirado de http://www1.folha.uol.com.br/esporte/2017/03/1868719-suspenso-brasil-passa-por-auditoria-da-agencia-mundial-antidoping.shtml no dia 24/03/2018.

6.2 Ferramentas

Todos os *scripts* produzidos neste projeto foram implementados em linguagem Python[3] com o auxílio de seus diversos módulos já existentes para a montagem das arquiteturas propostas para as redes neurais. Os *autoencoders* foram montados com os módulos: Theano[4] (Theano

[3]https://www.python.org/
[4]http://deeplearning.net/software/theano/

Development Team, 2016), Lasagne[5] (DIELEMAN, 2015), Tensorflow[6] (ABADI, 2016) e Keras[7] (CHOLLET et al., 2015). Para a rede BiDNN, foi utilizada a implementação original[8] disponibilizada por seu autor, a qual também faz uso de Lasagne e Theano.

Para a obtenção das representações parciais de texto e de imagem, conforme ilustrado nas Figuras 5.1 e 5.2, utilizamos as seguintes ferramentas:

- O módulo Gensim[9] (REHUREK; SOJKA, 2010), para gerar as representações de texto usando LSI, LDA e Word2Vec;
- O módulo Glove-Python[10], para gerar representações de texto usando GloVe;
- O módulo OpenCV[11] para manipulação de imagens e geração das representações usando SIFT, SURF e ORB;
- O módulo Keras, para geração dos vetores de imagem utilizando a VGG19;
- A biblioteca Scikit-Learn[12] (PEDREGOSA, 2012), para gerar os modelos de classificação onde as representações são comparadas.

6.3 Experimentos

Nesta seção são descritos os experimentos realizados com as duas bases de dados construídas para validação da proposta, e descritas na seção 6.1: (1) base de dados de produtos e (2) base de dados de notícias.

Os vetores de entrada são as representações unimodais e multimodais obtidas através dos métodos descritos nos Capítulos 2, 3 e 4. Os experimentos realizados com cada base de dados estão dividido em duas partes:

1. Classificação unimodal e multimodal, utilizando concatenação de vetores de características e fusão por meio de redes neurais. Nesse caso, a medida usada na avaliação é o F-Score obtido na tarefa de classificação, calculada considerando-se a estratégia de *10-fold cross-validation*;

[5]https://github.com/Lasagne/Lasagne
[6]https://www.tensorflow.org/
[7]https://keras.io/
[8]Disponível em: https://github.com/v-v/BiDNN
[9]https://radimrehurek.com/gensim/
[10]https://github.com/maciejkula/glove-python
[11]https://opencv.org/
[12]http://scikit-learn.org/

2. Análise de similaridade entre instâncias com representações unimodais e multimodais. Nesse caso, a medida usada na análise de similaridade é a distância euclidiana entre vetores de representação.

Cada método de representação unimodal gerou vetores de características com 128 dimensões, com a exceção dos vetores gerados pela rede VGG19 com 1.000 dimensões. As representações multimodais obtidas por meio de concatenação têm 256 dimensões, e as multimodais obtidas por meios neurais têm 64 dimensões. As representações multimodais foram geradas com base nas representações textuais e visuais de cada produto, para os experimentos descritos na seção 6.3.1; e com base nos textos originais em português e suas versões traduzidas para o inglês das notícias, para os experimentos descritos na seção 6.3.2.

6.3.1 Experimentos com a base de dados de Produtos

Na tarefa de classificação dos produtos presentes nesta base de dados, foram utilizados 3 algoritmos diferentes: um classificador linear probabilístico utilizando regressão logística, uma máquina de vetores de suporte (do inglês *Support Vector Machine*, SVM) e um Perceptron multi-camada totalmente conectado, com camadas escondidas de 256, 128 e 64 posições. Os resultados da primeira parte do experimento (avaliação da classificação) estão nas Tabelas 6.3 e 6.4.

Nos resultados apresentados nas Tabelas 6.3 e 6.4, utilizamos os três algoritmos citados anteriormente – classificação linear com regressão logística (SGD), máquina de vetores de suporte (SVM) e Perceptron multi-camada (MLP). Todas as combinações possíveis entre representações de texto e de imagem foram geradas e utilizadas. Estes resultados foram apresentados originalmente na publicação decorrente deste trabalho: (ITO; CASELI; MOREIRA, 2018).

No geral, as representações de texto tiveram performance muito superior às de imagem, tanto nos casos unimodais quanto nos multimodais. De acordo com os valores na Tabela 6.3, a melhor representação textual (com 87% de F-Score) foi a gerada com o LSI quando usada na MLP. Enquanto o melhor valor de F-Score usando apenas imagem obteve 30 pontos percentuais a menos (sendo de apenas 57% de F-Score) na MLP. Isto pode ser explicado, principalmente, por dois motivos: (1) a estrutura das descrições de cada produto valoriza palavras-chave e características específicas do produto que podem ser comparáveis (por exemplo, “802.11” e “IPv6” são palavras que aparecem em descrições de roteadores e produtos relacionados a redes Wi-Fi, com raras ocorrências em outros produtos), e (2) o fato de que produtos dentro de uma mesma classe podem ter atributos visuais completamente diferentes entre si (por exemplo, na classe

Tabela 6.3: F-Scores para a classificação de produtos utilizando características unimodais. Melhores resultados para cada modalidade estão destacados em negrito.

Modalidades	Representação	SGD	SVM	MLP
Texto	GloVe	0.71	0.74	0.80
	W2V	0.70	0.74	0.80
	LDA	0.62	0.68	0.75
	LSI	0.84	0.85	**0.87**
Imagem	SIFT	0.39	0.38	0.40
	SURF	0.36	0.38	0.36
	ORB	0.29	0.32	0.30
	VGG19	0.38	0.44	**0.57**
Imagem × Imagem (concatenação)	SIFT+SURF	0.45	0.42	0.43
	SIFT+ORB	0.41	0.40	0.41
	SIFT+VGG19	0.47	0.50	**0.53**
	SURF+ORB	0.38	0.38	0.39
	SURF+VGG19	0.46	0.49	0.52
	ORB+VGG19	0.45	0.45	0.49
Texto × Texto (concatenação)	GloVe+W2V	0.76	0.79	0.83
	GloVe+LDA	0.76	0.78	0.81
	GloVe+LSI	0.83	0.85	0.84
	W2V+LDA	0.75	0.78	0.82
	W2V+LSI	0.86	**0.88**	0.87
	LDA+LSI	0.84	0.86	0.84

"smartphones" são encontrados celulares, cabos USB e acessórios em geral). Resumidamente, os resultados parecem refletir as características da base de dados, na qual os atributos textuais são mais informativos e os atributos visuais são mais ruidosos.

A combinação das duas modalidades, apresentada na Tabela 6.4, não obteve resultados superiores aos já encontrados pelas *embeddings* de texto, sendo que o melhor resultado foi de 84% de F-Score. Entretanto, combinações entre representações da mesma modalidade tiveram resultados superiores aos encontrados pelas representações unimodais: a combinação entre LSI e W2V, duas representações obtidas por métodos totalmente diferentes, teve o melhor resultado dentre todas as representações utilizadas alcançando 88% de F-Score com o SVM.

Na maioria dos experimentos multimodais neurais (Tabela 6.4), o resultado da rede neural bidirecional (BiDNN) foi significativamente superior ao resultado do *autoencoder* multimodal simples (MMAE) com uma diferença de 10 pontos percentuais (77% de F-Score para a BiDNN contra 70% de F-Score para a MMAE). Os resultados são coerentes com os apresentados em (VUKOTIĆ; RAYMOND; GRAVIER, 2016), onde os dois algoritmos foram usados para a tarefa de criar *hyperlinks* para trechos de vídeos automaticamente, utilizando imagem e transcrições para classificação multimodal.

Tabela 6.4: F-Scores para a classificação de produtos utilizando características multimodais. Melhores resultados para cada modalidade estão destacados em negrito.

Modalidades	Representação	SGD	SVM	MLP
Imagem × Texto (concatenação)	SIFT+W2V	0.72	0.72	0.74
	SIFT+GloVe	0.74	0.71	0.80
	SIFT+LDA	0.66	0.68	0.69
	SIFT+LSI	**0.84**	**0.84**	**0.84**
	SURF+W2V	0.70	0.73	0.75
	SURF+GloVe	0.68	0.71	0.80
	SURF+LDA	0.65	0.66	0.67
	SURF+LSI	**0.84**	**0.84**	0.83
	ORB+W2V	0.68	0.71	0.73
	ORB+GloVe	0.70	0.72	0.76
	ORB+LDA	0.61	0.66	0.65
	ORB+LSI	0.83	0.84	0.81
	VGG19+W2V	0.73	0.74	0.78
	VGG19+GloVe	0.70	0.74	0.81
	VGG19+LDA	0.68	0.72	0.75
	VGG19+LSI	0.80	0.83	0.83
Imagem × Texto (MMAE)	SIFT+W2V	0.59	0.61	0.66
	SIFT+GloVe	0.45	0.44	0.52
	SIFT+LDA	0.51	0.54	0.60
	SIFT+LSI	0.33	0.33	0.31
	SURF+W2V	0.63	0.64	0.68
	SURF+GloVe	0.49	0.46	0.49
	SURF+LDA	0.48	0.49	0.58
	SURF+LSI	0.35	0.33	0.38
	ORB+W2V	0.58	0.61	**0.70**
	ORB+GloVe	0.38	0.40	0.44
	ORB+LDA	0.52	0.54	0.61
	ORB+LSI	0.31	0.33	0.39
	VGG19+W2V	0.60	0.61	0.67
	VGG19+GloVe	0.45	0.48	0.51
	VGG19+LDA	0.48	0.56	0.65
	VGG19+LSI	0.27	0.31	0.37
Imagem × Texto (BiDNN)	SIFT+W2V	0.48	0.60	0.75
	SIFT+GloVe	0.51	0.55	0.61
	SIFT+LDA	0.47	0.54	0.66
	SIFT+LSI	0.51	0.54	0.74
	SURF+W2V	0.49	0.55	0.72
	SURF+GloVe	0.51	0.52	0.61
	SURF+LDA	0.48	0.54	0.63
	SURF+LSI	0.52	0.59	**0.77**
	ORB+W2V	0.45	0.54	0.72
	ORB+GloVe	0.48	0.47	0.54
	ORB+LDA	0.43	0.50	0.62
	ORB+LSI	0.50	0.56	0.73
	VGG19+W2V	0.30	0.45	0.73
	VGG19+GloVe	0.37	0.42	0.54
	VGG19+LDA	0.21	0.40	0.67
	VGG19+LSI	0.27	0.44	0.76

Os resultados obtidos pelas representações multimodais neurais não foram melhores do que os obtidos pelas representações unimodais e multimodais por concatenação. Entretanto, ao analisar as distâncias entre os vetores de representação (Figuras 6.6, 6.5 e 6.7), podemos ver que as representações multimodais (na Figura 6.7) representam melhor a similaridade entre diferentes produtos, e podem ajudar a formar um vetor "conceitual" sobre uma determinada instância de dados. Acreditamos, neste momento, que o baixo desempenho das representações multimodais em relação às unimodais está mais relacionado aos ruídos nas representações de imagens ou às limitações dos modelos propostos, do que a possíveis limitações do potencial de aplicações multimodais. Propostas de trabalhos futuros são apresentadas no final deste documento para tentar investigar melhorias na abordagem testada.

Figura 6.5: Top-4 produtos similares usando características geradas pela VGG19.

Figura 6.6: Top-4 produtos similares usando características geradas pelo método LSI.

6.3.2 Experimentos com a base de dados de Notícias

Os experimentos com a base de dados de notícias foram similares aos anteriores, utilizando os mesmos algoritmos de classificação. Entretanto, nesta base de dados exploramos a multimodalidade entre diferentes linguagens (português e inglês), ou seja, a multilingualidade. Os resultados da primeira parte do experimento (avaliação da classificação) estão nas Tabelas 6.5 e 6.6.

Figura 6.7: Top-4 produtos similares usando características geradas pela junção das duas representações VGG19 e LSI.

Tabela 6.5: F-Scores para a classificação de notícias utilizando características unimodais. Melhores resultados para cada modalidade estão destacados em negrito.

Linguagens	Representação	SGD	SVM	MLP
Português	GloVe	0.55	0.62	0.55
	W2V	0.24	0.28	0.24
	LDA	0.22	0.19	0.41
	LSI	0.62	**0.65**	0.62
Inglês	GloVe	0.52	0.56	0.49
	W2V	0.29	0.19	0.27
	LDA	0.18	0.20	0.39
	LSI	0.54	**0.67**	0.61

Como pode ser visto pelos valores dessas tabelas, as representações multimodais e unimodais tiveram desempenho similar, com alguns resultados interessantes:

- Diferente do experimento unimodal anterior, o melhor modelo não foi o Perceptron multicamada, perdendo em muitos casos para as algoritmos mais simples;
- Uma das representações textuais mais conhecidas, o Word2Vec, teve desempenho muito inferior ao GloVe e ao LSI;
- A melhor representação obtida nesta parte do experimento vem de uma abordagem multimodal *early-stage*, a combinação entre LSI português e LDA inglês;

Como esta base de dados tem menos instâncias para comparação, as abordagens neurais não tiveram tanta eficiência, tanto em classificação (com a MLP tendo resultados menos relevantes) quanto em representação (W2V tendo menor F-Score do que as outras representações). O F-Score baixo pode ser explicado pelo mesmo fato: menos instâncias de treino traz como consequência um maior peso em erros pontuais.

Tabela 6.6: F-Scores para a classificação de notícias utilizando características multimodais. Melhores resultados para cada modalidade estão destacados em negrito.

Linguagens	Representação	SGD	SVM	MLP
Português × Inglês (concatenação)	GloVe+GloVe	0.48	0.53	0.56
	GloVe+W2V	0.53	0.55	0.54
	GloVe+LDA	0.54	0.54	0.52
	GloVe+LSI	0.57	0.57	0.56
	W2V+GloVe	0.53	0.55	0.62
	W2V+W2V	0.19	0.21	0.27
	W2V+LDA	0.25	0.26	0.34
	W2V+LSI	0.67	0.66	0.63
	LDA+GloVe	0.54	0.54	0.60
	LDA+W2V	0.25	0.23	0.39
	LDA+LDA	0.21	0.19	0.44
	LDA+LSI	0.62	0.55	0.51
	LSI+GloVe	0.52	0.62	0.56
	LSI+W2V	0.63	0.64	0.63
	LSI+LDA	0.59	0.63	**0.71**
	LSI+LSI	0.65	0.67	0.53
Português × Inglês (MMAE)	GloVe+GloVe	0.35	0.43	**0.45**
	GloVe+W2V	0.32	0.32	0.33
	GloVe+LDA	0.35	0.37	0.44
	GloVe+LSI	0.36	0.40	0.40
	W2V+GloVe	0.31	0.31	0.33
	W2V+W2V	0.19	0.25	0.25
	W2V+LDA	0.26	0.26	0.32
	W2V+LSI	0.25	0.23	0.23
	LDA+GloVe	0.32	0.35	0.39
	LDA+W2V	0.26	0.26	0.27
	LDA+LDA	0.30	0.31	0.37
	LDA+LSI	0.26	0.19	0.28
	LSI+GloVe	0.37	0.36	0.37
	LSI+W2V	0.27	0.25	0.26
	LSI+LDA	0.20	0.19	0.27
	LSI+LSI	0.19	0.19	0.19
Português × Inglês (BiDNN)	GloVe+GloVe	0.48	0.53	0.56
	GloVe+W2V	0.25	0.29	0.28
	GloVe+LDA	0.20	0.23	0.42
	GloVe+LSI	0.63	**0.68**	0.61
	W2V+GloVe	0.28	0.26	0.32
	W2V+W2V	0.19	0.19	0.25
	W2V+LDA	0.17	0.18	0.45
	W2V+LSI	0.24	0.40	0.57
	LDA+GloVe	0.25	0.31	0.42
	LDA+W2V	0.20	0.21	0.41
	LDA+LDA	0.15	0.21	0.42
	LDA+LSI	0.40	0.35	0.39
	LSI+GloVe	0.55	0.62	0.52
	LSI+W2V	0.41	0.37	0.56
	LSI+LDA	0.35	0.33	0.50
	LSI+LSI	0.46	0.53	0.61

Os resultados multimodais, apresentados na Tabela 6.6 são comparáveis aos do outro experimento na Seção 6.3.1: os resultados não chegam a ser melhores do que os apresentados pelas representações unimodais, com o modelo bidirecional sendo superior ao *autoencoder* simples. Esperava-se que o F-Score nos experimentos multimodais neurais fosse maior do que as abordagens unimodais, pelo fato de que existem poucas instâncias no banco de dados: mais informações extraídas trariam mais unicidade a cada instância, facilitando a classificação. Isto aconteceu na Tabela 6.6, onde a junção das representações LSI em português e inglês trouxe um resultado um pouco superior aos apresentados pelas representações unimodais parciais.

Como no experimento anterior, a utilização de representações multimodais não melhorou o escore obtido na tarefa de classificação, mas pode ser utilizada para melhores comparações entre notícias individuais. A Figura 6.8 mostra as notícias mais próximas da notícia *After 20 Years, Presence of Rare Harpy Eagle Is Registered in São Paulo* utilizando representações unimodais e multimodais.

Figura 6.8: Tabela com as Top-4 notícias mais próximas da notícia em negrito

Representação	**After 20 Years, Presence of Rare Harpy Eagle Is Registered in São Paulo**	Distância
Representação em Português (LSI)	Dinosaur Fossil Is Found on Construction Site in Minas Gerais	0,2634
	Enigma of the Amazon Earthworks Solved	0,3065
	Brazilian Researchers Find Ancestor of Dinosaurs	0,3081
	Robbery Suspect Is Tied to Pole and Beaten to Death in the State of Maranhão	0,3100
Representação em Inglês (LSI)	'Prove That I Am Corrupt and I Will Walk to Jail,' Says Lula	0,2803
	Factory Projected to Manufacture Brazilian Fighter Jet Parts Still Far From Becoming a Reality	0,2814
	Owner of JBS Receives 'Discount' in Agreement with Attorney General's Office	0,2849
	Birdwatching Gets New Destinations in Brazil	0,2858
Representação Multimodal (LSI Português * LSI Inglês)	Dinosaur Fossil Is Found on Construction Site in Minas Gerais	0,2954
	Brazilian Researchers Find Ancestor of Dinosaurs	0,3016
	Birdwatching Gets New Destinations in Brazil	0,3025
	Factory Projected to Manufacture Brazilian Fighter Jet Parts Still Far From Becoming a Reality	0,3091

No caso da língua portuguesa, sua representação unimodal consegue capturar o tema geral da notícia, mas não completamente: a quarta notícia mais próxima (*Robbery Suspect Is Tied to Pole and Beaten to Death in the State of Maranhão*) é sobre violência, tema completamente diferente do tom científico da notícia-alvo. No caso da língua inglesa, somente a quarta notícia mais próxima (*Birdwatching Gets New Destinations in Brazil*) tem relação direta com o tema da notícia alvo. Já a representação multimodal consegue capturar duas das notícias relevantes encontradas pela representação em português (sobre fósseis), a notícia relevante encontrada

pela representação em inglês (sobre observação de aves) e adiciona uma notícia sobre indústria aeronáutica, colocada entre as mais próximas por ter palavras em seu texto relacionadas a voo, aerodinâmica, e por se referir a uma fábrica na cidade de “Gavião” Peixoto.

Capítulo 7

Considerações Finais

Atualmente, a maior parte das notícias e informações que acessamos diariamente pela Internet utiliza múltiplas modalidades de dados: notícias são acompanhadas de imagens, vídeos ou *podcasts*; produtos têm vídeos de demonstração de usos, imagens ilustrativas e textos detalhando suas funcionalidades e detalhes técnicos; propagandas são criadas com músicas e elementos visuais que remetem rapidamente à ideia de seus produtos. O cérebro humano aprende melhor com múltiplas modalidades de informação trabalhando juntas, e devemos utilizar este conceito na computação para continuar aperfeiçoando a qualidade dos algoritmos de aprendizado de máquina.

Neste trabalho, exploramos o efeito que informações de diferentes tipos (modalidades) podem ter em uma representação única, suas aplicações práticas e descrevemos alguns dos métodos disponíveis na literatura para geração de representações uni e multimodais. Nos experimentos apresentados neste trabalho, notamos que representações multimodais têm o potencial de melhorar a comparação entre instâncias de dados (como descrito na Seção 6.3.1) e a acurácia de tarefas de classificação, mesmo quando utilizadas de maneira simplificada (como descrito na Seção 6.3.2).

Também provamos nossa hipótese de que diferentes representações unimodais levam a representações multimodais diferentes, e que nem sempre as abordagens neurais são os melhores pontos de partida para o desenvolvimento de uma representação de qualidade: a representação tópica LSI, simples e rápida, obteve os melhores resultados em muitos de nossos experimentos. O que concluímos com esses resultados é que não podemos deixar de utilizar métodos simples em favor de abordagens mais complicadas e custosas, e sempre devemos testar algoritmos bem-fundamentados antes de aumentar a complexidade de nossos sistemas inteligentes.

Dentre as contribuições desta dissertação estão:

- Duas bases de dados multimídia (bases de Produtos e de Notícias), com textos em português;
- Descrição detalhada de algumas das mais usadas representações de texto e de imagem na academia;
- *Baseline* de resultados de classificação para futuros experimentos nas bases de dados de produtos e notícias e;
- Primeiras comparações entre representações unimodais afim de melhorar a geração de representações multimodais, conforme descrito neste documento e no trabalho apresentado no LREC 2018 (ITO; CASELI; MOREIRA, 2018).

As bases de dados e códigos gerados neste trabalho serão disponibilizados na página do LALIC: www.lalic.dc.ufscar.br.

Como trabalhos futuros, propomos:

1. Utilizar diferentes bancos de dados com modalidades diferentes (áudio e vídeo, por exemplo) para comparar as representações unimodais e multimodais utilizadas neste trabalho;
2. Comparar diferentes representações aplicadas nos bancos de dados fornecidos neste trabalho (diferentes descritores de imagem e de texto, diferentes métodos de fusão multimodal);
3. Expansão da base de dados para novos experimentos, criação de novas bases de dados de produtos e notícias utilizando fontes diferentes;
4. Alinhar imagens e texto (VELTRONI, 2018), criando pares para melhorar representações multimodais de palavras específicas.

REFERÊNCIAS

ABADI, M. et al. TensorFlow: A system for large-scale machine learning. *Google Brain*, p. 18, 2016. Disponível em: <http://arxiv.org/abs/1605.08695>.

ACKLEY, D.; HINTON, G.; SEJNOWSKI, T. A learning algorithm for boltzmann machines. *Cognitive Science*, v. 9, n. 1, p. 147–169, 1985. ISSN 03640213. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0364021385800124>.

ANDREW, G. et al. Deep Canonical Correlation Analysis. *Proceedings of The 30th International Conference on Machine Learning*, v. 28, p. 1247–1255, 2013.

ATREY, P. K. et al. Multimodal fusion for multimedia analysis: A survey. *Multimedia Systems*, v. 16, n. 6, p. 345–379, 2010. ISSN 09424962.

BARONI, M.; DINU, G.; KRUSZEWSKI, G. Don't count , predict ! A systematic comparison of context-counting vs . context-predicting semantic vectors. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics.*, p. 238–247, 2014. ISSN 1529-1898.

BAY, H. et al. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, v. 110, n. 3, p. 346–359, 2008. ISSN 10773142.

Beatriz Ribas. Infografia Multimídia: um modelo narrativo para o webjornalismo. *Grupo de Pesquisa em Jornalismo On-line*, n. V, p. 16, 2004. Disponível em: <http://www.facom.ufba.br/jol/pdf/2004_5iberoamericano_salvador_infografia.pdf>.

BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation Learning: A Review and New Perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 35, n. 8, p. 1798–1828, 2013. ISSN 1939-3539.

BENGIO, Y. et al. A Neural Probabilistic Language Model. *The Journal of Machine Learning Research*, v. 3, p. 1137–1155, 2003. ISSN 15324435.

BENGIO, Y. et al. Greedy Layer-Wise Training of Deep Networks. *Advances in Neural Information Processing Systems*, v. 19, n. 1, p. 153, 2007. ISSN 01628828. Disponível em: <http://papers.nips.cc/paper/3048-greedy-layer-wise-training-of-deep-networks.pdf>.

BENVENUTO, N.; PIAZZA, F. The backpropagation algorithm. *IEEE Transactions on Signal Processing*, v. 40, n. 4, p. 967–969, 1992. ISSN 1053587X.

BERARDI, G.; ESULI, A.; MARCHEGGIANI, D. Word embeddings go to Italy: A comparison of models and training datasets. In: *CEUR Workshop Proceedings*. [S.l.: s.n.], 2015. v. 1404. ISSN 16130073.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *J. Mach. Learn. Res.*, v. 3, p. 993–1022, 2003. ISSN 1532-4435. Disponível em: <http://dx.doi.org/10.1162/jmlr.2003.3.4-5.993\nhttp://portal.acm.org.proxy.lib.utk.edu:90/citation.cfm?id=944937\nhttp://portal.acm.org.proxy.lib

CALONDER, M. et al. BRIEF: Binary robust independent elementary features. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [S.l.: s.n.], 2010. v. 6314 LNCS, p. 778–792. ISBN 364215560X. ISSN 03029743.

CHOLLET, F. et al. *Keras*. [S.l.]: GitHub, 2015. Https://github.com/keras-team/keras.

DEERWESTER, S.; DUMAIS, S. T.; HARSHMAN, R. Indexing by latent semantic analysis. *Journal of the American society for information science*, v. 41, n. 6, p. 391–407, 1990. ISSN 0002-8231. Disponível em: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Indexing+by+Latent+Semantic+Analy

DIELEMAN, S. et al. *Lasagne: First release.* ago. 2015. Disponível em: <http://dx.doi.org/10.5281/zenodo.27878>.

DUMOULIN, V.; VISIN, F. A guide to convolution arithmetic for deep learning. *Arxiv*, p. 1–28, 2016. ISSN 16113349. Disponível em: <http://arxiv.org/abs/1603.07285>.

ECKART, C.; YOUNG, G. The approximation of one matrix by another of lower rank. *Psychometrika*, Springer, v. 1, n. 3, p. 211–218, 1936.

ESKEVICH, M. et al. The search and hyperlinking task at MediaEval 2014. In: *CEUR Workshop Proceedings*. [S.l.: s.n.], 2014. v. 1263. ISSN 16130073.

GARDNER, H. *Frames of Mind: The Theory of Multiple Intelligences*. 1985. 463 p. Disponível em: <http://books.google.com/books?hl=en&lr=&id=2IEfFSYouKUC&oi=fnd&pg=PR7&dq=frames+of+min 6301-497B-8D19-ADE67261C433>.

HAAR, A. Zur Theorie der orthogonalen Funktionensysteme (On the theory of orthogonal function systems). *Math. Ann.*, v. 69, p. 331–371, 1910.

HARDOON, D. R.; SZEDMAK, S.; SHAWE-TAYLOR, J. Canonical correlation analysis: an overview with application to learning methods. *Neural computation*, v. 16, n. 12, p. 2639–64, 2004. ISSN 0899-7667. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/15516276>.

HARRIS, C.; STEPHENS, M. A Combined Corner and Edge Detector. In: *Procedings of the Alvey Vision Conference 1988*. [s.n.], 1988. p. 23.1–23.6. ISSN 09639292. Disponível em: <http://www.bmva.org/bmvc/1988/avc-88-023.html>.

HARRIS, Z. S. Distributional Structure. *Word*, v. 10, n. 2-3, p. 146–162, 1954. ISSN 0043-7956. Disponível em: <http://psycnet.apa.org/psycinfo/1956-02807-001>.

HERVÉ, A. The Eigen-Decomposition: Eigenvalues and Eigenvectors. *Encyclopedia of Measurements and Statistics*, p. 1–10, 2007.

HILL, R. J. et al. The Measurement of Meaning. *American Sociological Review*, v. 23, n. 2, p. 227, 1958. ISSN 00031224.

HOFMANN, T. Probabilistic latent semantic indexing. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '99*, p. 50–57, 1999. ISSN 15206882. Disponível em: <http://portal.acm.org/citation.cfm?doid=312624.312649>.

HOPPE, H. et al. Surface reconstruction from unorganized points. *ACM SIGGRAPH Computer Graphics*, v. 26, n. 2, p. 71–78, 1992. ISSN 00978930.

HOTELLING, H. Relations Between Two Sets of Variates. *Biometrika*, v. 28, n. 3, p. 321–377, 1936. ISSN 0006-3444. Disponível em: <http://www.springerlink.com/index/10.1007/978-1-4612-4380-9_14%5Cnpapers2://publication/doi/10.1007/978-1-4612-4380-9_14>.

HUBEL, D. H.; WIESEL, T. N. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, v. 195, n. 1, p. 215–43, 1968. ISSN 0022-3751. Disponível em: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1557912&tool=pmcentrez&rendertype=abstr

ICMI 2016: Proceedings of the 18th ACM International Conference on Multimodal Interaction. New York, NY, USA: ACM, 2016. ISBN 978-1-4503-4556-9.

ITO, F. T.; CASELI, H. d. M.; MOREIRA, J. The effects of unimodal representation choices on multimodal learning. In: *Proceedings of Language Resources and Evaluation Conference*. [S.l.: s.n.], 2018.

ITO, M. et al. Size and position invariance of neuronal responses in monkey inferotemporal cortex. *Journal of neurophysiology*, v. 73, n. 1, p. 218–226, 1995. ISSN 0022-3077.

JAITLY, N.; HINTON, G. Learning a better representation of speech soundwaves using restricted boltzmann machines. In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. [S.l.: s.n.], 2011. p. 5884–5887. ISBN 9781457705397. ISSN 15206149.

Jia Deng et al. ImageNet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. [s.n.], 2009. p. 248–255. ISBN 978-1-4244-3992-8. ISSN 1063-6919. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5206848>.

JOACHIMS, T. Text categorization with support vector machines: Learning with many relevant features. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [S.l.: s.n.], 1998. v. 1398, p. 137–142. ISBN 3540644172. ISSN 16113349.

KRAMER, M. A. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, v. 37, n. 2, p. 233–243, 1991. ISSN 15475905.

KUSNER, M. J. et al. From Word Embeddings To Document Distances. *Proceedings of The 32nd International Conference on Machine Learning*, v. 37, p. 957–966, 2015.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2323, 1998. ISSN 00189219.

LECUN, Y.; CORTES, C. MNIST handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2010.

LEONARDIS, A. et al. *Computer Vision – ECCV 2006 SURF: Speeded Up Robust Features*. [S.l.: s.n.], 2006. 404–417–417 p. ISBN 978-3-540-33832-1.

LEVY, O.; GOLDBERG, Y.; DAGAN, I. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics*, v. 3, p. 211–225, 2015. ISSN 2307-387X. Disponível em: <https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/570>.

LEWIS, J. P. Fast Template Matching. *Pattern Recognition*, v. 10, n. 11, p. 120–123, 1995. ISSN 1057-7149. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.157.3888&rep=rep1&type=pdf>.

LI, T. et al. Contextual bag-of-words for visual categorization. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 21, n. 4, p. 381–392, 2011. ISSN 10518215.

LIU, B. Sentiment Analysis and Subjectivity. *Handbook of Natural Language Processing*, n. 1, p. 1–38, 2010. ISSN 17422094. Disponível em: <http://www.cs.uic.edu/ liub/FBS/NLP-handbook-sentiment-analysis.pdf%5Cnhttp://people.sabanciuniv.edu/berrin/proj102/1-BLiu-Sentiment Analysis and Subjectivity-NLPHandbook-2010.pdf%5Cnhttp://www.cs.uic.edu/ liub/FBS/NLP-handbook-sentiment-analysis.pdf$%5Cbacksl>.

LOWE, D. G. Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, v. 2, n. [8, p. 1150–1157, 1999. ISSN 0-7695-0164-8. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=790410>.

MANNING, C. D.; RAGHAVAN, P. *An Introduction to Information Retrieval*. Cambridge University Press, 2009. 1 p. Disponível em: <http://dspace.cusat.ac.in/dspace/handle/123456789/2538>.

MASCI, J. et al. Stacked convolutional auto-encoders for hierarchical feature extraction. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [S.l.: s.n.], 2011. v. 6791 LNCS, p. 52–59. ISBN 9783642217340. ISSN 03029743.

MIKOLOV, T. et al. Efficient Estimation of Word Representations in Vector Space. *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, p. 1–12, 2013. ISSN 15324435. Disponível em: <http://arxiv.org/pdf/1301.3781v3.pdf>.

MITCHELL, J.; LAPATA, M. Composition in distributional models of semantics. *Cognitive science*, v. 34, n. 8, p. 1388–1429, 2010. ISSN 1551-6709.

MUNHALL, K. G.; VATIKIOTIS-BATESON, E.; TOHKURA, Y. X-ray film database for speech research. *The Journal of the Acoustical Society of America*, v. 98, n. 2, p. 1222–1224, 1995. ISSN 0001-4966. Disponível em: <http://scitation.aip.org/content/asa/journal/jasa/98/2/10.1121/1.413621\nhttp://scitation.aip.org/deliver/fu aip-live-02?itemId=/content/asa/journal/jasa/98/2/10.1121/1.413621&mimeType=pdf>.

NAVIGLI, R. Word sense disambiguation. *ACM Computing Surveys*, v. 41, n. 2, p. 1–69, 2009. ISSN 03600300. Disponível em: <http://portal.acm.org/citation.cfm?doid=1459352.1459355>.

NGIAM, J. et al. Multimodal Deep Learning. *Proceedings of The 28th International Conference on Machine Learning (ICML)*, p. 689–696, 2011.

NIKHIL, R.; SCANSAR, K. A review on image segmentation techniques. *Pattern Recognition*, v. 26, n. 9, p. 1277–1294, 1993. ISSN 00313203.

O'CONNOR, P. et al. Real-time classification and sensor fusion with a spiking deep belief network. *Frontiers in Neuroscience*, n. 7 OCT, 2013. ISSN 16624548.

PANG, B.; LEE, L. Opinion Mining and Sentiment Analysis. *Foundations and Trends® in InformatioPang, B., & Lee, L. (2006). Opinion Mining and Sentiment Analysis. Foundations and Trends® in Information Retrieval, 1(2), 91–231. doi:10.1561/1500000001n Retrieval*, v. 2, n. 1-2, p. 91–231, 2008. ISSN 1554-0669. Disponível em: <http://www.nowpublishers.com/product.aspx?product=INR&doi=1500000001>.

PAPERT, S.; TECHNOLOGY. ARTIFICIAL INTELLIGENCE LABORATORY, M. I. of. *The Summer Vision Project*. Massachusetts Institute of Technology, Project MAC, 1966. (AI memo). Disponível em: <https://books.google.com.br/books?id=qOh7NwAACAAJ>.

PATTERSON, E. et al. CUAVE: A new audio-visual database for multimodal human-computer interface research. *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, v. 2, p. II–2017–II–2020, 2002. ISSN 1520-6149.

PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2012. ISSN 15324435. Disponível em: <http://dl.acm.org/citation.cfm?id=2078195\nhttp://arxiv.org/abs/1201.0490>.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, p. 1532–1543, 2014. ISSN 10495258.

RADFORD, A.; METZ, L.; CHINTALA, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv*, p. 1–15, 2015. ISSN 0004-6361. Disponível em: <http://arxiv.org/abs/1511.06434>.

REHUREK, R.; SOJKA, P. Software Framework for Topic Modelling with Large Corpora. *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, p. 45–50, 2010. ISSN 2951740867.

ROSIN, P. L. Measuring Corner Properties. *Computer Vision and Image Understanding*, v. 73, n. 2, p. 291–307, 1999. ISSN 10773142.

ROSTEN, E.; PORTER, R.; DRUMMOND, T. Faster and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 32, n. 1, p. 105–119, 2010. ISSN 01628828.

RUBLEE, E. et al. ORB: An efficient alternative to SIFT or SURF. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2011. p. 2564–2571. ISBN 9781457711015. ISSN 1550-5499.

RUDER, S. An overview of gradient descent optimization algorithms. *Web Page*, p. 1–12, 2016. Disponível em: <http://arxiv.org/abs/1609.04747>.

SALAKHUTDINOV, R.; HINTON, G. Deep belief networks. *Scholarpedia*, p. 4–5, 2009. ISSN 1941-6016. Disponível em: <http://scholarpedia.org/article/Deep_Belief_Networks>.

SALAKHUTDINOV, R.; MNIH, A.; HINTON, G. Restricted Boltzmann machines for collaborative filtering. *Proceedings of the 24th international conference on Machine learning - ICML '07*, p. 791–798, 2007. Disponível em: <http://portal.acm.org/citation.cfm?doid=1273496.1273596>.

SALTON, G.; BUCKLEY, C. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, v. 24, n. 5, p. 513–523, 1988. ISSN 03064573.

SIMONYAN, K.; ZISSERMAN, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations (ICRL)*, p. 1–14, 2015. ISSN 09505849. Disponível em: <http://arxiv.org/abs/1409.1556>.

SOBEL, I.; FELDMAN, G. A 3x3 isotropic gradient operator for image processing. *in Hart, P. E. & Duda R. O. Pattern Classification and Scene Analysis*, p. 271–272, 1973. Disponível em: <papers2://publication/uuid/F6C98D8E-0A99-40EF-A91C-0ECA53448D1F>.

STAMOU, G.; KOLLIAS, S. *Multimedia Content and the Semantic Web: Standards, Methods and Tools*. [S.l.: s.n.], 2005. 1–392 p. ISBN 9780470012611.

STEINBACH, M.; KARYPIS, G.; KUMAR, V. A Comparison of Document Clustering Techniques. In: *KDD workshop on text mining*. [s.n.], 2000. v. 400, p. 1–2. ISBN 9781424428748. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4721382>.

STEWART, G. W. *On the Early History of the Singular Value Decomposition*. 1993. 551–566 p.

SZELISKI, R. Computer Vision : Algorithms and Applications. *Computer*, v. 5, p. 832, 2010. ISSN 10636919. Disponível em: <http://research.microsoft.com/en-us/um/people/szeliski/book/drafts/szelski_20080330am_draft.pdf>.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, maio 2016. Disponível em: <http://arxiv.org/abs/1605.02688>.

TURK, M. M.; PENTLAND, A. A. *Face Recognition Using Eigenfaces*. 1991. 72 – 86 p. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=139758\nhttp://ieeexplore.ieee.org/lpdocs/epic03/w

Van Tulder, G.; De Bruijne, M. Combining Generative and Discriminative Representation Learning for Lung CT Analysis With Convolutional Restricted Boltzmann Machines. *IEEE Transactions on Medical Imaging*, v. 35, n. 5, p. 1262–1272, 2016. ISSN 1558254X.

VELTRONI, W. C. *Alinhamento texto-imagem em sites de notícias*. 96 p. Dissertação (Mestrado) — Programa de Pós-graduação em Ciência da Computação – Universidade Federal de São Carlos (UFSCar), 2018.

VINCENT, P. et al. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion Pierre-Antoine Manzagol. *Journal of Machine Learning Research*, v. 11, p. 3371–3408, 2010. ISSN 15324435.

VUKOTIĆ, V.; RAYMOND, C.; GRAVIER, G. Bidirectional Joint Representation Learning with Symmetrical Deep Neural Networks for Multimodal and Crossmodal Applications. In: *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval - ICMR '16*. [s.n.], 2016. p. 343–346. ISBN 9781450343596. Disponível em: <http://dl.acm.org/citation.cfm?doid=2911996.2912064>.

WANG, W. et al. On Deep Multi-View Representation Learning: Objectives and Optimization. *Arxiv Preprint*, XX, n. XX, p. XX, 2016. ISSN 10414347. Disponível em: <http://arxiv.org/abs/1602.01024>.

WESTBURY, J. *X-ray Microbeam Speech Production Database User's Handbook: Version 1.0 (June 1994)*. Waisman Center on Mental Retardation & Human Development, 1994. Disponível em: <https://books.google.com.br/books?id=H1HTjwEACAAJ>.

WITKIN, A. P. Scale-space filtering. *International Joint Conference on Artificial Intelligence*, v. 2, p. 1019–1022, 1983. Disponível em: <http://portal.acm.org/citation.cfm?id=1623607>.

ZHANG, Y.; JIN, R.; ZHOU, Z. H. Understanding bag-of-words model: A statistical framework. *International Journal of Machine Learning and Cybernetics*, v. 1, n. 1-4, p. 43–52, 2010. ISSN 18688071.

GLOSSÁRIO

CNN – *Convolutional Neural Network*

DCCAE – *Deep canonically correlated autoencoder*

DCCA – *Deep Canonical Correlation Analysis*