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

ON THE TRAINING ALGORITHMS FOR RESTRICTED BOLTZMANN MACHINE-BASED MODELS

Leandro Aparecido Passos Júnior

Orientador: Prof. Dr. João Paulo Papa

São Carlos – SP
Dezembro/2018

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ON THE TRAINING ALGORITHMS FOR RESTRICTED BOLTZMANN MACHINE-BASED MODELS

Leandro Aparecido Passos Júnior

Tese 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 Doutor em Ciência da Computação, área de concentração: Processamento de Imagens e Sinais
Orientador: Prof. Dr. João Paulo Papa

São Carlos – SP

Dezembro/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 Tese de Doutorado do candidato Leandro Aparecido Passos Junior, realizada em 05/12/2018:

Prof. Dr. Ricardo Cerri
UFSCar

João Paulo Papa.

Prof. Dr. João Paulo Papa
UFSCar

Alexandre Luis M. Levada

Prof. Dr. Alexandre Luis Magalhães Levada
UFSCar

Fabricio Aparecido Breve

Prof. Dr. Fabricio Aparecido Breve
UNESP

Prof. Dr. Daniel Carlos Guimarães Pedronette
UNESP

Certifico que a defesa realizou-se com a participação à distância do(s) membro(s) João Paulo Papa 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.

Prof. Dr. Ricardo Cerri

À minha mãe, Silvia Aparecida Bianco.

Agradecimentos

Agradeço primeiramente à minha *mãe*, por todo amor, apoio e dedicação. Agradeço também ao meu *pai*, pela amizade e companheirismo.

Agradeço às minhas amadas irmãs *Dé* e *Kel*, que sobreviveram ao meu mau-humor por todos estes anos e ainda assim são capazes de continuar me amando. À *Gi* e ao *Lipe*, por todo carinho.

Agradeço aos meus falecidos *avôs* e minhas queridas *avós*, por todo ensinamento e lição de vida.

Agradeço à minha doce namorada *Amanda*, pessoa maravilhosa que espero ter ao meu lado por todos os meus dias.

Agradeço aos meus *tios*, *primos*, à minha sogra *Ana Maria*, ao meu cunhado *Dyaulas*, à minha sobrinha *Ana Luisa* (xodózinho), e a todos os meus *familiares*, que sempre me apoiaram durante todo o percurso.

Agradeço aos meus amigos de (muito) longa data: *Dogrão*, *Júnior*, *Angelo*, *Rhuber*, *Leon* e *Diogo*, aos que conheci na jornada musical: *Sabatino*, *André Ferrari*, *Guga*, *Otávio*, *João* e *Picello*, a todos os amigos que fiz enquanto trabalhei na empresa *STi3*, e a todos os outros que fizeram minha vida repleta de momentos felizes.

Agradeço a todo o pessoal do **Recogna**[1]. Em especial aos queridos amigos e companheiros de viagem *Clayton*, *Kelton*, *Luis Japa*, *Douglas* e *Rafael.* Agradeço também ao *Raniere*, que acabou se aproximando do nosso grupo e se tornou um grande amigo.

Agradeço ao pessoal da UFSCAR, em especial aos professores *Nilceu*, *Cerri* e *Levada*, que colaboraram muito com minha evolução acadêmica.

E como não poderia faltar, gostaria de agradecer especialmente ao meu grande mestre e querido amigo, Professor *João Paulo Papa.* Minha gratidão será eterna!

[1] http://www.recogna.tech/

Resumo

Técnicas de aprendizado em profundidade têm sido amplamente investigadas pela comunidade científica nos últimos anos, principalmente devido ao seu bom desempenho em tarefas tidas como essenciais em diversas aplicações, tais como reconhecimento de faces e comandos por voz, bem como classificação de objetos. Um dos métodos mais empregados é o das Máquinas de Boltzmann Restritas, do inglês *Restricted Boltzmann Machines* (RBMs), as quais são, basicamente, redes neurais estocásticas que objetivam estimar os pesos das conexões entre camadas distintas utilizando, dentre algumas técnicas, aquelas baseadas em amostragem em cadeias de Markov. Atualmente, grande parte dos trabalhos científicos têm concentrado sua atenção em métodos de amostragem nessas cadeias, dado que a sua eficiência e eficácia estão intimamente ligadas ao sucesso do processo de treinamento de uma RBM. Assim, a presente Tese contribui na área de aprendizado de RBMs, bem como de suas variantes chamadas de *Deep Belief Networks* e *Deep Boltzmann Machines*. Métodos de otimização para seleção dos parâmetros dessas técnicas também são estudados e validados no contexto de reconstrução de imagens e reconhecimento de padrões. De uma maneira geral, esta Tese objetiva estabelecer paralelos entre diferentes abordagens de treinamento dessas técnicas, bem como estudar e avaliar a eficiência de seu treinamento por meio de técnicas meta-heurísticas. Além disso, a proposta apresenta uma coleção de trabalhos desenvolvidos pelo autor durante o período de estudo, que foram publicados/submetidos para publicação em periódicos e conferências até o presente momento, sendo eles relacionados à: (i) inclusão do parâmetro temperatura na formulação da DBM, (ii) utilização de temperatura adaptativa para DBM, (iii) otimização dos meta-parâmetros da DBM utilizando técnicas meta-heurísticas e (iv) otimização dos meta-parâmetros da iRBM utilizando técnicas meta-heurísticas.

Palavras-chave: Aprendizado de Máquina, Restricted Boltzmann Machine, Otimização

Abstract

Deep learning techniques have been studied extensively in the last years, due to its good results related to essential tasks on a large range of applications, such as speech and face recognition, as well as objects classification. Among the most employed techniques is the Restrict Boltzmann Machines (RBMs), which are energy-based stochastic neural networks composed of two layers of neurons., i.e., visible and hidden, whose objective is to estimate the connection weights between both layers, generally using Markov chains. Recently, the scientific community spent many efforts on sampling methods, since RBMs effectiveness is directly related to the success of the sampling process. Thereby, the present work contributes with RBMs Learning area, as well as its variants DBNs and DBMs. Further, the work covers the application of meta-heuristic methods concerning a proper fine-tune of these techniques. Moreover, the validation of the model is presented in the context of image reconstruction and pattern recognition. In general, the present work presents different approaches to training these techniques, as well as the evaluation of meta-heuristic methods efficiency in training. Finally, this thesis presents a collection of works developed by the author during the study period, which was published/submitted until the present time, concerning: (i) temperature parameter introduction in DBM formulation, (ii) DBM using adaptive temperature, (iii) DBM meta-parameters optimization through meta-heuristic techniques, and (iv) iRBM meta-parameters optimization through meta-heuristic techniques.

Keywords: Machine Learning, Restricted Boltzmann Machine, Optimization

List of Figures

List of tables

Summary

Chapter 1
Introduction

In the last decades, machine learning techniques employment has grown exponentially in a wide range of applications, even more, the ones regarding decision-making tasks. Such tasks become of extreme interest in environments that involve large amounts of data, such as laboratory diagnosis, image and video processing, and data mining, just to cite a few.

Usually, the traditional data flow employed to "solve" machine learning related problems tend to follows four mains steps: (i) data processing, (ii) feature extraction, (iii) feature selection/transformation, and (iv) pattern recognition. Although each of the aforementioned steps had evolved in the last decades, a new set of techniques based in deep learning (DL) strategies provide an approach that mimics the brain behavior while processing visual information, where the data extraction is performed on distinct layers, and each layer is responsible for extracting different kinds of information.

Convolutional Neural Networks (CNNs) (LECUN et al., 1998) and Restricted Boltzmann Machines (RBMs) (SMOLENSKY, 1986b) are among the most used techniques nowadays. CNNs model the hierarchical information processing performed by the human brain in a natural manner, which is composed of three main steps: (i) application of convolutions using distinct filters over the input signal, (ii) signal sampling, and (iii) a normalization process. On the other hand, RBMs are classified as stochastic neural networks composed of a set of "hidden" or latent units employed to encode a representation of a set of input data. Honestly speaking, RBMs are not a DL method itself, though its "stacking" process characterizes it. In a nutshell, RBMs are used as building blocks for deep learning models, such as the well-known Deep Belief Network (DBNs) (HINTON; OSINDERO; TEH, 2006a) and the Deep Boltzmann Machines(DBMs) (SALAKHUTDINOV; HINTON, 2012a).

One of the major constraints regarding RBMs stands on the training step, which can be interpreted as an optimization problem where the minimization of the system's energy implies directly in an increase of the a posteriori probability of activating a hidden neuron. Such assumption lead many studies towards a more efficient manner of solving this optimization problem and approximate the output to the log-likelihood, which is considered the "perfect result", however intractable to reach when the number of variables is relatively large. Since the number of visible units generally stand for the number of pixels when dealing with image problems, the number of visible units tends to be great enough to convert such log-likelihood approximation into a prohibitive task.

Recently, many works addressed the task of modeling such log-likelihood approximation as a sampling over a Markov chain (HINTON, 2002; TIELEMAN, 2008a; TIELEMAN; HINTON, 2009; BRAKEL; DIELEMAN; SCHRAUWEN, 2012; XU; LI; ZHOU, 2014), where the initial solution, i.e., the model input, stands for some data sample, as well as the expected output stands for the corresponding sample approximation. Such process is then repeated over the training dataset until reaching some stopping criterion. Since RBM-based algorithms are a recent subject and some research groups abroad have spent an intensive effort to understand them, only a few works have been addressed towards such models in Brazil.

The hypothesis and main contributions of the present thesis regard answering the following question: which strategies could one adopt towards enhancing RBM-based models training process? Two approaches are proposed to accomplish such task: (i) the application of meta-heuristic optimization algorithms to fine-tune the hyper-parameters of such models, and (ii) the introduction of the temperature parameter into the DBM-based formulation. The experimental results, discussed in the following chapters, support the proposed hypothesis. Moreover, this thesis is composed of a collection of works published/submitted by the authors during the period of study.

The works presented in the next sections aim towards optimization of Restricted Boltzmann Machines based machine learning algorithms, i.e., Restricted Boltzmann Machines itself, Deep Belief Networks, Deep Boltzmann Machines, and infinity Restricted Boltzmann Machines. Additionally, the authors published yet three more papers related to this subject during the study process, which were not included in this thesis: (i) Learning Parameters in Deep Belief Networks Through Firefly Algorithm (ROSA et al., 2016b), (ii) Parkinson's Disease Identification Using Restricted Boltzmann Machines (PEREIRA et al., 2017), and (iii) Fine Tuning Deep Boltzmann Machines Through Meta-Heuristic Approaches (PASSOS; RODRIGUES; PAPA, 2018). The previously mentioned works employ meta-heuristic techniques for such tasks, as well as an approximation of the computational formulation to the original Boltzmann formulation, by introducing the temperature parameter in the DBM domain.

Chapter 2 presents a meticulous referential background regarding RBMs, DBNs, and DBMs, as well as the sampling methods most commonly used for training such techniques. Further, the chapter presents some procedures employed for optimization of the computational burden using CPUs and GPUs, as well as the optimization of the model's meta-parameters. Finally, it presents the most known techniques for RBM regularization, such as the employment of dropout or the temperature parameter, which has a direct relation to the activation of visible and hidden units, for instance.

The temperature meta-parameter is introduced for the very first time into the DBM formulation in the paper presented in Chapter 3. Its impact is evaluated through the learning steps, and the results are compared even with a distinct activation function, once such parameter inserted in the energy function can be interpreted as a scalar multiplication of the Sigmoid function input. The provided results confirm the hypothesis suggested by Li et al. (LI et al., 2016a) that lower temperatures tend to reach more curate results. Furthermore, one can observe that lower temperatures also support sparseness representations of the hidden layer, which leads to a dropout like regularization.

A continuation of the work presented in Chapter 3 is provided in Chapter 4. The work proposes an adaptive temperature, where it increases smoothly while the training progresses. Such approach can be compared to the behavior observed in meta-heuristic algorithms, where each agent initially explores the search space in the quest for better solutions, and later converges to the points whose results are more promising as training advances. The main contribution of the work is the exemption of the task of fine-tuning the temperature parameter, providing a friendly interface for less experienced users. Additionally, it presents results at least competitive with the ones where the temperature is fine-tuned.

The paper presented in Chapter 5 introduces the problem of DBMs meta-parameter fine-tuning aided by meta-heuristic optimization techniques. The work compares seven distinct techniques: IHS, AIWPSO, CS, FA, BSA, JADE, and CoBiDE, as well as a random search. Further, DBM's performance is compared against the DBN, outperforming the results of the latter in two out of three datasets.

Following the same idea, the work presented in Chapter 6 introduces a similar approach for meta-parameter optimization regarding oRBM and iRBM domain. The main objective of iRBM is precisely to ease the proper selection of its meta-parameters, setting automatically the number of hidden units that best fit the model. Notwithstanding, the proper selection of such parameter, i.e., the number of hidden units, is replaced by the need of fine-tuning a penalty parameter. The latter, however, is leastwise less sensitive than the proper selection of the hidden layer size, but still deserves some attention while fine-tuning the model. Finally, Chapter 7 presents a continuation of the work presented in Chapter 6, applying iRBM for Barret's Esophagus lesions detection. Finally, Chapter 8 provides the conclusions, as well as contributions of this work.

Chapter 2

Theoretical Background

This chapter presents the theoretical background regarding RBM-based models, as well as proposed methods for sampling, optimization, and regularization.

2.1 Restricted Boltzmann Machines

Invented under the name Harmonium by Paul Smolensky in 1986, (SMOLENSKY, 1986a) and renamed in the mid-2000s by Geoffrey Hinton, after invented fast learning algorithms for them, Restricted Boltzmann Machines are energy-based stochastic neural networks composed of two layers of neurons (visible and hidden), in which the learning phase is conducted by means of an unsupervised fashion. A naïve architecture of a Restricted Boltzmann Machine comprises a visible layer $\mathbf{v}$ with m units and a hidden layer $\mathbf{h}$ with n units. Additionally, a real-valued matrix $\mathbf{W}_{m\times n}$ models the weights between the visible and hidden neurons, where w_{ij} stands for the weight between the visible unit v_i and the hidden unit h_j. Figure reff.rbm depicts the RBM architecture.

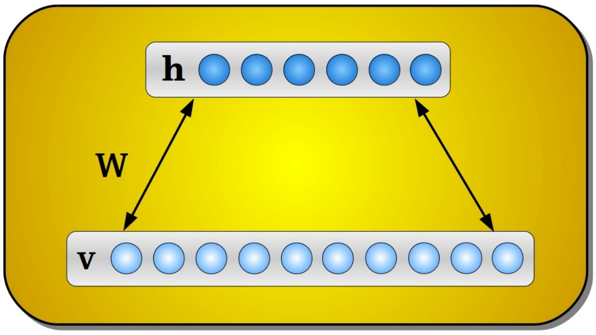


Figure 2.1: The RBM architecture.

Let us assume both $\mathbf{v}$ and $\mathbf{h}$ as being binary-valued units. In other words, $\mathbf{v} \in \{0,1\}^m$

e $\mathbf{h} \in \{0,1\}^n$. The learning process is conducted using the minimization of the system's energy, analogous to the Maxwell-Boltzmann distribution law of thermodynamics. The energy function of a Restricted Boltzmann Machine is given by:

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{n} b_j h_j - \sum_{i=1}^{m}\sum_{j=1}^{n} v_i h_j w_{ij}, \tag{2.1}$$

where $\mathbf{a}$ e $\mathbf{b}$ stand for the biases of the visible and hidden units, respectively.

The probability of a joint configuration $(\mathbf{v},\mathbf{h})$ is computed as follows:

$$P(\mathbf{v},\mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})}, \tag{2.2}$$

where Z stands for the so-called partition function, which is basically a normalization factor computed over all possible configurations involving the visible and hidden units. Similarly, the marginal probability of a visible (input) vector is given by:

$$P(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}. \tag{2.3}$$

Since the RBM is a bipartite graph, the activations of both visible and hidden units are mutually independent, thus leading to the following conditional probabilities:

$$P(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{m} P(v_i|\mathbf{h}), \tag{2.4}$$

and

$$P(\mathbf{h}|\mathbf{v}) = \prod_{j=1}^{n} P(h_j|\mathbf{v}), \tag{2.5}$$

where

$$P(v_i = 1|\mathbf{h}) = \phi\left(\sum_{j=1}^{n} w_{ij} h_j + a_i\right), \tag{2.6}$$

and

$$P(h_j = 1|\mathbf{v}) = \phi\left(\sum_{i=1}^{m} w_{ij} v_i + b_j\right). \tag{2.7}$$

Note that $\phi(\cdot)$ stands for the logistic-sigmoid function.

Let $\boldsymbol{\theta} = (W, a, b)$ be the set of parameters of an RBM, which can be learned through a training algorithm that aims at maximizing the product of probabilities given by the available training data $\mathscr{D}$, as follows:

$$\arg\max_{\Theta} \prod_{\mathbf{v}\in\mathscr{D}} P(\mathbf{v}). \tag{2.8}$$

One can solve the aforementioned equation using Contrastive Divergence, for instance, depicted on section 2.6.1.

2.2 Deep Belief Networks

Deep belief network (DBN) is a generative graphical model composed of multiple layers of latent variables ("hidden units"), with connections between the layers but not between units within each layer. In a nutshell, DBNs are composed of a set of stacked RBMs, being each of them trained using the learning algorithm presented in Section 2.1 in a greedy fashion, which means an RBM at a certain layer does not consider others during its learning procedure. In this case, we have a DBN composed of L layers, being $\mathbf{W}^i$ the weight matrix of the RBM at layer i. Additionally, we can observe the hidden units at layer i become the input units to the layer $i+1$. Figure reff.dbn depicts the model.

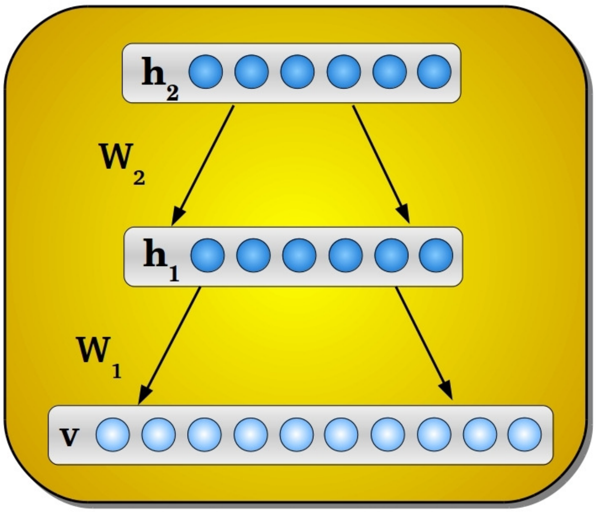


Figure 2.2: The DBN architecture.

The approach proposed by Hinton et al. (HINTON; OSINDERO; TEH, 2006a) for the training step of DBNs also considers a fine-tuning as a final step after the training of each RBM. Such procedure can be performed by means of a Backpropagation or Gradient descent algorithm, for instance, in order to adjust the matrices $\mathbf{W}^i$, $i = 1, 2, \ldots, L$. The optimization algorithm aims at minimizing some error measure considering the output of an additional layer placed on the top of the DBN after its former greedy training. Such layer is often composed of softmax or logistic units, or even some supervised pattern recognition technique.

2.3 Deep Boltzmann Machines

DBM formulation is rather similar to DBN one, with some slightly differences. Suppose we have a DBM with two layers, where $\mathbf{v}$ stand for the visible units, as well as $\mathbf{h}^1$ and $\mathbf{h}^2$ stand for the hidden units at the first and second layer, respectively. Figure 2.3 depicts the architecture of a standard DBM, which formulation is slightly different from a DBN one.

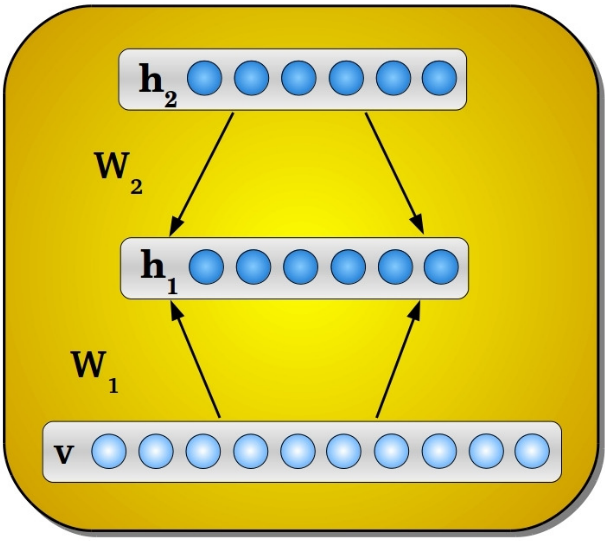


Figure 2.3: The DBM architecture with two hidden layers.

The energy of a DBM can be computed as follows:

$$E(\mathbf{v},\mathbf{h}^1,\mathbf{h}^2) = -\sum_{i=1}^{m^1}\sum_{j=1}^{n^1} v_i h_j^1 w_{ij}^1 - \sum_{i=1}^{m^2}\sum_{j=1}^{n^2} h_i^1 h_j^2 w_{ij}^2, \tag{2.9}$$

where m^1 and m^2 stand for the number of visible units in the first and second layers, respectively, and n^1 and n^2 stand for the number of hidden units in the first and second layers, respectively. In addition, we have the weight matrices $\mathbf{W}^1_{m^1\times n^1}$ and $\mathbf{W}^2_{m^2\times n^2}$, which encode the weights of the connections between vectors $\mathbf{v}$ and $\mathbf{h}^1$, and vectors $\mathbf{h}^1$ and $\mathbf{h}^2$, respectively. For the sake of simplification, we dropped the bias terms out.

The marginal probability the model assigns to a given input vector $\mathbf{v}$ is given by:

$$P(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}^1,\mathbf{h}^2} e^{-E(\mathbf{v},\mathbf{h}^1,\mathbf{h}^2)}. \tag{2.10}$$

Finally, the conditional probabilities over the visible and the two hidden units are given as follows:

$$P(v_i = 1|\mathbf{h}^1) = \phi\left(\sum_{j=1}^{n^1} w_{ij}^1 h_j^1\right), \tag{2.11}$$

$$P(h_z^2 = 1|\mathbf{h}^1) = \phi\left(\sum_{i=1}^{m^2} w_{iz}^2 h_i^1\right), \tag{2.12}$$

and

$$P(h_j^1 = 1|\mathbf{v},\mathbf{h}^2) = \phi\left(\sum_{i=1}^{m^1} w_{ij}^1 v_i + \sum_{z=1}^{n^2} w_{jz}^2 h_z^2\right). \tag{2.13}$$

After learning the first RBM using Contrastive Divergence 2.6.1, for instance, the generative model can be written as follows:

$$P(\mathbf{v}) = \sum_{\mathbf{h}^1} P(\mathbf{h}^1)P(\mathbf{v}|\mathbf{h}^1), \tag{2.14}$$

where $P(\mathbf{h}^1) = \sum_{\mathbf{v}} P(\mathbf{h}^1,\mathbf{v})$. Further, we shall proceed with the learning process of the second RBM, which then replaces $P(\mathbf{h}^1)$ by $P(\mathbf{h}^1) = \sum_{\mathbf{h}^2} P(\mathbf{h}^1,\mathbf{h}^2)$. In short, using such procedure, the conditional probabilities given by Equations 2.11-2.13, and Contrastive Divergence, one can learn DBM parameters one layer at a time (SALAKHUTDINOV; HINTON, 2012b).

2.4 Ordered Restricted Boltzmann Machines

The ordered Restricted Boltzmann Machine is a variant of the RBM such that the hidden units are trained sequentially, from the left to the right. The current number of trained units at a given time step is represented by the variable $z \leq n$, as depicted in Figure 2.4.

Given a number z of hidden units, one can compute the energy of the current model as follows:

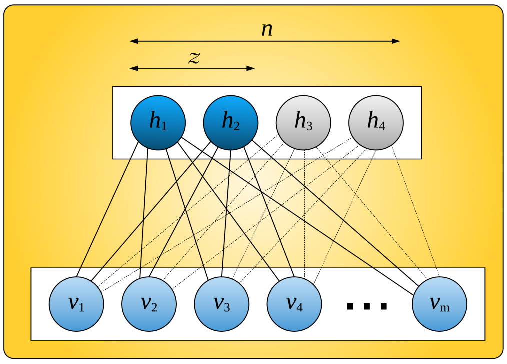


Figure 2.4: An oRBM with $z = 2$ and $n = 4$.

$$\mathbf{E}(\mathbf{v},\mathbf{h},z) = -\sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{z} b_j h_j - \sum_{i=1}^{m}\sum_{j=1}^{z}\left(v_i h_j w_{ij} - \beta_j\right), \tag{2.15}$$

where β_j represents the energy penalty associated to the hidden unit h_j. Actually, β_j forces the model to avoid using more hidden units than needed, thus generating smaller networks.

Therefore, the joint probability over $\mathbf{v}$, $\mathbf{h}$ and z is given as follows:

$$P(\mathbf{v},\mathbf{h},z) = \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h},z)}. \tag{2.16}$$

and the marginal probability is given by:

$$P(\mathbf{v}) = \frac{1}{Z}\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h},z)}. \tag{2.17}$$

Similarly to the RBM, since Z is intractable in the above equation, the probabilities over $\mathbf{v}$ and $\mathbf{h}$ are estimated by means of Gibbs sampling:

$$P(h_j = 1|\mathbf{v},z) \quad = \quad \begin{cases} \phi\left(\sum_{i=1}^{m} w_{ij} v_i + b_j\right) & \text{if } j \leq z \\ 0 & \text{otherwise,} \end{cases} \tag{2.18}$$

and

$$P(v_i = 1|\mathbf{h},z) = \phi\left(\sum_{j=1}^{z} w_{ij} h_j + a_i\right). \tag{2.19}$$

However, oRBM has an additional information that concerns the maximum number of hidden units that is going to be used, i.e., variable z. Given an input data $\mathbf{v}$, the conditional distribution over the value of z is given as follows:

$$P(z|\mathbf{v}) = \frac{\exp(-F(\mathbf{v},z))}{\sum_{z'=1}^{n}\exp(-F(\mathbf{v},z'))}, \tag{2.20}$$

where $F(\mathbf{v},z)$ is the so-called "free energy", being computed as follows:

$$F(\mathbf{v},z) = -\sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{z}\left(\psi\left(\sum_{i=1}^{m} w_{ij}v_i + b_j\right) - \beta_j\right), \tag{2.21}$$

where $\psi(x) = \ln(1+e^x)$.

Equation 2.20 tells us we need to consider sampling z from the Markov chain as well. In this case, Gibbs steps alternate between sampling $(h,z) \sim P(\mathbf{h},z|\mathbf{v})$ and $\mathbf{v} \sim P(\mathbf{v}|\mathbf{h},z)$. Notice the sampling from $P(\mathbf{h},z|\mathbf{v})$ can be performed in two steps: $z \sim P(z|\mathbf{v})$ followed by $\mathbf{h} \sim P(\mathbf{h}|\mathbf{v},z)$.

Finally, the weight matrix $\mathbf{W}$ and the biases $\mathbf{a}$ and $\mathbf{b}$ in the oRBM model are than updated by the following equations:

$$\mathbf{W}^{t+1} = \mathbf{W}^{t} + \eta(\boldsymbol{\xi}\mathbf{v}^T - \tilde{\boldsymbol{\xi}}\tilde{\mathbf{v}}^T), \tag{2.22}$$

where $\boldsymbol{\xi} = P(\mathbf{h}|\mathbf{v}) \odot (1-\rho(z|\mathbf{v}))$ and $\tilde{\boldsymbol{\xi}} = P(\tilde{\mathbf{h}}|\tilde{\mathbf{v}}) \odot (1-\rho(z|\tilde{\mathbf{v}}))$. Notice the operator $\odot$ stands for the element-wise product, and $\rho(z|\mathbf{v}) = [P(z<1|\mathbf{v}), P(z<2|\mathbf{v}), \ldots, P(z<n|\mathbf{v})]^T$.

The biases can be updated as follows:

$$\mathbf{a}^{t+1} = \mathbf{a}^{t} + \eta(\mathbf{v} - \tilde{\mathbf{v}}) \tag{2.23}$$

and

$$\mathbf{b}^{t+1} = \mathbf{b}^{t} + \eta(\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}}), \tag{2.24}$$

where $\boldsymbol{\lambda} = (P(\mathbf{h}|\mathbf{v}) - \boldsymbol{\beta}\boldsymbol{\phi}(\mathbf{b})) \odot (1-\rho(z|\mathbf{v}))$ and $\tilde{\boldsymbol{\lambda}} = (P(\tilde{\mathbf{h}}|\tilde{\mathbf{v}}) - \boldsymbol{\beta}\boldsymbol{\phi}(\mathbf{b})) \odot (1-\rho(z|\tilde{\mathbf{v}}))$. Notice that $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_z]$, and $\boldsymbol{\phi}$ is the same sigmoid-logistic function as before, but now applied to the array $\mathbf{b}$.

In short, the rationale of oRBMs is to perform the training step adding one hidden unit at time, from the left to the right. Since $P(z|\mathbf{v})$ usually increases according to greater values of z (i.e., we have more complex models), the term $(1-\rho(z|\mathbf{v}))$ decreases monotonically from the left to the right, thus forcing the model using less hidden units (i.e., smaller values of z).

2.5 Infinity Restricted Boltzmann Machines

The infinity RBM mimics the same growing behavior of the oRBM, but the maximum number of hidden units is not specified. This number increases automatically until its capacity is sufficiently high, which is possible by taking the limit of $n \rightarrow \infty$. The model is presented in Figure 2.5.

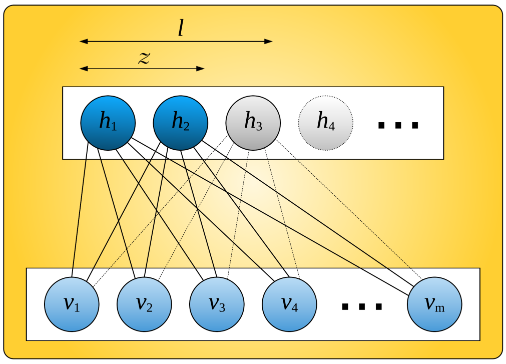


Figure 2.5: An iRBM trained previously with $z=2$ units. There are some non-zero (dashed lines) values connecting the third unit ($l=3$) that is going to be used for training. All remaining hidden units (i.e., $l>3$) have zero-valued weights.

The updating equations concerning iRBM are given as follows:

$$\mathbf{W}^{t+1} = \mathbf{W}^{t} + \eta(P(\mathbf{h}|\mathbf{v},z)\mathbf{v}^{T} - P(\tilde{\mathbf{h}}|\tilde{\mathbf{v}},\tilde{z})\tilde{\mathbf{v}}^{T}), \tag{2.25}$$

$$\mathbf{a}^{t+1} = \mathbf{a}^{t} + \eta(\mathbf{v} - \tilde{\mathbf{v}}) \tag{2.26}$$

and

$$\mathbf{b}^{t+1} = \mathbf{b}^{t} + \eta(\boldsymbol{\alpha} - \tilde{\boldsymbol{\alpha}}), \tag{2.27}$$

where $\boldsymbol{\alpha} = (P(\mathbf{h}|\mathbf{v}) - \boldsymbol{\beta}\boldsymbol{\phi}(\mathbf{b}))\odot\boldsymbol{I}_z$ and $\tilde{\boldsymbol{\alpha}} = (P(\tilde{\mathbf{h}}|\tilde{\mathbf{v}}) - \boldsymbol{\beta}\boldsymbol{\phi}(\mathbf{b}))\odot\boldsymbol{I}_z$, and $\boldsymbol{I}_z = [\underbrace{1,\ldots,1}_{z},\underbrace{0,\ldots,0}_{n-z}]^T$.

2.6 Sampling Methods

Initially, the strategy adopted to estimate $E[\mathbf{hv}]^{model}$, which is the representation of the data learned by the system, was basically to start the visible units with random values and run alternating Gibbs chain until equilibrium, (i.e., convergence). However, this approach is computationally expensive, since a good model is obtained when the number of Gibbs steps $k \rightarrow \infty$. Figure 2.6 depicts the model.

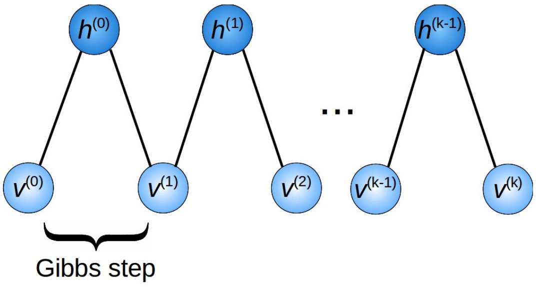


Figure 2.6: Gibbs sampling

To tackle the aforementioned problem, some alternatives to Gibbs sampling were presented in the following years. The next sections discuss some of the most used techniques for such purpose.

2.6.1 Contrastive Divergence

Hinton (HINTON, 2002) introduced a faster methodology to compute $E[\mathbf{hv}]^{model}$ based on contrastive divergence. Basically, the idea is to initialize the visible units with a training sample, to compute the states of the hidden units using Equation 2.7, and then to compute the states of the visible unit (reconstruction step) using Equation 2.6. In short, this is equivalent to perform Gibbs sampling using $k = 1$ and initializing the chain with the the training samples.

Based on the above assumption, we can now compute $E[\mathbf{hv}]^{model}$ as follows:

$$E[\mathbf{hv}]^{model} = P(\tilde{\mathbf{h}}|\tilde{\mathbf{v}})\tilde{\mathbf{v}}^T, \tag{2.28}$$

where $\tilde{\mathbf{v}}$ stands for the reconstruction of the visible layer given $\mathbf{h}$, and $\tilde{\mathbf{h}}$ denotes a estimation of the hidden vector $\mathbf{h}$ given $\tilde{\mathbf{v}}$.

Therefore, the equation below leads to a simple learning rule for updating the weight matrix $\mathbf{W}$, as follows:

$$\begin{aligned}\mathbf{W}^{t+1} &= \mathbf{W}^{t} + \eta(E[\mathbf{hv}]^{data} - E[\mathbf{hv}]^{model}) \\ &= \mathbf{W}^{t} + \eta(P(\mathbf{h}|\mathbf{v})\mathbf{v}^{T} - P(\tilde{\mathbf{h}}|\tilde{\mathbf{v}})\tilde{\mathbf{v}}^{T}), \end{aligned} \tag{2.29}$$

where $\mathbf{W}^t$ stands for the weight matrix at time step t, and η corresponds to the learning rate. Additionally, we have the following formulae to update the biases of the visible and hidden units:

$$\begin{aligned}\mathbf{a}^{t+1} &= \mathbf{a}^{t} + \eta(\mathbf{v} - E[\mathbf{v}]^{model}) \\ &= \mathbf{a}^{t} + \eta(\mathbf{v} - \tilde{\mathbf{v}}), \end{aligned} \tag{2.30}$$

and

$$\begin{aligned}\mathbf{b}^{t+1} &= \mathbf{b}^{t} + \eta(E[\mathbf{h}]^{data} - E[\mathbf{h}]^{model}) \\ &= \mathbf{b}^{t} + \eta(P(\mathbf{h}|\mathbf{v}) - P(\tilde{\mathbf{h}}|\tilde{\mathbf{v}})), \end{aligned} \tag{2.31}$$

where $\mathbf{a}^t$ and $\mathbf{b}^t$ stand for the visible and hidden units biases at time step t, respectively. In short, Equations 2.29, 2.30 and 2.31 are the standard formulation for updating the RBM parameters.

Later on, Hinton (HINTON, 2012) introduced a weight decay parameter λ, which penalizes weights with large magnitude, as well as a momentum parameter α to control possible oscillations during the learning process. Therefore, we can rewrite Equations 2.29, 2.30 and 2.31 as follows:

$$\mathbf{W}^{t+1} = \mathbf{W}^{t} + \underbrace{\eta(P(\mathbf{h}|\mathbf{v})\mathbf{v}^{T} - P(\tilde{\mathbf{h}}|\tilde{\mathbf{v}})\tilde{\mathbf{v}}^{T}) - \lambda\mathbf{W}^{t} + \alpha\Delta\mathbf{W}^{t-1}}_{=\Delta\mathbf{W}^{t}}, \tag{2.32}$$

$$\mathbf{a}^{t+1} = \mathbf{a}^{t} + \underbrace{\eta(\mathbf{v} - \tilde{\mathbf{v}}) + \alpha\Delta\mathbf{a}^{t-1}}_{=\Delta\mathbf{a}^{t}} \tag{2.33}$$

and

$$\mathbf{b}^{t+1} = \mathbf{b}^{t} + \underbrace{\eta(P(\mathbf{h}|\mathbf{v}) - P(\tilde{\mathbf{h}}|\tilde{\mathbf{v}})) + \alpha\Delta\mathbf{b}^{t-1}}_{=\Delta\mathbf{b}^{t}}. \tag{2.34}$$

2.6.2 Persistent Contrastive Divergence

Most of the issues related to contrastive divergence approach are related to the number of iterations employed to approximate the model to the real data. Although the approach proposed by Hinton (HINTON, 2002) takes $k = 1$ and works well for real world problems, one can settle different values for k (CARREIRA-PERPIÑÁN; HINTON, 2005)[1].

Notwithstanding contrastive divergence provides a good approximation to the likelihood gradient, i.e., it provides a good approximation of the model to the data when $k \to \infty$. However, its convergence might becomes poor when the Markov chain has a "low mixing". Furthermore, contrastive divergence has a good convergence only on the early iterations, getting slower as iterations go by, thus, demanding the use of parameters decay (as shown in equations 2.32, 2.33 and 2.34, for instance).

Therefore, an interesting alternative for contrastive divergence would be using higher values for k, usually named CD-k. However, a major problem related to this approach dues to its computational burden, since a greater number of iterations are required to approximate the model to the data. Given such premise, Tieleman (TIELEMAN, 2008a) proposed the Persistent Contrastive Divergence - PCD for short - which aims to approximate the model to the data similarly to CD-k, but with a lower computational burden. The idea is quite simple: on CD-1, each training sample is employed to start an RBM and rebuilds a model after a single Gibbs sampling iteration. Once every training sample is presented to the RBM, we have a so-called "epoch". The process is repeated for each next epoch, i.e., the same training samples are used to feed the RBM and the Markov chain is restarted at each epoch. PCD aims to achieve an "ideal" approximation of the model to the data given CD-k (when $k \to \infty$) by means of not restarting the Markov chain, but using the model built in the former epoch to feed the RBM in the current epoch. Therefore, as the number of epochs increases, the model tends to be similar to the one obtained through CD-k. The only problem related to this technique concerns the number of epochs demanded for convergence, but yet the reconstruction error rate is generally still lower than CD.

2.7 Regularization

This section presents some approaches applied to regularize RBM-based models.

[1] Usually, contrastive divergence with a single iteration is called CD-1.

2.7.1 Sparce Representation

It was demonstrated by Ranzinato et. al.(BOUREAU; CUN et al., 2008) that unsupervised methods based on reconstructing the input data from its representation, such as Restricted Boltzmann Machines, has a better performance if this representation is composed by a sparse vector. The authors proposed the Sparse Encoding Symmetric Machine (SESM) , which has an architecture similar to the DBN, and compared both theoretically and experimentally. Meanwhile, Lee et al. (LEE; EKANADHAM; NG, 2008) employed a sparse two-layered DBN to measure the degree to which it faithfully mimics biological measurements of V2 visual cortex area, who has as input the response properties of neurons in cortical areas receiving projections from V1 area. It was stated that the second layer captures a variety of both collinear ("contour") features as well as corners and junctions. Furthermore, it provided similar responses along several dimensions in quantitative comparison to measurements of V2 taken by (ITO; KOMATSU, 2004).

Since then, many works emerged in literature taking advantage of this property, such as Swersky et al. (SWERSKY et al., 2012), that used a cardinality potential to control the sparsity of the RBM, i.e., limiting the number of hidden units that can be active, as well as the well known Dropout (SRIVASTAVA et al., 2014a), temperature based approaches (LI et al., 2016a) (PASSOS; PAPA, 2017c) and the Infinite RBM (CÔTÉ; LAROCHELLE, 2016).

2.7.2 Dropout

Dropout is a technique for addressing the problem of over-fitting in large networks (SRIVASTAVA et al., 2014a). The main idea behind the concept is to temporarily remove a hidden or visible unit from the network, along with all its incoming and outgoing connections, as shown in Figure 2.7. The units to drop are chosen randomly and removed with a fixed probability p, commonly set to 0.5. This procedure is capable of dim the noise resulted by sampling from limited training data on complicated relationships present on RBMs and other networks. Experimentally, applications on vision, speech recognition, document classification and computational biology obtained state-of-the-art results on many benchmark data sets.

2.7.3 Temperature-based Models

Recently Li et. al. (LI et al., 2016a) introduced the "temperature"term, which was a key factor of the Boltzmann distribution, in an RBM formulation. They revealed that tempe-

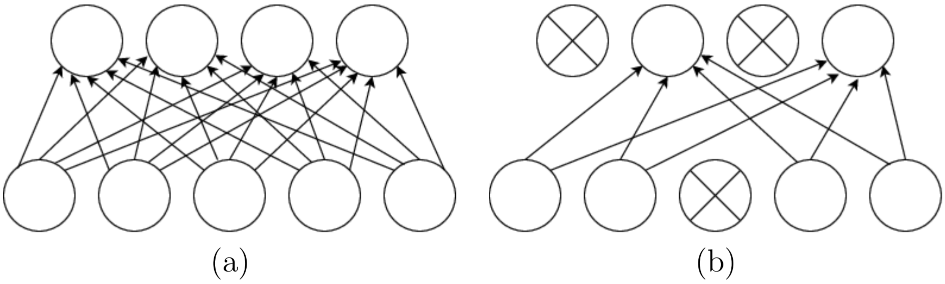


Figure 2.7: Examples of: (a) standard and (b) dropped-out RBM

rature controls the selectivity of the firing neurons in the hidden layers and theoretically proved that the effect of temperature can be adjusted by setting the parameter of sharpness on the logistic function, proposing the Temperature-based RBMs (TRBM). They evinced that the performance of RBMs can be improved by adjusting the temperature parameter of TRBMs. Passos and Papa (PASSOS; PAPA, 2017c) (PASSOS; COSTA; PAPA, 2017) adapted this idea to the DBM domain, and noticed that at lower temperatures, the input information is limited and produces even more sparse representations, which is an effect similar to the ones achieved through dropout-based approaches.

2.8 Parameter Optimization

The task of fine-tuning parameters in machine learning aims at finding suitable parameters values that maximize some fitness function, such as the classifier's recognition accuracy or the reconstruction error. In the subject, Papa et al. (PAPA et al., 2015a) employed the Harmony Search (HS) to optimize the parameters of Restricted Boltzmann Machines in the context of binary image reconstruction. In the very same year, Papa et al. (PAPA et al., 2015c) evaluated some meta-heuristic techniques to fine-tune Discriminative Restricted Boltzmann Machines, and finally employed Harmony Search and some of its variants to optimize Deep Belief Networks (PAPA; SCHEIRER; COX, 2016a).

Recently, Papa et al. (PAPA et al., 2016) and Rosa et al. (ROSA et al., 2016b) employed quaternion algebra and the Firefly Algorithm for RBM and DBN parameter fine-tuning, respectively. Moreover, Passos et al. (JUNIOR; PAPA, 2016) compared several meta-heuristic techniques, such as Particle Swarm Optimization (PSO) and its variations, and the Harmonic Search (HS) and some variations for the task of Deep Boltzmann Machine parameters and Infinity Restricted Boltzmann Machines (PASSOS; PAPA, 2017a) meta-parameter fine-tuning.

2.8.1 Metaheuristic Optimization Techniques

Below, we present a brief description of the metaheuristic techniques employed in this work:

- Brain Storm Optimization (BSO) (SHI, 2011): the Brain Storm Optimization is a meta-heuristic optimization technique inspired by the human behavior, whose motivation is the brainstorming process performed by human beings to find solutions and solve problems. The process can be divided into three main steps: (i) similar solutions are clustered together, (ii) new solution is generated, and finally (iii) the best solutions are selected.

- Harmony Search (HS) (GEEM, 2009): found inspiration in the creative process of musicians while improvising. The idea is to start a song with a set of initial harmonies and search in the memory for the harmonies that best fits the melody.

- Improved Harmony Search (IHS) (MAHDAVI; FESANGHARY; DAMANGIR, 2007): a variant of the HS, which models the problem of function minimization based on way musicians create their songs with optimal harmonies. This approach uses dynamic values for both the Harmony Memory Considering Rate (HMRC), which is responsible for creating new solutions based on previous experience of the music player, and the Pitch Adjusting Rate (PAR), which is in charge of applying some disruption to the solution created with HMRC in order to avoid the pitfalls of local optima. Both parameters are updated at each iteration with the new values within the range [HMCR$_{min}$,HMCR$_{max}$] e [PAR$_{min}$,PAR$_{max}$], respectively. Concerning PAR calculation, the bandwidth variable *(bandwidth)* ρ is used, and its values must be between $[\rho_{min}, \rho_{max}]$.

- Particle Swarm Optimization (PSO) (RODRIGUES et al., 2015): Any possible solution is represented as a particle (agent) in a swarm. Each agent has a position that represents a parameter value and velocity vector in the search space. A fitness value is associated with each position, and after some iterations, the global best position is selected as the best solution to the problem.

- Adaptive Inertia Weight Particle Swarm Optimization (AIWPSO)(AIWPSO) (YU; LIU; LI, 2009): a variant of the PSO, which considers any possible solution as a particle (agent) in a swarm. Each agent has a position that represents a parameter value and velocity vector in the search space. A fitness value is associated with each position, and after some iterations, the global best position is selected as the best solution to the problem. The AIWPSO is proposed to balance the global exploration and local exploitation abilities for PSO. For each iteration, every particle chooses an appropriate inertia weight along the search space by dynamically adjusting the inertia weight.

- Bat Algorithm (BA) (YANG; GANDOMI, 2012): is a nature-inspired metaheuristic optimization algorithm based on the echolocation behavior of bats. Each bat flies with a random velocity, position, and frequency. Additionally, they can vary the wavelength and loudness to search for prey/food (best solutions), adjusting the rate of pulses depending on the proximity of their target.

- Cuckoo Search (CS) (YANG; DEB, 2010): Cuckoo Search (YANG; DEB, 2009, 2010) employs a combination of the Lévy flight, which may be defined as a bird flight-inspired random walk over a Markov chain, together with a parasitic behavior of some cuckoo species. The model follows three basic ideas: i) each cuckoo lays one egg at a time in randomly chosen nests, ii) the host bird discover the cuckoo's egg with a probability $p_a \in [0,1]$ and either discard the egg or abandon the chest and build a new one (a new solution is created), and iii) the nests with best eggs will carry over to the next generations.

- Firefly Algorithm (FA) (YANG, 2010): is derived from the fireflies' flash attractiveness when mating partners and attracting potential preys. The attractiveness of a firefly is computed by its position related to other fireflies in the swarm, as well as its brightness is determined by the value of the objective function at that position. Furthermore, the attractiveness depends on each firefly light absorption coefficient γ. In order to avoid local optima, the system is exposed to a random perturbation α, and the best firefly performs a random walk across the search space.

- Backtracking Search Optimization Algorithm (BSA) (CIVICIOGLU, 2013): it is a simple, effective and fast evolutionary algorithm developed to deal with problems characterized by slow computation and excessive sensitivity to control parameters. In a nutshell, it employs crossover and mutation operations together with a random selection of stored memories to generate a new population of individuals based on past experiences. BSA requires a proper selection of two parameters: the mixing rate (mix_rate), which controls the number of elements of individuals that will mutate in the population, as well as the F parameter, which controls the amplitude of the search-direction matrix.

Adaptive Differential Evolution (JADE) (ZHANG; SANDERSON, 2009): a differential evolution-based algorithm that implements the "DE/current-to-*p*-best"mutation strategy, which employs only the *p*-best agents in the mutation process. Additionally, JADE uses an optional archive for historical information, as well as an adaptive updating in the control parameter. JADE requires the selection of the parameter c, which stands for the rate of parameter adaptation, and g (greediness), that determines the greediness of the mutation strategy.

- Differential Evolution Based on Covariance Matrix Learning and Bimodal Distribution Parameter Setting Algorithm (CoBiDE) (WANG et al., 2014): it also a differential evolution-based technique that employs a covariance matrix for a better representation of the system's coordinates during the crossover process. Additionally, mutation and crossover are controlled using a bimodal distribution to achieve a good trade-off between exploration and exploitation. The probability of executing the differential evolution according to the covariance matrix is defined by the parameter pb, as well as the proportion of individuals chosen from the current population to calculate the covariance matrix is denoted by ps.

2.9 Datasets

This section presents a brief description of the datasets imployed in this work.

- MNIST dataset[2]: it is composed of images of handwritten digits. The original version contains a training set with $60,000$ images from digits '0'-'9', as well as a test set with $10,000$ images[3]. Due to the high computational burden for DBM model selection, we decided to employ the original test set together with a reduced version of the training set[4].
- CalTech 101 Silhouettes Data Set[5]: it is based on the former Caltech 101 dataset, and it comprises $9,146$ silhouettes of images split between 101 classes with resolution of 28×28. We have used only the training and test sets, since our optimization model aims at minimizing the mean squared error (MSE) over the training set.

[2] http://yann.lecun.com/exdb/mnist/

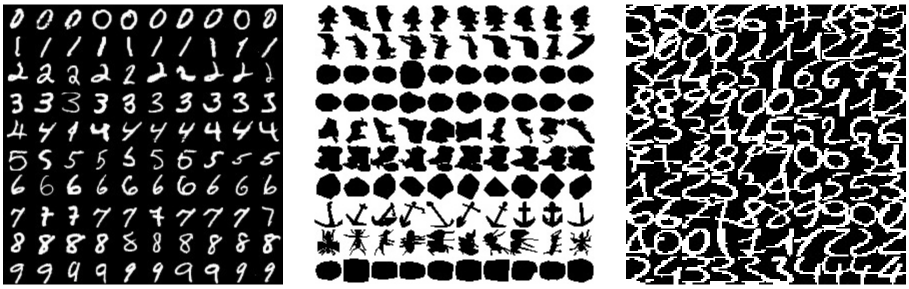
[3] The images are originally available in gray scale with resolution of 28×28, but they were reduced to 14×14 images.

[4] The original training set was reduced to 2% of its former size, which corresponds to $1,200$ images.

[5] https://people.cs.umass.edu/~marlin/data.shtml

- Semeion Handwritten Digit Dataset[6]: it is formed by $1{,}593$ images from handwritten digits '0' - '9' written in two ways: the first time in a normal way (accurately) and the second time in a fast way (no accuracy). In the end, they were stretched with resolution of 16×16 in a gray scale of 256 values and then each pixel was binarized.

Figure 2.8 displays some training examples from the above datasets.



(a) (b) (c)

Figure 2.8: Some training examples from (a) MNIST, (b) CalTech 101 Silhouettes and (c) Semeion datasets.

[6] `https://archive.ics.uci.edu/ml/datasets/Semeion+Handwritten+Digit`

Chapter 3

Temperature-Based Deep Boltzmann Machines

This chapter presents the content published in the journal Neural Processing Letters (PASSOS; PAPA, 2017c), and it proposes the introduction of the temperature parameter T into DBMs formulation.

3.1 Introduction

Deep learning techniques have attracted considerable attention in the last years due to their outstanding results in a number of applications (GOH et al., 2012; DUONG et al., 2015; SOHN; LEE; YAN, 2015), since such techniques possess an intrinsic ability to learn different information at each level of a hierarchy of layers (LECUN; BENGIO; HINTON, 2015). Restricted Boltzmann Machines (HINTON, 2012), for instance, are among the most pursued techniques, even though they are not deep learning-oriented themselves, but by building blocks composed of stacked RBMs on top of each other one can obtain the so-called Deep Belief Networks (HINTON; OSINDERO; TEH, 2006a) or the Deep Boltzmann Machines (SALAKHUTDINOV; HINTON, 2012b), which basically differ from each other by the way the inner layers interact among themselves.

The Restricted Boltzmann Machine is a probabilistic model that uses a layer of hidden units to model the distribution over a set of inputs, thus compounding a generative stochastic neural network (LAROCHELLE et al., 2012; SCHMIDHUBER, 2015). RBMs were firstly idealized under the name of "Harmonium" by Smolensky in 1986 (SMOLENSKY, 1986a), and some years later renamed to RBM by Hinton et. al. (HINTON, 2002). Since then, the scientific community has been putting a lot of effort in order to improve the results in a number of application that somehow make use of RBM-based models (HINTON; SALAKHUTDINOV, 2006, 2011; PAPA et al., 2015b, 2015d; PAPA; SCHEIRER; COX, 2016a; TOMCZAK; GONCZAREK, 2016).

In a nutshell, the key role in RBMs concerns their learning parameter step, which is usually carried out by sampling in Markov chains in order to approximate the gradient of the logarithm of the likelihood concerning the estimated data with respect to the input one. In this context, Li et. al. (LI et al., 2016a) recently highlighted the importance of a crucial concept in Boltzmann-related distributions: their "temperature", which has a main role in the field of statistical mechanics (MENDES et al., 2015), (Beraldo e Silva et al., 2014), (GADJIEV; PROGULOVA, 2015), idealized by Wolfgang Boltzmann. In fact, a Maxwell-Boltzmann distribution (GORDON, 2002; SHIM; GATIGNOL, 2010; NIVEN, 2005) is a probability distribution of particles over various possible energy states without interacting with one another, expect for some very brief collisions, where they exchange energy. Li et. al. (LI et al., 2016a) demonstrated the temperature influences on the way RBMs fire neurons, as well as they showed its analogy to the state of particles in a physical system, where a lower temperature leads to a lower particle activity, but higher entropy (BEKENSTEIN, 1973), (RRNYI, 1961).

Since DBMs are a natural extension of RBMs and DBNs, we believe the temperature can also play an important role in these models. In short, the main core of DBMs still relies on the RBM formulation, which uses the temperature to approximate the distribution of the data during learning step. Therefore, this work aims at evaluating whether our suspicion that temperature influences DBMs holds or not. However, as far we are concerned, the impact of different temperatures during the Markov sampling has never been considered in Deep Boltzmann Machines. Therefore, the main contributions of this work are three fold: (i) to foster the scientific literature regarding DBMs, (ii) to evaluate the impact of temperature during DBM learning phase, and (iii) to evaluate a different Sigmoid function in the context of DBNs and DBMs. Since the temperature parameter in the energy formulation can be interpreted as a multiplication of the Sigmoid function by a scalar number, we also considered the Gompertz curve as an activation function (GOMPERTZ, 1825), given that its parameters allow one to map the outputs within $[0, 1]$, just as the regular sigmoid function. Also, we considered Deep Belief Networks for comparison purposes concerning the task of binary image reconstruction over three public datasets.

The remainder of this chapter is organized as follows. Section 3.2 presents the theoretical background related to the proposed temperature-based approach, and Section 3.3

describes the methodology adopted in this work. The experimental results are discussed in Section 3.4, and conclusions and future works are stated in Section 3.5.

3.2 Theoretical Background

In this section, we briefly explain the theoretical background related to the proposed approaches.

3.2.1 Temperature-based Deep Boltzmann Machines

Li et. al. (LI et al., 2016a) showed that a temperature parameter T controls the sharpness of the logistic-sigmoid function. In order to incorporate the temperature effect into the RBM context, they introduced this parameter to the joint distribution of the vectors $\mathbf{v}$ and $\mathbf{h}$ in Equation 2.2, which can be rewritten as follows:

$$P(\mathbf{v},\mathbf{h};T) = \frac{1}{Z}e^{\frac{-E(\mathbf{v},\mathbf{h})}{T}}. \tag{3.1}$$

As such, when $T = 1$ the aforementioned equation degenerates to Equation 2.2. Therefore, the probability of a given sample $\mathbf{v}$ given by Equation 2.10 can be rewritten considering now the temperature:

$$P(\mathbf{v};T) = \frac{1}{Z}\sum_{\mathbf{h}^1,\mathbf{h}^2} e^{\frac{-E(\mathbf{v},\mathbf{h}^1,\mathbf{h}^2)}{T}}. \tag{3.2}$$

In addition, Equation 2.7 can be rewritten in order to accommodate the temperature parameter as follows:

$$P(h_j = 1|\mathbf{v}) = \phi\left(\frac{\sum_{i=1}^{m} w_{ij}v_i}{T}\right). \tag{3.3}$$

Notice the temperature parameter does not affect the conditional probability of the input units (Equation 2.6).

In order to apply the very same idea to DBMs, the conditional probabilities over the two hidden layers given by Equations 2.12 and 2.13 can be derived and expressed using the following formulation, respectively:

$$P(h_z^2 = 1|\mathbf{h}^1) = \phi\left(\frac{\sum_{i=1}^{m^2} w_{iz}^2 h_i^1}{T}\right), \tag{3.4}$$

and

$$P(h_j^1 = 1|\mathbf{v}, \mathbf{h}^2) = \phi\left(\frac{\sum_{i=1}^{m^1} w_{ij}^1 v_i + \sum_{z=1}^{n^2} w_{jz}^2 h_z^2}{T}\right). \tag{3.5}$$

3.2.2 Gompertz Function

The Gompertz function is a generalization of the well-known logistic function, where its growth is slowest at the beginning and at the end, and it gradually increases according to a given parameter. Such behavior can not be observed in the standard logistic function, in which both sides are approached by the curve symmetrically. The Gompertz function can be formulated as follows:

$$f(t) = ae^{-be^{-ct}}, \tag{3.6}$$

where a controls the bounds of the function such that $f(t) \in [0, a]$, b and c are positive numbers such that b sets the displacement along the x-axis (translates the graph to the left or right) and c sets the growth rate (y scaling). Finally, t stands for a time step. Figure 3.1 depicts the behavior of the function concerning its parameters.

3.3 Methodology

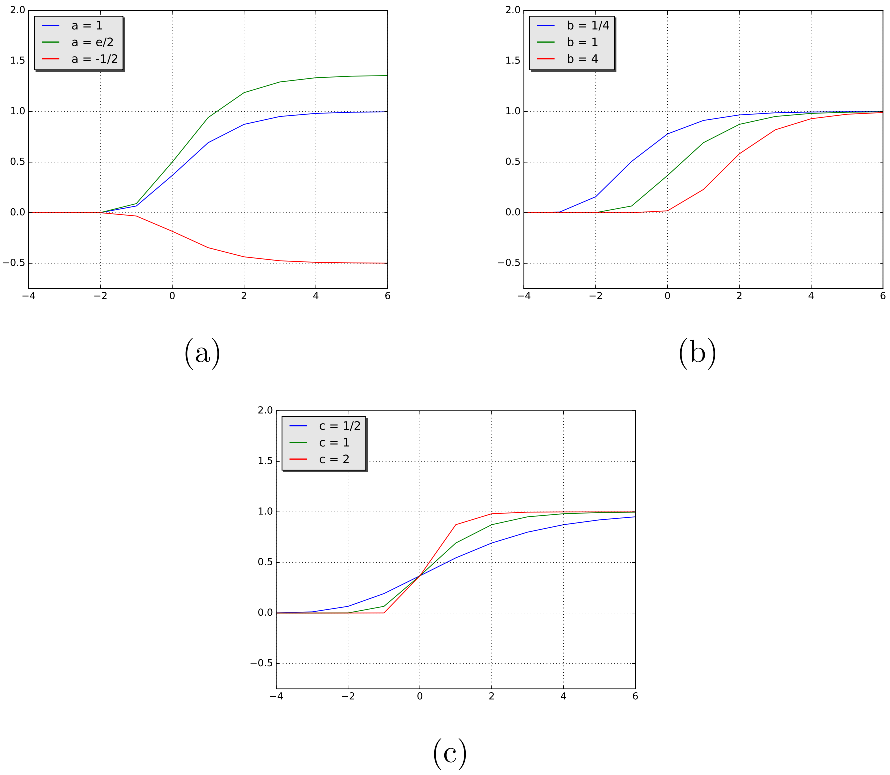
In this section, we present the methodology employed to evaluate the proposed approach, as well the datasets and the experimental setup.

3.3.1 Datasets

We propose to evaluate the behavior of DBMs under different temperatures in the context of binary image reconstruction using three public datasets, i.e., MNIST, CalTech 101 Silhouettes Data Set, and Semeion Handwritten Digit Data Set, presented in Section 2.9.

3.3.2 Experimental Setup

We employed a 3-layered architecture for all datasets as follows: i-500-500-2,000, where i stands for the number of pixels used as input for each dataset, i.e., 196 (14×14 images), 784 (28×28 images) and 256 (16×16 images) considering MNIST, Caltech 101



(a)

(b)

(c)

Figure 3.1: Gompertz curves variating theirs parameters for a, b and c, respectively. Note the remaining parameters are fixed to 1.

Silhouettes and Semeion Handwritten Digit datasets, respectively. Therefore, we have a first and a second hidden layers with 500 neurons each, followed by a third hidden layer with 2,000 neurons[1]. The remaining parameters used during the learning steps were fixed for each layer as follows: $\eta = 0.1$ (learning rate), $\lambda = 0.1$ (weight decay), $\alpha = 0.00001$ (penalty parameter). In addition, we compared DBMs against DBNs using the very same configuration, i.e., architecture and parameters[2].

In order to provide a statistical analysis by means of the Wilcoxon signed-rank test with significance of 0.05 (WILCOXON, 1945), we conducted a cross-validation procedure with 20 runnings. In regard to the temperature, we considered a set of values within the range $T \in \{0.1, 0.2, 0.5, 0.8, 1.0, 1.2, 1.5, 2.0\}$ for the sake of comparison purposes. Additionally, we employed the Gompertz curve for both regular (i.e., $T = 1.0$) DBN and DBM, being its parameters b and c fine-tuned by means of the well-known Particle Swarm Optimization (PSO) (KENNEDY, 2011). Since a controls the bounds of the function, and we are dealing with a binary reconstruction problem, we set $a = 1$.

[1] Similar architectures have been commonly employed in the literature (LI et al., 2016a), (HINTON; OSINDERO; TEH, 2006a), (SALAKHUTDINOV; HINTON, 2009b), (WICHT; FISCHER; HENNEBERT, 2016) and (SALAKHUTDINOV; HINTON, 2009a).

[2] Notice all parameters and architectures have been empirically chosen (PAPA; SCHEIRER; COX, 2016a).

Finally, we employed 10 epochs for DBM and DBN learning weights procedure with mini-batches of size 20. In order to provide a more precise experimental validation, we trained both DBMs and DBNs with two different algorithms[3][4]: Contrastive Divergence (CD) (HINTON, 2002) and Persistent Contrastive Divergence (PCD) (TIELEMAN, 2008a).

3.4 Experimental Results

In this section, we present the experimental results concerning the proposed temperature-based Deep Boltzmann Machine over three public datasets aiming at the task of binary image reconstruction. Table 3.1 presents the results considering Semeion Handwritten Digit dataset, in which the values in bold stand for the most accurate ones by means of the Wilcoxon signed-rank test. Notice we considered the mean squared error (MSE) over the test set as the measure for comparison purposes.

Table 3.1: Average MSE over the test set considering Semeion Handwritten Digit dataset.

	0.1	0.2	0.5	0.8	1.0	1.2	1.5	2.0	Gompertz
DBM-CD	**0.18518**	**0.18503**	**0.18504**	0.19087	0.19718	0.20432	0.21495	0.21591	0.26833
DBM-PCD	**0.18527**	**0.18606**	**0.18655**	0.19154	0.19735	0.20511	0.21423	0.21532	0.27248
DBN-CD	0.21613	0.21977	0.21814	0.21465	0.21352	0.21413	0.21725	0.22455	0.22142
DBN-PCD	0.21051	0.21155	0.21660	0.21104	0.21012	0.21031	0.21080	0.21431	0.21617

One can observe the best results were obtained by DBM when using $T \in \{0.1, 0.2, 0.5\}$. Also, DBN-CD benefit from lower temperatures, thus confirming the results obtained by Lin et al. (LI et al., 2016a), i.e., the lower the temperature the higher the entropy. In short, we can learn more information at low temperatures, thus obtaining better results (obviously, we are constrained to a minimum bound concerning the temperature). According to Ranzato et al. (RANZATO; BOUREAU; CUN, 2008), sparsity in the neuron's activity favors the power of generalization of a network, which is somehow related to dropping neurons out in order to avoid overfitting (SRIVASTAVA et al., 2014b).

We have observed the lower the temperature values, the higher the probability of turning "on" hidden units (Equation 3.5), which forces DBM to push down the weights ($\mathbf{W}$) looking at sparsity. When we push the weights down, we also decrease the probability of turning on the hidden units, i.e., we try to deactivate them, thus forcing the network

[3] One sampling iteration was used for all learning algorithms.

[4] We did not fine-tune parameters using back-propagation, since the main goal of this chapter is to show the temperature does affect the behavior of DBMs.

to learn by other ways. We observed the process of pushing the weights down to be more "radical" at lower temperatures. Additionally, the Gompertz function did not achieve good results for DBMs, but close ones considering DBNs.

Figure 3.2 displays the values of the connection weights between the input and the first hidden layer. Since we used an architecture with 500 hidden neurons in the first layer, we chose 225 neurons at random to display what sort of information they have learned. According to Table I, some of the better results were obtained using $T = 0.5$ (Figure 3.2b), with $T = 1$ (Figure 3.2c) achieving close results either, which can be observed in the images either. Notice we can observe some digits at these images (e.g., highlighted regions in Figure 3.2b), while they are scarce in others. Additionally, DBNs seemed to benefit from lower temperatures, but their results were inferior to the ones obtained by DBMs. Once again, the Gompertz function did not obtain suitable results concerning DBMs.

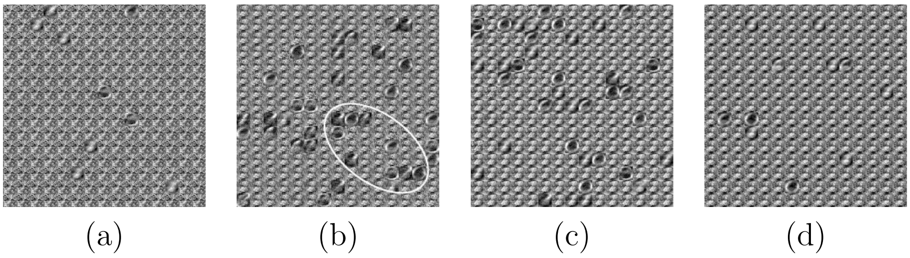

(a) (b) (c) (d)

Figure 3.2: Effect of different temperatures by means of DBM-PCD considering Semeion Handwritten Digit dataset with respect to the connection weights of the first hidden layer for: (a) $T = 0.1$, (b) $T = 0.5$, (c) $T = 1.0$, (d) $T = 2.0$.

Table 3.2 displays the MSE results over MNIST dataset, where the best results were obtained with $T = 0.2$. Once again, the results confirmed the hypothesis that better results can be obtained at lower temperatures, probably due to the lower interaction between visible and hidden units, which may imply in a slower convergence, but avoiding local optima (learning in DBMs is essentially an optimization problem, where we aim at minimizing the energy of each training sample in order to increase its probability - Equations 2.9 and 2.10). Figure 3.3 displays the connection weights between the input and the first hidden layer concerning DBM-PCD, where the highlighted region depicts some important information learned from the hidden neurons. Notice the neurons do not seem to contribute a lot with respect to different information learned from each other at higher temperatures (Figure 3.3d), since most of them have similar information encoded.

Figures 3.4 and 3.5 depict the evolution of two distinct measures to monitor the learning process of a DBM at the first layer over the Semeion Handwritten Digits and MNIST datasets, respectively. Figures 3.4a and 3.5a are the logarithms of the pseudo-likelihood

Table 3.2: Average MSE over the test set considering MNIST dataset.

	0.1	0.2	0.5	0.8	1.0	1.2	1.5	2.0	Gompertz
DBM-CD	0.08298	**0.08164**	0.08676	0.08868	0.09109	0.09230	0.09347	0.09338	0.10257
DBM-PCD	0.08238	0.08280	0.08650	0.08866	0.09105	0.09227	0.09352	0.09335	0.10036
DBN-CD	0.08993	0.09432	0.09259	0.09012	0.08933	0.08924	0.08966	0.09110	0.10629
DBN-PCD	0.08784	0.08811	0.08919	0.08874	0.08833	0.08820	0.08838	0.08994	0.11112

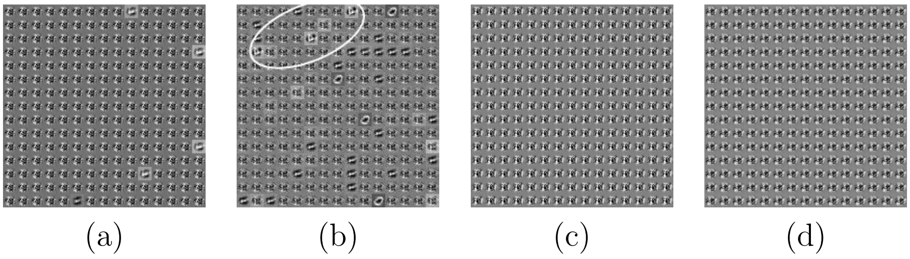

(a) (b) (c) (d)

Figure 3.3: Effect of different temperatures by means of DBM-PCD considering MNIST dataset with respect to the connection weights of the first hidden layer for: (a) $T = 0.1$, (b) $T = 0.5$, (c) $T = 1.0$, (d) $T = 2.0$.

(PL) of Equation 3.2, where the larger its value, the more similar the reconstructed data is concerning its original version. Figures 3.4b and 3.5b concern the mean squared error of the reconstruction data over the training set, where the lower values are the best ones. One can observe Gompertz presents an oscillatory behavior regarding PL values over Semeion dataset, while the lowest errors are achieved with the values $T \in \{0.2, 0.5, 0.8\}$. Regarding MNIST dataset, lower temperatures obtained the largest PL values (e.g. $T = 0.1$ and $T = 0.5$) and the lowest errors ($T = 0.1$ and $T = 0.2$) in Figure 3.5.

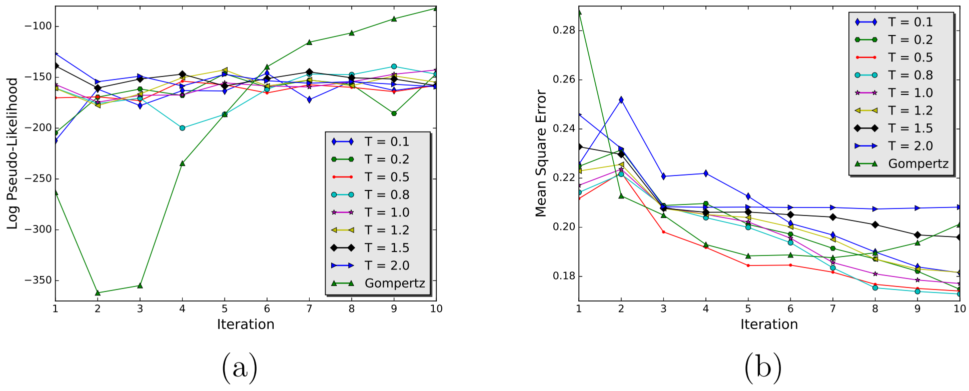
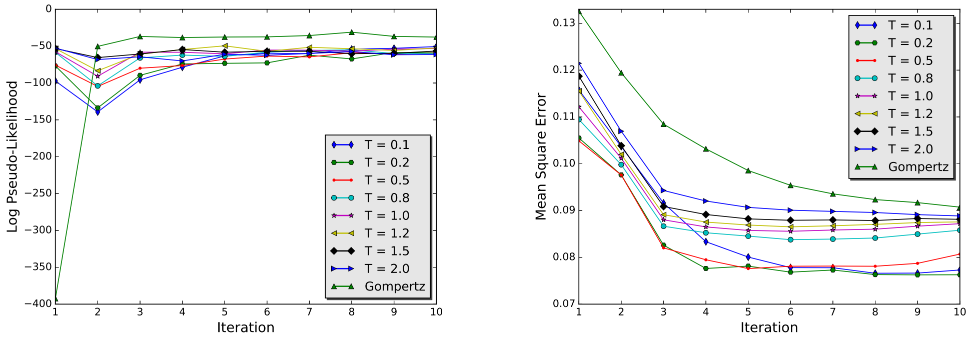


Figure 3.4: Evolution of the: (a) logarithm of the pseudo-likelihood, and (b) mean squared error by means of DBM-PCD considering Semeion Handwritten Digit dataset.

Table 3.3 presents the MSE results obtained over Caltech 101 Silhouettes dataset, where the lower temperatures obtained the best results, but being statistically similar to other temperatures (except for $T = 2.0$). In this case, both DBM and DBN obtained



(a) (b)

Figure 3.5: Evolution of the: (a) logarithm of the pseudo-likelihood and (b) mean squared error by means of DBM-PCD considering MNIST dataset.

similar results. Since this dataset comprises a number of different objects and classes, it is more complicated to figure out some shape with respect to the neurons' activity in Figure 3.6. Curiously, the neurons' response at the lower temperatures (Figure 3.6a) led to a different behavior that has been observed in the previous datasets, since the more "active" neurons with respect to different information learned were the ones obtained with $T = 2$ at the training step. We believe such behavior is due to the number of iterations for learning used in this chapter, which might not be enough for convergence purposes at lower temperatures, since this dataset poses a greater challenge than the others (it has a great intra-class variability).

We can also observe the best results were obtained by Gompertz function, which were much better than the standard Sigmoid ones. The Gompertz function is not symmetric, i.e., in the standard Logistic-Sigmoid function, we can obtain a probability equal or greater than 50% to activate a given neuron when the input to the function is a positive value, and a probability smaller then 50% when the input value is negative. However, such behavior can not be observed in the Gompertz function, where most of its coverage area (co-domain) is located above the 50% of probability of neuron activation, i.e., we can obtain values greater then 50% with negative input values (domain) as well. Therefore, this means that Gompertz function also forces the weights to be pushed down, similarly to lower temperature values.

3.5 Conclusions and Future Works

In this work, we dealt with the problem of different temperatures at the DBM learning step. Inspired by a very recent work that proposed the Temperature-based Restricted

Table 3.3: Average MSE over the test set considering Caltech 101 Silhouettes dataset.

	0.1	0.2	0.5	0.8	1.0	1.2	1.5	2.0	Gompertz
DBM-CD	0.16052	0.16133	0.16392	0.16383	0.16365	0.16243	0.16203	0.16025	0.16928
DBM-PCD	0.16077	0.16152	0.16364	0.16342	0.16261	0.16271	0.16197	0.16033	0.16791
DBN-CD	0.16061	0.16107	0.16269	0.16301	0.16320	0.16310	0.16320	0.16282	**0.14932**
DBN-PCD	0.16062	0.16085	0.16146	0.16158	0.16158	0.16190	0.16176	0.16267	0.16104

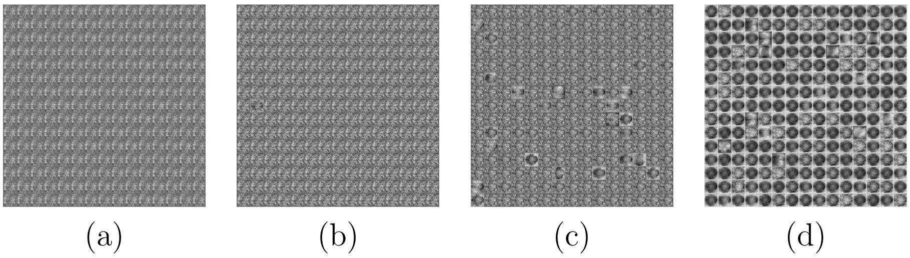
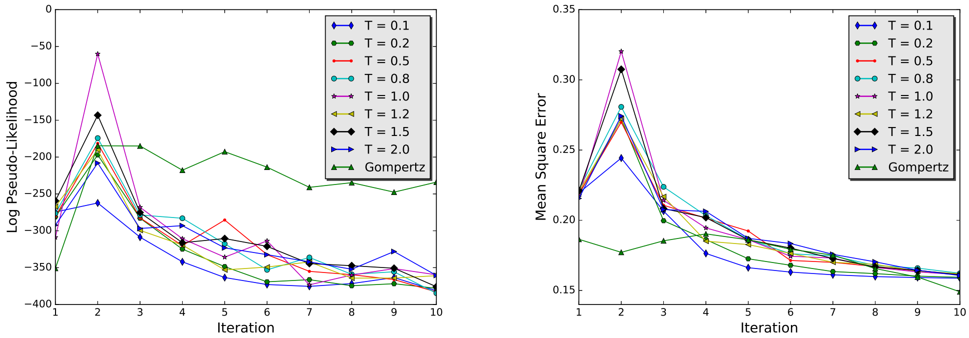


Figure 3.6: Effect of different temperatures by means of DBM-PCD considering Caltech 101 Silhouettes dataset with respect to the connection weights of the first hidden layer for: (a) $T = 0.1$, (b) $T = 0.5$, (c) $T = 1.0$, (d) $T = 2.0$..

Boltzmann Machines (LI et al., 2016a), we decided to evaluate the influence of the temperature when learning with Deep Boltzmann Machines aiming at the task of binary image reconstruction. Our results confirm the hypothesis raised by Li et al. (LI et al., 2016a), where the lower the temperature, the more generalized is the network for some applications. Thus, more accurate results can be obtained.

We observed the network pushes the weights down at lower temperatures in order to favor the sparsity, since the probability of tuning on hidden units is greater at lower temperatures. Therefore, by using proper temperature values, one can obtain a learning algorithm that can converge faster, thus saving computational resources and learning more accurate features. On the other hand, there is a need to fine-tune this hyper-parameter, which is application-dependent.

In regard to future works, we aim to propose an adaptive temperature, which can be linearly increased/decreased along the iterations in order to speed up the convergence process. Additionally, we will model the problem of learning proper temperature values by means of meta-heuristic-based optimization techniques.



(a) (b)

Figure 3.7: Evolution of log pseudo-likelihood and mean squared error for (a) and (b), respectively, by means of DBM-PCD considering Caltech 101 Silhouettes dataset.

Chapter 4

Deep Boltzmann Machines Using Adaptive Temperatures

Chapter 4 is continuity of the work started in Chapter 3. Here, one can observe the behavior of DBMs under adaptive temperatures. The work was presented in the 17th International Conference on Computer Analysis of Images and Patterns (PASSOS; COSTA; PAPA, 2017)

4.1 Introduction

In the last years, deep learning-driven techniques have been the foremost feature learner tools for a number of applications, that range from object detection to speech recognition, just to name a few. Such techniques are based on the hierarchical-oriented mechanism of the human brain, which learns different levels of information at each processing step. Convolutional Neural Networks (LECUN et al., 1998), Deep Belief Networks (HINTON; OSINDERO; TEH, 2006a), and Deep Boltzmann Machines (SALAKHUTDINOV; HINTON, 2012a) appear to be the most used techniques concerning the deep learning paradigm.

Deep Boltzmann Machines and Deep Belief Networks extend the well-known Restricted Boltzmann Machines to deeper representations, since they are composed of RBMs stacked on top of each other. In a nutshell, RBMs are stochastic neural networks composed of an input and a latent (i.e., hidden) layer, being the latter one in charge of learning the probability distribution of the input data. Roughly speaking, DBNs and DBMs differ in the way the upper layers interact, thus leading to slightly different formulations.

The main problem related to deep architectures concerns the large amount of data that is required for learning purposes; otherwise, the technique may overfit the data. As a consequence, a number of works have focused on mitigating such drawback, such as regularizing techniques (WAN et al., 2013; SRIVASTAVA et al., 2014b) and parameter fine-tuning (PAPA et al., 2015b; PAPA; ROSA; YANG, 2016; PAPA et al., 2015d; ROSA et al., 2015; PAPA; SCHEIRER; COX, 2016a). An interesting approach related to RBM-based techniques concerns working on the "stability" of the convergence process to prevent overfitting. Recently, Li et al. (LI et al., 2016b) studied the influence of different temperatures during DBN learning procedure, and later on Passos and Papa (PASSOS; PAPA, 2017c) conducted a similar work, tough in the context of Deep Boltzmann Machines. Both studies agreed that temperature helps preventing overfitting, where the lower the temperature values, the better the results. The aforementioned works concluded that low temperature values lead to higher sparsity levels, thus contributing to the regularization of the network. As a matter of fact, sparsity is somehow analogous to dropping out neurons, i.e., one can switch neurons "on" or "off", forcing the network to adapt under such circumstances.

Basically, the problem of learning weights in the RBM training procedure aims at minimizing the energy of each training sample, which leads us to increasing its probability. Therefore, the training procedure of RBMs and related approaches is nothing more than an optimization process. In this work, we borrow the idea from meta-heuristic-based optimization processes, which aim at finding the best trade-off between exploitation and exploration. The first term refers to improving the solutions around the neighborhood of a given sample (local search), meanwhile exploration focuses on improving the solution in far away locations (e.g., global search). At the very beginning of the optimization process, meta-heuristic techniques usually converge faster (high exploration), thus decreasing the step-size (high exploitation) along the iterations in order to avoid overshooting the global/near-global optimum.

Therefore, we propose to use an adaptive temperature-based schema, where the temperature (step-size) decreases along the training procedure, thus simulating the behaviour of exploitation and exploration found out in many meta-heuristic techniques. We showed the proposed approach can outperform temperature-fixed DBNs and DBMs in the context of binary image reconstruction for some situations, or it can be at least competitive to them. Additionally, the adaptive-driven approach does not need a fine-tuning step since it requires the minimum and maximum temperature values only, which are considerably less sensitive and easy to set than the temperature value itself.

In this chapter, we also considered two different formulations to control the temperature values. The remainder of this chapter is organized as follows. Section 4.2 presents the temperature-based DBMs and the approaches used in this work. The methodology and experiments are presented in Sections 4.3 and 4.4, respectively, and Section 4.5 states conclusions and future works.

4.2 Temperature-based Deep Boltzmann Machines

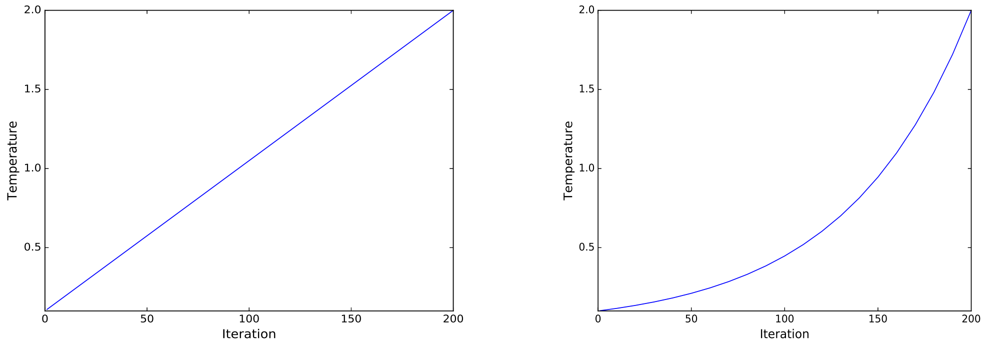
The theoretical background regarding temperature-based DBM is presented in Section 3.2.1. The following section presents the adaptive temperature approach.

4.2.1 Adaptive Temperature-based Model

In this chapter, we study the influence of two different functions during the convergence process:

- $f_1(t) = L - \frac{t}{t_{max}}(L-U)$; and
- $f_2(t) = L\exp((\log\left(\frac{U}{L}\right)/t_{max})t)$.

In the above functions, $L = 0.1$ and $U = 2.0$ stand for the lower and upper temperature boundaries, respectively. Also, $t_{max} = 200$ denotes the maximum number of iterations concerning DBN/DBM learning procedure. Figures 4.1a and 4.1b display the behaviour of functions f_1 and f_2, respectively. In a nutshell, f_1 stands for a bounded linear function, meanwhile f_2 represents a bounded exponential function. The reason for using functions bounded in $[0.1, 2.0]$ concerns the fact that lower temperatures lead to better results (LI et al., 2016b; PASSOS; PAPA, 2017c).



(a)

(b)

Figure 4.1: Function F_1 and F_2 for (a) and (b), respectively. Used to update temperature values along the convergence process.

Additionally, we used 100 iterations with step-size of 10 for convergence purposes (i.e., the temperature changes every 10 iterations). Although such number may not be enough to achieve state-of-the-art results, we would like to emphasize we are interested into showing the proposed approach can outperform temperature-fixed ones even using a small number of iterations.

4.3 Methodology

In this section, we present the methodology employed to evaluate the proposed approach, as well the datasets and the experimental setup. Notice the approach used in this chapter is based on the one employed by Passos et al. (PASSOS; PAPA, 2017c).

4.3.1 Datasets

We propose to evaluate the behavior of DBNs and DBMs under adaptive temperatures in the context of binary image reconstruction using two public datasets, i.e., MNIST and CalTech 101 Silhouettes Data Set, presented in Section 2.9.

4.3.2 Experimental Setup

We employed a 3-layered architecture for all datasets as follows: I-500-500-2,000, where I stands for the number of pixels used as input for each dataset, i.e., 196 (14×14 images) and 784 (28×28 images) considering MNIST and Caltech 101 Silhouettes datasets, respectively. Therefore, we have a first and a second hidden layers with 500 neurons each, followed by a third hidden layer with 2,000 neurons[1]. The remaining parameters used during the learning steps were chosen empirically and fixed for each layer as follows: $\eta = 0.1$ (learning rate), $\lambda = 0.1$ (weight decay), $\alpha = 0.00001$ (penalty parameter).

In order to provide a statistical analysis by means of the Wilcoxon signed-rank test with significance of 0.05 (WILCOXON, 1945), we conducted a cross-validation procedure with 20 runnings. In regard to the fixed-temperature experiment, we considered a set of values within the range $T \in \{0.1, 0.2, 0.5, 0.8, 1.0, 1.2, 1.5, 2.0\}$ for the sake of comparison purposes.

[1] Since this architecture has been commonly employed in several works in the literature, we opted to employ it in our work either.

Finally, we employed 100 epochs for DBM and DBN learning weights procedure with mini-batches of size 20. In order to provide a more precise experimental validation, we trained both DBMs and DBNs with two different algorithms[2]: Contrastive Divergence (CD) (HINTON; OSINDERO; TEH, 2006a) and Persistent Contrastive Divergence (PCD) (TIELEMAN, 2008a). Also, in order to evaluate the techniques considered in this work, we computed the mean square error (MSE) error over the training set. Therefore, the smaller the MSE, the better the technique is.

4.4 Experiments

This section presents the experimental results concerning DBN and DBM optimization by means of adaptive temperatures. Two different adaptive functions, i.e, f_1 and f_2, as well as eight constant temperatures were used for the baseline approach (i.e., fixed-size temperature): 0.1, 0.2, 0.5, 0.8, 1.0, 1.2, 1.5 and 2.0. Furthermore, DBM results were compared against DBN using two different learning algorithms, i.e., Contrastive Divergence and Persistent Contrastive Divergence in a three-layered model. Table 4.1 presents the average MSE results for DBMs and DBNs over Caltech 101 Silhouettes datasets. The most accurate results according to the Wilcoxon signed rank test are in bold.

Table 4.1: Average DBM/DBN MSE over the test set considering Caltech 101 Silhouettes dataset with 200 iterations.

	0.1	0.2	0.5	0.8	1.0	1.2	1.5	2.0	Linear	Curve
DBM-CD	0.16048	0.16048	0.16048	0.16049	0.16048	0.16049	0.16049	0.15983	**0.15822**	0.16053
DBM-PCD	0.16049	0.16049	0.16050	0.16048	0.16049	0.16048	0.16049	0.15983	**0.15929**	0.16039
DBN-CD	0.16049	0.16050	0.16049	0.16050	0.16049	0.16058	0.16249	0.17040	**0.15822**	0.16523
DBN-PCD	0.16048	0.16049	0.16049	0.16049	0.16048	0.16049	0.16081	0.16120	**0.15929**	0.16321

Clearly, the best results were obtained using the linear adaptive function for both DBMs and DBNs. A closer look may suggest that adaptive temperature optimization works well for challenging datasets, such as Caltech 101 Silhouettes. Also, one can observe that both DBMs and DBNs obtained pretty much similar results, which can be explained by the fact we are not fine-tuning DBMs with the mean-field learning process. However, it is beyond the scope of this work to show that DBMs may be more accurate than DBNs, since we are interested to show the robustness in using adaptive temperatures for both models.

Table 4.2 presents the behavior of adaptive temperatures concerning DBMs and DBNs

[2] One sampling iteration was used for all learning algorithms.

considering the MNIST dataset. Despite the adaptive linear function did not achieve the best results according to Wilcoxon signed-rank test, the difference between fixed- and adaptive-temperature is pretty much irrelevant. With respect to DBNs, both models evaluated in this work, i.e., fixed and adaptive temperatures, obtained quite close results. Additionally, in regard to DBMs, one can observe the best results were obtained with smaller temperatures, as discussed by Passos et al. (PASSOS; PAPA, 2017c). In this case, it is expected that adaptive models will not outperform fixed ones, since the temperature values in these dynamic approaches increase along the iterations.

Table 4.2: Average DBM/DBN MSE over the test set considering MNIST dataset with 200 iterations.

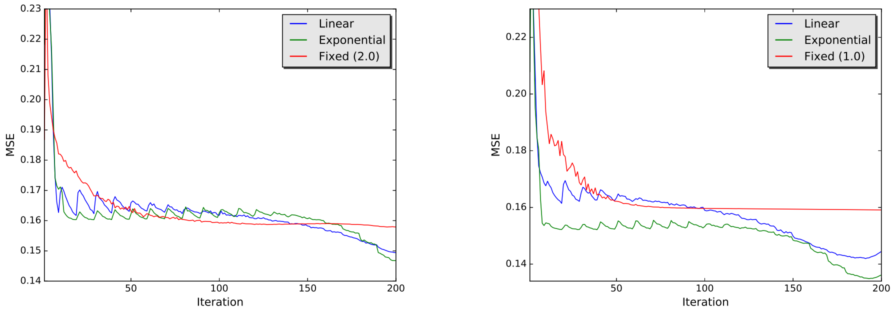
	0.1	0.2	0.5	0.8	1.0	1.2	1.5	2.0	Linear	Curve
DBM-CD	**0.08642**	**0.08642**	**0.08674**	0.08745	0.08753	0.08750	0.08751	0.08752	0.08747	0.08751
DBM-PCD	**0.08674**	**0.08659**	0.08681	0.08744	0.08752	0.08751	0.08750	0.08752	0.08747	0.08751
DBN-CD	0.08760	0.08771	0.08763	0.08752	**0.08751**	**0.08751**	**0.08751**	0.08749	0.08752	0.08775
DBN-PCD	0.08760	0.08769	0.08762	0.08751	**0.08751**	0.08751	**0.08750**	**0.08751**	0.08752	0.08775

We performed an extra round of experiments to analyze the impact of adaptive temperatures during the convergence process. For comparison purposes, we considered both the temperature value and learning algorithm that achieved the best results concerning the fixed-temperature approach. Figure 4.2 depicts the MSE of the first layer during the learning process across the iterations for both DBMs and DBNs considering the Caltech 101 Silhouettes dataset. Therefore, we compared DBM-CD with $T = 2.0$ against the proposed approach in Figure 4.2a, as well as we compared DBN-PCD with $T = 1.0$ against the proposed approach in Figure 4.2b.

Clearly, one can observe the adaptive temperatures converged faster during the first 50 iterations, and for DBN (Figure 4.2b) they did not get stuck in local optima, as one can observe in the experiment with the fixed temperature, which stabilized after 75 iterations. Also, it seems there is no difference in using the linear or exponential function to update the temperature values considering DBMs, while the exponential model seemed to fit better for DBNs, but for a very small difference.

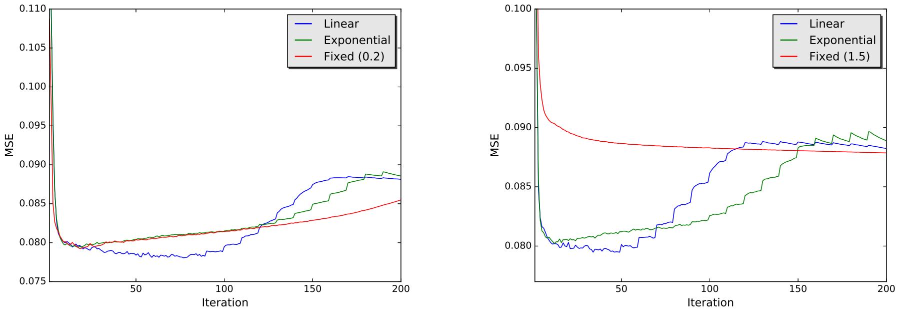
Figure 4.3 shows the very same procedure for MNIST dataset. Once again, the fast convergence of the proposed approaches can be evidenced. Notice that adaptive-temperatures achieve by far the lower MSE since the beginning, but the model starts to "unlearn" and moves back to a point where the MSE is higher than the one achieved by the fixed-temperature after a long period of training.

Roughly speaking, the proposed approaches can benefit in situations where higher



(a) (b)

Figure 4.2: MSE during the learning step of the first layer considering Caltech 101 Silhouettes dataset for (a) DBM and (b) DBN.



(a) (b)

Figure 4.3: MSE during the learning step of the first layer considering MNIST dataset for (a) DBM and (b) DBN.

temperatures lead to the better results. However, since the adaptive model always increases the temperature, one may not get suitable results at the very end of the convergence process, which means one can halt the process much earlier.

4.5 Conclusions

In this work, we dealt with the problem of hastening the DBM learning step using adaptive temperatures, as well as we also evaluated them in the context of DBNs. Recent works presented the influence of different temperatures during DBN (LI et al., 2016b) and DBM (PASSOS; PAPA, 2017c) learning process, which introduces an additional parameter to the model. Adaptive temperatures exempt the need for the aforementioned extra parameter, thus becoming easier to handle those models.

Furthermore, the experimental results over two public well-known datasets showed the technique is at least competitive to optimize DBMs and DBNs, outperforming temperature-fixed DBNs and DBMs in one of the cases, but being much faster for convergence at the early iterations in both datasets. In regard to future works, we aim to validate the proposed approach to reconstruct gray-scale images either.

Chapter 5

A Metaheuristic-Driven Approach to Fine-Tune Deep Boltzmann Machines

This chapter presents a paper under revision in the journal Applied Soft Computing. The scope of the work is to enhance the robustness of DBMs using a proper selection of its meta-parameters. For such task, eight distinct meta-heuristic optimization techniques are compared, using both classical and state-of-art models.

5.1 Introduction

Restricted Boltzmann Machines (RBMs) are probabilistic models that employ a layer of hidden binary units, also known as latent units, to model the distribution of the input data (visible layer). Such models have been applied to deal with problems involving images (HINTON; OSINDERO; TEH, 2006b; LAROCHELLE et al., 2007), text (WELLING; ROSEN-ZVI; HINTON, 2004; SALAKHUTDINOV; HINTON, 2009b), and detection of malicious content (FIORE et al., 2013a; SILVA et al., 2016), just to name a few. Moreover, RBMs are also used for building deep learning architectures, such as Deep Belief Networks (DBNs) (HINTON; OSINDERO; TEH, 2006b) and Deep Boltzmann Machine (DBM) (SALAKHUTDINOV; HINTON, 2009a), where the main difference is related to the interaction among layers of RBMs.

Deep Learning techniques have been extensively used to deal with tasks related to signal processing and computer vision, such as feature selection (RUANGKANOKMAS; ACHALAKUL; AKKARAJITSAKUL, 2016) (SOHN; LEE; YAN, 2015), face (TAIGMAN et al., 2014) (DUONG et al., 2015) and image reconstruction (DONG et al., 2014), multimodal learning (SRIVASTAVA; SALAKHUTDINOV, 2012), and topic modeling (HINTON; SALAKHUT-

DINOV, 2009), among others. Despite the outstanding results obtained by these models, an intrinsic constraint associated with deep architectures is related to their complexity, which can become an insurmountable problem due to the hundreds of hyperparameters one must deal with. The present work focuses on this problem.

Some works have recently modeled the issue of hyperparameter fine-tuning as a metaheuristic optimization task. Such techniques show up as an interesting alternative for such a task since they do not require computing derivatives of hundreds of parameters as usually happen with standard optimization techniques, which is not recommended for high-dimensional spaces. Papa et. al. (PAPA et al., 2015b, 2015d; PAPA; SCHEIRER; COX, 2016b; ROSA et al., 2016a, 2015; RODRIGUES; YANG; PAPA, 2016; PASSOS; PAPA, 2017b) are among the first to introduce metaheuristic-driven optimization in the context of RBMs, DBNs, Infinity Restricted Boltzmann Machines (iRBMs), and Convolutional Neural Networks fine-tuning, obtaining more precise results than the ones achieved using some well-known optimization libraries in the literature.

However, as far as we are concerned, metaheuristic approaches have never been used for the task of DBM hyperparameter optimization, which turns out to be one of the main contributions of this paper. Therefore, in this work, we considered DBM fine-tuning using seven techniques: Improved Harmonic-Search (IHS) (MAHDAVI; FESANGHARY; DAMANGIR, 2007), Adaptive Inertia Weight Particle Swarm Optimization (AIWPSO) (YU; LIU; LI, 2009), Cuckoo Search (CS) (MAHDAVI; FESANGHARY; DAMANGIR, 2007) (YANG; DEB, 2009), Firefly Algorithm (FA) (YANG, 2010), Backtracking Search Optimization Algorithm (BSA) (CIVICIOGLU, 2013), Adaptive Differential Evolution (JADE) (ZHANG; SANDERSON, 2009), and the Differential Evolution Based on Covariance Matrix Learning and Bimodal Distribution Parameter Setting Algorithm (CoBiDE) (WANG et al., 2014). Furthermore, all techniques are compared with a random search for experimental purposes. The application addressed in this paper concerns the task of binary image reconstruction, and for that purpose, we considered three public datasets.

In a nutshell, the main contribution of this paper is to introduce metaheuristic optimization to the context of DBM hyperparameter fine-tuning, as well as to foster the research towards such area. Additionally, we provided an extensive experimental evaluation with distinct learning algorithms over a different number of layers. As far as we are concerned, we have not observed any study with such level of details. The remainder of this paper is presented as follows. Section 5.2 introduces the main foundations related to the metaheuristic optimization techniques employed in this work. Sections 5.3 and 5.4

present the methodology and experiments, respectively, and Section 5.5 states conclusions and future works.

5.2 DBM Fine-Tuning as an Optimization Problem

In general, Restricted Boltzmann Machines demands a proper selection of four main parameters: number of hidden units n, the learning rate η, the weight decay λ, and the momentum φ. Since Deep Boltzmann Machines stack RBMs on top of each other, if one has L layers, then each optimization encodes $4L$ variables to be optimized. However, as the training procedure of DBMs are greedy-wise (we are not considering mean-field-based learning in this work), which means each layer is trained independently, only 4 variables are optimized per layer.

In short, the idea is to initialize all optimization techniques at random, and them the algorithm takes place. The following ranges were considered in this work parameters[1]: $\eta \in [0.1, 0.9]$, $n \in [5, 100]$, $\varphi \in [0.00001, 0.01]$ and $\lambda \in [0.1, 0.9]$. Aiming to fulfill the requirements of any optimization technique, one shall design a fitness function to guide the search into the best solutions. For such purpose, the mean squared error (MSE) over the training set was considered for the task of binary image reconstruction as the fitness function. Therefore, we adopted the very same methodology used by (PAPA; SCHEIRER; COX, 2016b) to allow a fair comparison against the works. Figure 7.3 depicts the optimization model employed in this paper. In short, the approach proposed in this paper models the whole set of $4L$ decision variables as being an optimization agent.

Figure 5.2 presents an overall idea of the pipeline used in this work to perform DBM hyperparameter fine-tuning. Roughly speaking, the optimization technique selects the set of hyperparameters that minimize the MSE over the training set considering a dataset of binary images as an input to the model. After learning the hyperparameters, one can proceed to the reconstruction step concerning the testing images, whose MSE is the one used to finally evaluate the metaheuristic techniques considered in this work.

[1] The ranges used for each parameter were empirically selected based on values commonly adopted in the literature (PAPA et al., 2017a; ROSA et al., 2016a; PAPA et al., 2015b; RODRIGUES; YANG; PAPA, 2016; PASSOS; PAPA, 2017b)

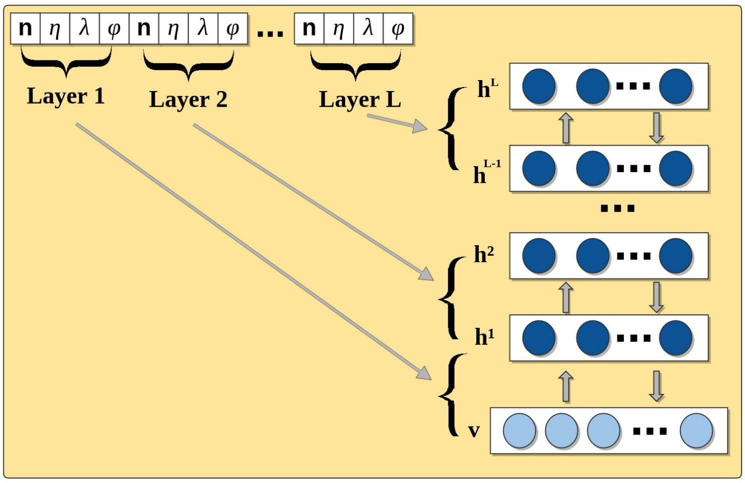


Figure 5.1: Proposed approach to encode the decision variables of each optimization agent.

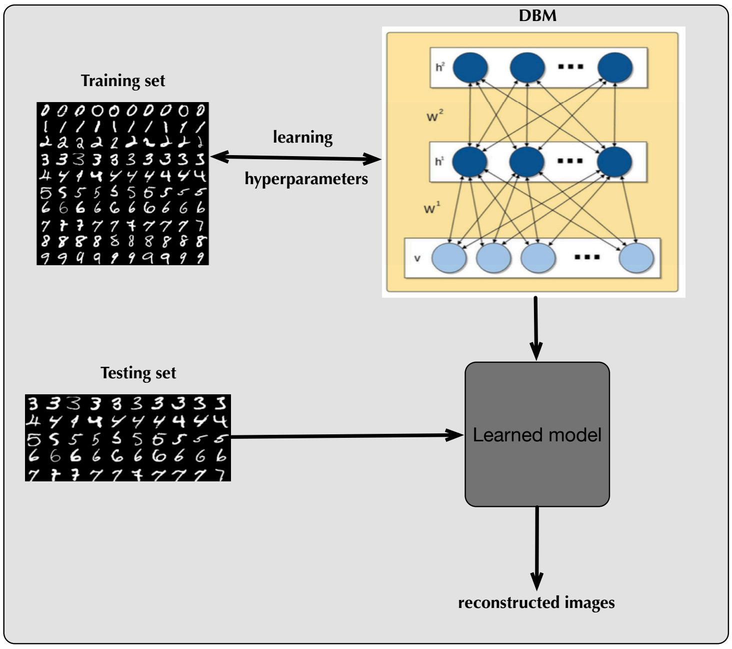


Figure 5.2: Proposed approach to encode the decision variables of each optimization agent.

5.2.1 Optimization Techniques

This work employs seven metaheuristic techniques to the task of DBM fine-tuning, i.e., IHS, AIWPSO, CS, FA, BSA, JADE, and CoBiDE, presented in Section 2.8.1.

5.3 Methodology

In this section, we describe the datasets and the experimental setup employed in this work.

5.3.1 Datasets

We validate DBM fine-tuning in the task of binary image reconstruction over three public datasets, i.e., MNIST, CalTech 101 Silhouettes Data Set, and Semeion Handwritten Digit Data Set, presented in Section 2.9.

5.3.2 Parameter Setting-up

One of the main shortcoming in using RBM-based models, such as DBM and DBN, concerns their fine-tuning hyperparameter task, which aims at selecting a suitable set of parameters in such a way that the reconstruction error is minimized. In this work, we considered IHS, FA, CS, AIWPSO, BSA, JADE, and the CoBiDE against RS for DBM hyperparameter fine-tuning. We also evaluated the robustness of the proposed approach using three distinct DBN and DBM models: one layer (1L), two layers (2L) and three layers (3L). Finally, Table 6.1 presents the parameters used for each optimization technique[2], where 5 agents (initial solutions) were used for all optimization techniques during 50 iterations for convergence [3].

Table 5.1: Parameter configuration.

Technique	Parameters
IHS	$HMCR = 0.7$, $PAR_{MIN} = 0.1$ $PAR_{MAX} = 0.7$, $\rho_{MIN} = 1$ $\rho_{MAX} = 10$
AIWPSO	$c_1 = 1.7$, $c_2 = 1.7$
CS	$\alpha = 0.1$, $\alpha_{MIN} = 0.5$, $\alpha_{MAX} = 1$ $p = 0.25$, $p_{MIN} = 0.05$, $p_{MAX} = 0.5$
FA	$\gamma = 1$, $\beta = 1$, $\alpha = 0.2$
BSA	$mix\_rate = 1.0$, $F = 3$
JADE	$c = 0.1$, $g = 0.05$
CoBiDE	$pb = 0.4$, $ps = 0.5$

[2] Parameters were empirically selected based on each technique author's suggestions, as well as the values commonly adopted in the literature (PAPA et al., 2017a; ROSA et al., 2016a; PAPA et al., 2015b; RODRIGUES; YANG; PAPA, 2016; PASSOS; PAPA, 2017b)

[3] The selected number of agents and iterations for convergence were empirically chosen based on values commonly adopted in the literature (PAPA et al., 2017a; ROSA et al., 2016a; PAPA et al., 2015b).

We conducted a cross-validation approach with 20 runnings, 10 iterations for the learning procedure of each RBM, and mini-batches of size 20. In addition, we also considered two learning algorithms: Contrastive Divergence (CD) (HINTON, 2002) and Persistent Contrastive Divergence (PCD) (TIELEMAN, 2008b). Finally, the Wilcoxon signed-rank test (WILCOXON, 1945) with significance of 0.05 was used for statistical validation purposes.

Finally, the codes used to reproduce the experiments of the paper are available on GitHub[4][5][6][7]. The experiments were conducted using an Ubuntu 16.04 Linux machine with 64Gb of RAM running an 2x Intel Xeon Bronze 3106 with a frequency of 1.70 GHz. All the coding was built in C.

5.4 Experiments

In this section, we present the experimental results concerning DBM and DBN hyperparameter optimization on the task of binary image reconstruction. Both techniques were compared using two different learning algorithms, i.e. Contrastive Divergence and Persistent Contrastive Divergence. Also, seven optimization methods were employed. Additionally, three distinct models used for comparison purposes: one layer (1L), two layers (2L), and three (3L) layers.

5.4.1 Experimental Results

Tables 5.2 presents the average values of the minimum squared error over the MNIST dataset, being the values in bold the best results considering the Wilcoxon signed-rank test. One can observe the metaheuristic techniques obtained the best results, with special attention to IHS, JADE, and CoBiDE for both DBN and DBM models. Also, one can not figure a considerable difference between shallow and deep models, since we limited the number of iterations for convergence to 10, as well as we did not employ fine-tuning as a final step for DBN and DBM connection weights. The main reasons for limiting the number of iterations are related to time constraints, as well as the convergence process itself. As a matter of fact, if one has unlimited resources in terms of computational load, a standard random search may obtain results as good as the ones obtained by metaheuristic

[4] LibOPF: https://github.com/jppbsi/LibOPF

[5] LibDEEP: https://github.com/jppbsi/LibDEEP

[6] LibDEV: https://github.com/jppbsi/LibDEV

[7] LibOPT (PAPA et al., 2017b): https://github.com/jppbsi/LibOPT

techniques, since they will have enough time for convergence. However, we would like to emphasize that DBM hyperparameter fine-tuning is quite useful when time is limited and a serious constraint.

Table 5.2: Average MSE values considering MNIST dataset.

	1L				2L				3L			
	DBN		DBM		DBN		DBM		DBN		DBM	
	CD	PCD	CD	PCD	CD	PCD	CD	PCD	CD	PCD	CD	PCD
IHS	**0.08758**	0.08762	**0.08744**	0.08766	0.08762	0.08762	0.08761	0.08761	0.08762	0.08762	0.08760	0.08761
AIWPSO	0.08764	0.08761	0.08765	0.08771	0.08763	0.08762	0.08762	0.08761	0.08762	0.08762	**0.08759**	0.08760
CS	0.08763	0.08764	0.08767	0.08770	0.08764	0.08765	0.08760	0.08760	0.08764	0.08765	0.08762	0.08761
FA	0.08763	0.08764	0.08766	0.08762	0.08763	0.08763	0.08761	0.08763	0.08763	0.08763	0.08761	0.08761
BSA	0.08762	0.08762	0.08774	0.08766	0.08762	0.08763	0.08761	0.08762	0.08763	0.08762	0.08762	0.08762
JADE	0.08760	0.08763	**0.08754**	**0.08749**	0.08763	0.08764	0.08761	0.08761	0.08763	0.08763	0.08761	0.08761
CoBiDE	0.08763	0.08762	**0.08757**	0.08765	0.08763	0.08764	0.08762	0.08760	0.08763	0.08762	0.08761	0.08760
RS	0.08762	0.08763	0.08780	0.08782	0.08762	0.08763	0.08761	0.08760	0.08763	0.08763	0.08761	0.08761

Table 5.3 presents the results concerning CalTech 101 Silhouettes dataset. In this case, the best results were achieved by DBN with one layer only. Caltech poses a greater challenge, since it has more classes than MNIST, which should us to believe more iterations for convergence would be required for DBM learning, since it a more complex model than DBN. Also, the best results were obtained by means of Improved Harmony Search, BSA, JADE, and CoBiDE.

Table 5.3: Average MSE values considering CalTech 101 Silhouettes dataset.

	1L				2L				3L			
	DBN		DBM		DBN		DBM		DBN		DBM	
	CD	PCD	CD	PCD	CD	PCD	CD	PCD	CD	PCD	CD	PCD
IHS	**0.15554**	0.15731	0.15983	0.15980	0.16057	0.16054	0.16055	0.16055	0.16059	0.16058	0.16057	0.16056
AIWPSO	0.15641	0.15825	0.16006	0.16014	0.16056	0.16060	0.16056	0.16061	0.16058	0.16057	0.16057	0.16057
CS	0.15923	0.15992	0.16023	0.16024	0.16057	0.16062	0.16057	0.16056	0.16059	0.16061	0.16055	0.16057
FA	0.16002	0.15956	0.16051	0.16034	0.16060	0.16058	0.16069	0.16056	0.16060	0.16058	0.16055	0.16055
BSA	**0.15599**	0.15775	0.15992	0.15983	0.16056	0.16056	0.16052	0.16054	0.16057	0.16058	0.16057	0.16055
JADE	**0.15608**	0.15790	0.15945	0.15988	0.16058	0.16057	0.16055	0.16058	0.16059	0.16057	0.16058	0.16054
CoBiDE	**0.15638**	0.15800	0.15982	0.15982	0.16059	0.16057	0.16059	0.16056	0.16060	0.16059	0.16056	0.16054
RS	0.15676	0.15845	0.15967	0.15976	0.16060	0.16062	0.16059	0.16057	0.16057	0.16056	0.16056	0.16056

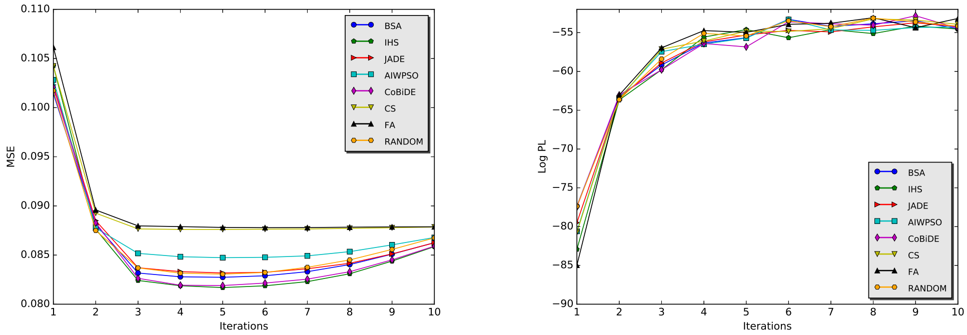
Table 5.4 presents the results obtained over Semeion Handwritten Digit dataset, being IHS and JADE the most accurate techniques. The best results concerning MNIST and Semeion Handwritten Digits datasets, as can be clearly seen on Tables 5.2 and 5.4, was acquired using the DBM. DBN, however, had the best results considering CalTech 101 Silhouettes dataset, as presented in Table 5.3. Some interesting conclusions can be extracted from a closer look at these results: (i) meta-heuristic-based optimization allows more accurate results than a random search, as argued by the works of Papa et al. (PAPA et al., 2015b, 2015d; PAPA; SCHEIRER; COX, 2016b) already; (ii) DBMs seem to produce

more accurate results than DBNs; (iii) the number of layers do not seem to influence the results when one fine-tune parameters; (iv) IHS achieved the best results in all datasets (concerning both DBN and DBN), but with results statistically similar to other meta-heuristic techniques as well; and (v) we could not realize a significant difference between CD and PCD, since we employed 10 iterations for learning only. Actually, PCD is expected to work better, but at the price of a longer convergence process.

Table 5.4: Average MSE values considering Semeion Handwritten Digit dataset.

	1L				2L				3L			
	DBN		DBM		DBN		DBM		DBN		DBM	
	CD	PCD	CD	PCD	CD	PCD	CD	PCD	CD	PCD	CD	PCD
IHS	0.19359	0.20009	**0.19025**	**0.19078**	0.20961	0.20961	0.20956	0.20956	0.20961	0.20963	0.20958	0.20958
AIWPSO	0.20044	0.20274	0.19679	0.19426	0.20959	0.20961	0.20958	0.20956	0.20964	0.20961	0.20959	0.20959
CS	0.20528	0.20647	0.20728	0.20651	0.20965	0.20960	0.20957	0.20959	0.20964	0.20963	0.20960	0.20960
FA	0.20638	0.20894	0.20649	0.20319	0.20966	0.20965	0.20960	0.20960	0.20964	0.20965	0.20960	0.20928
BSA	0.19571	0.20002	0.19221	0.19325	0.20961	0.20959	0.20960	0.20958	0.20962	0.20962	0.20960	0.20956
JADE	0.19893	0.20165	**0.19152**	**0.19170**	0.20962	0.20960	0.20957	0.20958	0.20964	0.20959	0.20956	0.20961
CoBiDE	0.19328	0.19896	0.19190	0.19138	0.20962	0.20961	0.20959	0.20958	0.20960	0.20961	0.20958	0.20959
RS	0.19710	0.20361	0.19458	0.19463	0.20962	0.20959	0.20960	0.20957	0.20960	0.20960	0.20960	0.20959

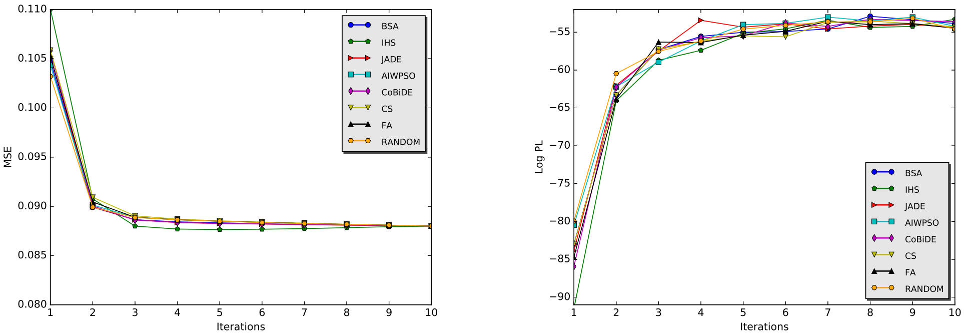
Figures 5.3 and 5.4 display the convergence process regarding the mean squared error (MSM) and logarithm of the pseudo-likelihood (PL) values obtained during the learning step for DBM and DBN, respectively, trained with CD over MNIST dataset. We used the mean values of the first layer for all optimization algorithms. One can observe DBM obtained the better approximation of the model during all iterations, and both ended up with similar log PL values (iteration #10). However, it is important to shed light over the main contribution of this paper is not to show DBM may learn better models than DBNs, but to stress meta-heuristic techniques are suitable to fine-tune DBM parameters as well.



(a) (b)

Figure 5.3: MSE and Log PL values during the convergence process considering DBM over MNIST dataset for (a) and (b), respectively.

Although one can realize an oscillating behavior of the optimization techniques, all of them obtained better models at the last iteration (i.e. a highest log PL) than RS, except for the nature-inspired algorithms, that achieved similar results in most of the experiments, probably due to its demand for more iterations to convergence. The results implies that using meta-heuristic techniques to fine-tune DBMs seems to be reasonable. DBMs optimized by meta-heuristic-based techniques obtained the best results considering all datasets used in this work as well.



(a) (b)

Figure 5.4: MSE and Log PL values during the convergence process considering DBN over MNIST dataset for (a) and (b), respectively.

5.4.2 Statistical Analysis

In this section, we detailed the Wilcoxon signed-rank test obtained through a pairwise comparison among the techniques. For such purpose, we used 5% of significance to provide the statistical similarity among the best results obtained by each technique, i.e., considering both number of layers and learning algorithm. Tables 5.5, 5.6 and 5.7 presents the statistical evaluation concerning MNIST, CalTech 101 Silhouettes, and Semeion datasets.

It is interesting to point out that memory- (IHS) and evolutionary-based (BSA, JADE, and CoBiDE) techniques obtained the best results for all datasets, outperforming swarm collective approaches (AIWPSO, FA, and CS). Regarding evolutionary techniques, mutation and crossover operators may move solutions far apart from each other (i.e., they favor the exploration), which can be interesting in the context of DBM/DBN hyperparameter fine-tuning. Usually, the hyperparameters we are optimizing (i.e., learning rate, number of hidden units, weight decay and momentum) do not lead to different reconstruction errors under some small intervals, i.e., the fitness landscape figures some flat zones that can trap optimization techniques.

Table 5.5: Statistical analysis considering MNIST dataset.

	IHS	AIWPSO	CS	FA	BSA	JADE	CoBiDE	RS
IHS								
AIWPSO	=							
CS	≠	=						
FA	≠	=	=					
BSA	≠	≠	=	=				
JADE	=	=	=	=	=			
CoBiDE	=	=	=	=	=	=		
RS	≠	=	=	=	=	=	=	

Table 5.6: Statistical analysis considering CalTech 101 Silhouettes dataset.

	IHS	AIWPSO	CS	FA	BSA	JADE	CoBiDE	RS
IHS								
AIWPSO	≠							
CS	≠	≠						
FA	≠	≠	=					
BSA	=	=	≠	≠				
JADE	=	=	≠	≠	=			
CoBiDE	=	=	≠	≠	=	=		
RS	≠	=	≠	≠	=	=	=	

Table 5.7: Statistical analysis considering Semeion dataset.

	IHS	AIWPSO	CS	FA	BSA	JADE	CoBiDE	RS
IHS								
AIWPSO	≠							
CS	≠	≠						
FA	≠	≠	=					
BSA	≠	=	≠	≠				
JADE	=	=	≠	≠	=			
CoBiDE	≠	=	≠	≠	=	=		
RS	≠	=	≠	≠	=	≠	≠	

Regarding the relatively good results obtained using the Random Search, one may question the contribution of employing metaheuristic techniques for DBM hyperparameter optimization. Despite the statistical similarity among optimization techniques, the random search did not obtain the best results for any dataset.

5.4.3 Time Analisys

Tables 5.8, 5.9, and 5.10 present an analysis of the computational load required by the optimization tasks regarding MNINST, CalTech 101 Silhouettes, and Semeion datasets, respectively. The results in bold stand for the fastest aproaches for each model.

Table 5.8: Computational load (in hours) considering MNIST dataset.

	1L				2L				3L			
	DBN		DBM		DBN		DBM		DBN		DBM	
	CD	PCD	CD	PCD	CD	PCD	CD	PCD	CD	PCD	CD	PCD
IHS	0.35	**0.25**	**0.45**	**0.46**	0.60	0.52	0.57	**0.55**	0.54	0.56	**0.82**	**0.53**
AIWPSO	2.21	2.28	2.64	2.41	3.39	2.68	3.89	3.62	4.31	4.73	5.67	4.28
CS	**0.30**	0.45	0.53	0.56	**0.49**	**0.45**	**0.44**	0.80	**0.47**	**0.29**	0.84	0.97
FA	0.75	1.49	1.81	1.06	1.37	1.30	1.95	2.41	2.23	2.22	2.52	1.29
BSA	1.28	1.31	0.98	1.21	1.12	0.71	2.67	1.61	1.48	1.43	2.65	3.74
JADE	1.00	1.63	0.79	0.88	1.93	1.76	2.12	1.81	1.34	1.69	3.17	2.34
CoBiDE	1.25	1.29	1.11	1.11	1.50	1.67	2.13	2.22	2.29	1.60	2.92	2.26

One can notice that, in general, IHS has been the fastest technique, followed by CS, which is somehow expected due to their updating mechanism. IHS evaluates a single solution each iteration, while CS evaluates a reduced number of solutions, given by the probability parameter p.

Table 5.9: Computational load (in hours) considering CalTech 101 Silhouettes dataset.

	1L				2L				3L			
	DBN		DBM		DBN		DBM		DBN		DBM	
	CD	PCD	CD	PCD	CD	PCD	CD	PCD	CD	PCD	CD	PCD
IHS	1.64	1.47	**1.81**	**1.62**	1.28	**1.37**	**1.84**	2.26	**1.13**	**1.06**	**1.98**	1.58
AIWPSO	8.87	9.44	10.54	11.50	9.41	7.79	12.30	12.34	11.17	7.95	13.50	13.82
CS	**1.55**	**1.01**	1.86	1.63	**0.93**	1.76	2.45	**2.17**	1.46	1.35	2.00	**0.80**
FA	3.38	5.27	6.03	3.00	6.25	3.27	7.26	2.62	3.58	8.08	6.55	8.83
BSA	6.40	5.08	6.55	8.30	6.04	5.60	9.19	8.42	4.23	4.53	7.95	9.90
JADE	8.24	4.31	9.22	7.90	7.71	4.10	11.15	7.40	8.25	4.57	9.43	8.29
CoBiDE	5.64	5.28	7.48	7.02	5.64	5.36	7.52	7.61	4.47	5.38	6.63	8.70

Likewise, one can expect that BSA, JADE, and CoBiDE to behave similarly regarding the computational load, since they are evolutionary-based techniques and the number of new solutions (the ones that employ mutation and crossover operations) to be evaluated depends upon a probability.

Table 5.10: Computational load (in hours) considering Semeion Handwritten Digit dataset.

	1L				2L				3L			
	DBN		DBM		DBN		DBM		DBN		DBM	
	CD	PCD	CD	PCD	CD	PCD	CD	PCD	CD	PCD	CD	PCD
IHS	**0.16**	0.19	**0.22**	**0.25**	**0.23**	**0.20**	**0.22**	0.31	**0.28**	0.28	**0.35**	**0.38**
AIWPSO	1.14	1.00	1.49	1.44	1.61	1.41	2.04	1.98	2.15	1.80	2.51	2.45
CS	0.26	**0.18**	0.31	0.26	0.26	0.24	0.31	**0.20**	**0.28**	**0.23**	0.40	**0.38**
FA	0.49	0.74	0.82	0.42	0.62	0.98	0.82	0.53	0.84	1.14	0.90	0.76
BSA	0.68	0.65	0.57	0.88	0.54	0.44	0.57	1.16	0.83	0.92	1.30	1.51
JADE	0.54	0.22	0.92	1.18	0.25	0.80	0.92	1.60	0.37	1.29	1.91	2.09
CoBiDE	0.71	0.58	0.74	0.91	0.49	0.89	0.96	1.03	0.68	1.01	1.52	1.30

One shortcoming of FA and AIWPSO concerns their computational burden since every agent in the swarm generates a new solution to be evaluated at each iteration. In fact, they are expected to present a slower convergence than IHS, which creates a single solution instead (i.e., it evaluates the fitness function only once per iteration). Such behavior makes them much faster than swarm-based techniques, but having a slower convergence as well.

5.5 Conclusions

In this work, we dealt with the problem of fine-tuning Deep Boltzmann Machines by means of meta-heuristic-driven optimization techniques to reconstruct binary images. The experimental results over three public datasets showed the validity in using such techniques to optimize DBMs when compared against a random search. Also, we showed DBMs can learn more accurate models than DBNs considering two out of three datasets. Moreover, we provided a detailed analysis of the similarity among each optimization technique using the Wilcoxon signed-rank test, as well the trade-off between the computational load demanded by each metaheuristic and its effectiveness.

Even though all techniques have obtained close results, we observed that evolutionary- and memory-based approaches might be more suitable for DBM/DBN fine-tuning hyperparameters. Since we are coping with hyperparameters that, under small intervals, do not influence the learning step (i.e., the reconstruction error), evolutionary operators and the process of creating new harmonies seem to introduce some sort of perturbation that moves possible solutions far apart from each other. In regard to future works, we aim to validate the proposed approach to reconstruct and also classify gray-scale images.

Chapter 6

Fine-Tuning Infinity Restricted Boltzmann Machines

The idea developed in Chapter 5 is now applied in the infinity Restricted Boltzmann Machine domain. This work was published in the 30th Conference on Graphics, Patterns, and Images (PASSOS; PAPA, 2017a).

6.1 Introduction

Restricted Boltzmann Machines are two-layered undirected graphical models that use a layer of hidden units to model the distribution over a set of inputs, thus compounding a generative stochastic neural network (LAROCHELLE et al., 2012; SCHMIDHUBER, 2015). RBMs have been highlighted in the scientific community over the last years, as well as some variants concerning deep learning models, e.g., Deep Belief Networks (HINTON; OSINDERO; TEH, 2006a) and Deep Boltzmann Machines (SALAKHUTDINOV; HINTON, 2012b), due to their outstanding results in a number of domains, such as human motion (TAYLOR; HINTON; ROWEIS, 2006), classification (LAROCHELLE et al., 2012), spam (SILVA et al., 2016) and anomaly detection (FIORE et al., 2013b), and collaborative filtering (SALAKHUTDINOV; MNIH; HINTON, 2007), just to cite a few.

However, one of the main concerns related to RBMs is associated with the number of hidden units, which is application-dependent and has a great impact in the final results. Montufar and Ay (MONTUFAR; AY, 2011) showed that an RBM with $2^{m-1}-1$ hidden units is a universal approximator, where m stands for the number of visible (input) units. Moreover, such a big representation may not be efficient in practice, which motivated researchers to study models that can automatically increase their capacity during learning.

Cotê and Larochelle (CÔTÉ; LAROCHELLE, 2016) proposed an extension of the RBM that does not require specifying the number of hidden units, and it can increase its capacity (i.e., number of hidden units) during training, hereinafter called infinite RBM. In this work, they also presented an extension of the RBM that is sensitive to the position of each unit in the hidden layer, named ordered Restricted Boltzmann Machines (oRBM), which can be interpreted as a special case of an implicit mixture of RBMs (NAIR; HINTON, 2009). This is achieved by adding new units in the hidden layer, where each one is trained gradually from left to right. Effectively, the model is growing in capacity during training until it reaches the maximum capacity defined previously.

Based on the aforementioned assumption, it turns out to be possible to devise a model where the number of hidden units increases automatically to a capacity that is similar to the universal approximator (i.e., when the number of hidden units tends to infinite), though being much smaller. Such model is possible due to the following assumptions: (i) that a finite number of hidden units has non-zero weights and biases, and (ii) the parametrization of the per-unit energy penalty ($\boldsymbol{\beta}$) ensures the infinite sums during probability computation will converge. Since the role of this energy penalty is to ensure the iRBM is properly defined only, the penalty imposed in the energy function can be compensated by the learned parameters (weight decay). Therefore, we can remove one of the RBM hyper-parameters from its project, i.e., the number of hidden units.

Despite dropping out the number of hidden units that is usually required beforehand, the iRBM still demands the selection of the remaining hyper-parameters, such as the learning rate, momentum and weight decay. Furthermore, its formulation incorporates the $\boldsymbol{\beta}$ hyper-parameter, which is less sensitive than the number of hidden units, but still requires its fine-tuning. In this chapter, we propose to find suitable hyper-parameters concerning the iRBM model by means of meta-heuristic optimization techniques, such as the Particle Swarm Optimization (PSO) (RODRIGUES et al., 2015), Bat Algorithm (BA) (YANG; GANDOMI, 2012), Cuckoo Search (CS), and the Firefly Algorithm (FA) (YANG, 2010). Although one can use any other optimization technique, we opted to use these ones mainly because they are well recognized in the literature, and they do not require computing derivatives as usually demanded by standard optimization techniques.

Recently, Papa et al. (PAPA et al., 2015b, 2015d; PAPA; SCHEIRER; COX, 2016a), Rosa et al. (ROSA et al., 2016b, 2015) and Rodrigues et al. (RODRIGUES; YANG; PAPA, 2016) demonstrated the robustness of these algorithms to the optimization of RBMs and DBNs, but to the best of our knowledge, we have not observed any work that dealt with the problem of iRBM fine-tuning by means of meta-heuristic techniques to date. Therefore, the main contributions of this chapter are twofold: (i) to foster the scientific literature concerning iRBMs, and (ii) to deal with the problem of iRBM hyper-parameter optimization. Additionally, we also considered both standard RBMs and oRBM for comparison purposes concerning the task of binary image reconstruction over two public datasets. The remainder of this chapter is organized as follows. Sections 6.2 and 6.3 present theoretical details about the iRBM and the proposed fine-tuning process, respectively. Section 6.4 discusses the methodology and Section 6.5 presents the experimental results. Finally, Section 6.6 states conclusions and future works.

6.2 Theoretical Background

The Theoretical Background regarding the Ordered Restricted Boltzmann Machines and Infinity Boltzmann machines are presented in Sections 2.4 and 2.5, respectively.

6.3 Infinity RBM Fine-Tuning as an Optimization Problem

The proposed approach requires the optimization of three hyper-parameters for both RBM and iRBM, and four parameters for the oRBM, as follows:

- RBM: the learning rate η, the L_1 regularization parameter, and the number of hidden units n;
- oRBM: the learning rate η, the L_1 regularization parameter, the number of hidden units n, and the energy penalty parameter $\boldsymbol{\beta} \in \Re^n$ for each hidden unit; and
- iRBM: the learning rate η, the L_1 regularization parameter, and the energy penalty parameter $\boldsymbol{\beta} \in \Re^n$ for each hidden unit.

Notice the regular learning rate (i.e., $\eta \in \Re$) is used to update the RBM, and the ADAGRAD stochastic gradient technique is used for both oRBM and iRBM (DUCHI; HAZAN; SINGER, 2011). In this case, we have a per-dimension learning rate method, i.e., $\boldsymbol{\eta} \in \Re^n$, with $\varepsilon = 10^{-6}$ (CÔTÉ; LAROCHELLE, 2016). This latter parameter stands for a small number to avoid numerical instabilities. Cotê and Larochelle (CÔTÉ; LAROCHELLE,

2016) claims that one can throw away the parameter n, thus replacing the RBM model by the iRBM one. However, $\boldsymbol{\beta}$ is still a hyper-parameter to be optimized[1].

Figure 7.3 depicts the proposed approach to optimize the RBM, oRBM and iRBM models. Roughly speaking, the idea is to initialize all decision variables techniques at random, and then the optimization algorithm takes place. In this work, we used the following ranges concerning the parameters: $n \in [5,500]$, $\boldsymbol{\eta} \in [0.01,0.5]$, $\boldsymbol{\beta} \in [0.01,1.5]$ and $\text{L}_1 \in [0.00001,0.01]$.

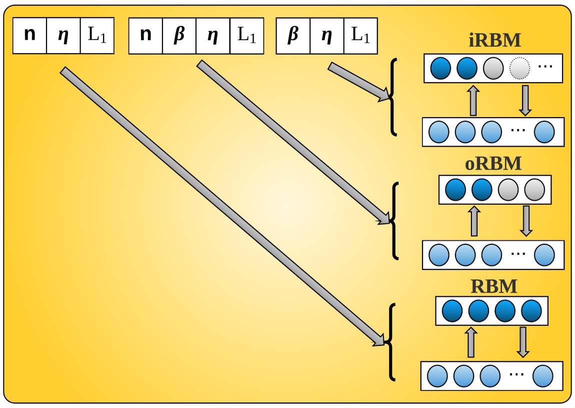


Figure 6.1: Proposed approach to model the fine-tuning problem as an optimization task.

In order to fulfill the requirements of any optimization technique, one shall design a fitness function to guide the search into the best solutions. In this chapter, we used the average negative log-likelihood (NLL) over the training set considering the task of binary image reconstruction as the fitness function. Therefore, we adopted the very same methodology used by (CÔTÉ; LAROCHELLE, 2016), but presenting the mean results obtained over 20 runs in order to provide a statistical comparison[2].

In short, the optimization technique selects the set of hyper-parameters that minimize the NLL over the training set considering a dataset of binary images as an input to the model. After learning the hyper-parameters, one can proceed to the reconstruction step concerning the testing images, whose effectiveness is assessed by the NLL method.

[1] Notice the regularization parameter $\boldsymbol{\beta}$ is way less sensitive than the number of hidden units **n**

[2] Notice the work by Cotê and Larochelle (CÔTÉ; LAROCHELLE, 2016) presents the best result over all runs only.

6.4 Methodology

In this section, we present the methodology employed to evaluate the proposed approach, optimization techniques, datasets, and the experimental setup.

6.4.1 Optimization Techniques

This work employs four metaheuristic techniques to the task of iRBM fine-tuning, i.e., PSO, BA, CS, and FA, presented in Section 2.8.1.

Table 6.1 presents the parameters used for each aforementioned optimization technique, where five agents (initial solutions) were used for all optimization techniques during 20 iterations for convergence purposes[3]. In regard to PSO, w stands for the inertia weight, and c_1 and c_2 control the step size towards the best local and global solutions, respectively. With respect to BA, f_{min} and f_{max} bound the minimum and maximum frequency values, and A and r denote the the loudness and pulse rate values, respectively. Parameters φ and τ are used to avoid the technique getting trapped from local optima. FA uses μ and γ, which stand for a random perturbation and the light absorption coefficient, respectively. Variable ς denotes the attractiveness of each firefly. Finally, CS uses Γ to compute the Lévy distribution, ζ for the switch probability (i.e., the probability of replacing the worst nests by new ones), and s for the step size.

Table 6.1: Parameter configuration for each optimization technique.

Technique	Parameters
PSO	$c_1 = 1.7$, $c_2 = 1.7$, $w = 0.7$
BA	$\varphi = 0.9$, $\tau = 0.9$ $f_{min} = 0$, $f_{max} = 100$ $A = 1.5$, $r = 0.5$
CS	$\Gamma = 1.5$, $\zeta = 0.25$, $s = 0.8$
FA	$\gamma = 1$, $\varsigma = 1$, $\mu = 0.2$

6.4.2 Datasets

We propose to evaluate the behavior of different optimization techniques to fine-tune RBM/oRBM/iRBM in the context of binary image reconstruction using two public datasets, i.e., MNIST and CalTech 101 Silhouettes Data Set, as described in Section 2.9.

[3] Notice these parameters were set empirically.

6.4.3 Experimental Setup

This work employs a cross-validation procedure with 20 runs in order to provide a statistical analysis by means of the Wilcoxon signed-rank test with significance of 0.05 (WILCOXON, 1945). The training step is conducted with $5,000$ epochs, with an Annealed Importance Sampling (AIS) evaluation every $1,000$ epochs to keep the best NLL approximation (SALAKHUTDINOV; MURRAY, 2008).

For the learning procedure, we used ten Gibbs sampling steps with mini-batches of size 64. In addition, we also considered two learning algorithms: Contrastive Divergence (CD) (HINTON, 2002) and Persistent Contrastive Divergence (PCD) (TIELEMAN, 2008a). Furthermore, all NLL results were obtained by estimating the log-partition function using AIS with $100,000$ intermediate distributions and $5,000$ chains.

Finally, the codes used to reproduce the experiments of the chapter are available on GitHub[45]. The experiments were conducted using a Ubuntu 16.04 Linux machine with 16Gb of RAM running an Intel Core™$i7-4790$ with a frequency of 3.60 GHz and a GPU GeForce® GTX970 with 4GB. The source-codes run on top of Python with Theano (BASTIEN et al., 2012) and C for the RBM/oRBM/iRBM and optimization approaches, respectively.

6.5 Experimental Results

In this section, we present the experimental results concerning iRBM, oRBM and RBM hyper-parameter optimization in the task of binary image reconstruction. Additionally, all techniques are compared using two different learning algorithms, i.e., Contrastive Divergence and Persistent Contrastive Divergence. In order to validate the proposed approach, we also considered a random search (RS) as a baseline for hyper-parameter optimization.

Table 6.2 presents the averaged NLL results concerning the MNIST dataset, being the values in bold the best results considering the Wilcoxon signed-rank. Although RBM achieved the best results using PSO, both iRBM and the oRBM obtained similar results according to the Wilcoxon signed-rank test, using BA and FA techniques, respectively. This behavior is expected as it matches the results obtained in (CÔTÉ; LAROCHELLE, 2016), which concluded that RBMs are still more accurate, but at the price of having a more sensitive parameter to be set (i.e., the number of hidden units). Also, one can clearly

[4] iRBM: http://github.com/MarcCote/iRBM

[5] LibOPT (PAPA et al., 2017b): https://github.com/jppbsi/LibOPT

observe the meta-heuristic techniques are able to achieve much more accurate results than the baseline provided by the random search.

Table 6.2: Average NLL values considering MNIST dataset.

	RBM		oRBM		iRBM	
	CD	PCD	CD	PCD	CD	PCD
RS	192.84±24.31	195.25±18.61	163.80±28.44	153.16±16.39	160.94±20.54	153.91±18.53
BA	188.07±66.12	220.75±24.35	179.15±37.24	166.13±46.36	184.18±42.05	**165.83±85.07**
CS	154.35±20.82	178.21±20.13	161.33±18.96	156.39±20.71	**149.40±8.34**	**150.71±23.32**
FA	**125.39±39.59**	243.55±42.11	156.50±23.11	**133.82±40.26**	206.24±18.78	171.92±153.11
PSO	**124.60±44.96**	216.94±41.43	171.24±40.32	164.13±41.18	208.10±37.19	179.25±72.79

It is worth mentioning that PCD has provided better results only for oRBM and iRBM. As a matter of fact, it is arguable that PCD may provide more accurate results than CD, since it does not restart the Markov chain when a new training sample is presented to the network, but it uses the last sampled data from the previous training sample to initiate the chain. However, such behavior was not observed for RBMs, and it quite reasonable to assume that PCD learning can really work well for iRBM and oRBM, since such models may not achieve results so accurate than standard RBMs due to their smaller hidden layers, which means they may have a poorer capacity for learning.

Figure 6.2 depicts some testing images reconstructed by RBM, oRBM and iRBM. One can observe the images are better reconstructed by RBM, with less noise as well, thus confirming the numerical results presented in Table 6.2. Additionally, one can refer to the network's weights, as displayed in Figure 6.3, in which a more variety of filters can be observed for standard RBM. Such behavior evidences a greater capacity for learning, which can also be observed for oRBM as well.

Finally, we also considered the computational load of each technique for comparison purposes, as presented in Table 6.3. The fastest optimization technique has been the Cuckoo Search for RBM, oRBM and iRBM, being RBM the fastest of all since its formulation is less complex than oRBM and iRBM.

Table 6.4 presents the average NLL results concerning Caltech 101 Silhouettes dataset. In this case, iRBM achieved the best results with all meta-heuristic techniques using CD for learning, except for CS. Additionally, oRBM obtained the best results with the FA algorithm. Actually, iRBM trained with CD and optimized by FA achieved the best result so far. Such results are pretty much interesting, since Caltech dataset poses a greater challenge than MNIST (greater NLL values). Although oRBM and iRBM were

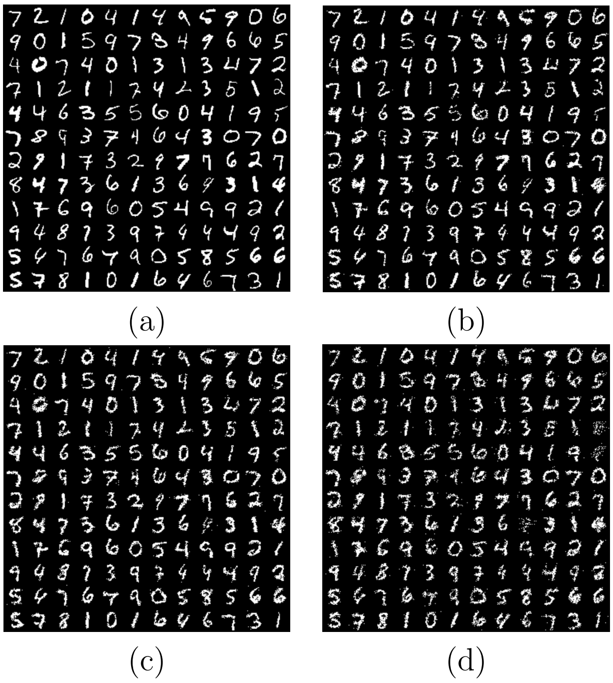


Figure 6.2: Random (a) MNIST testing images reconstructed by (b) RBM fine-tuned with FA and trained with CD, (c) oRBM fine-tuned with FA and trained with PCD, and (d) iRBM fine-tuned with CS and trained with CD.

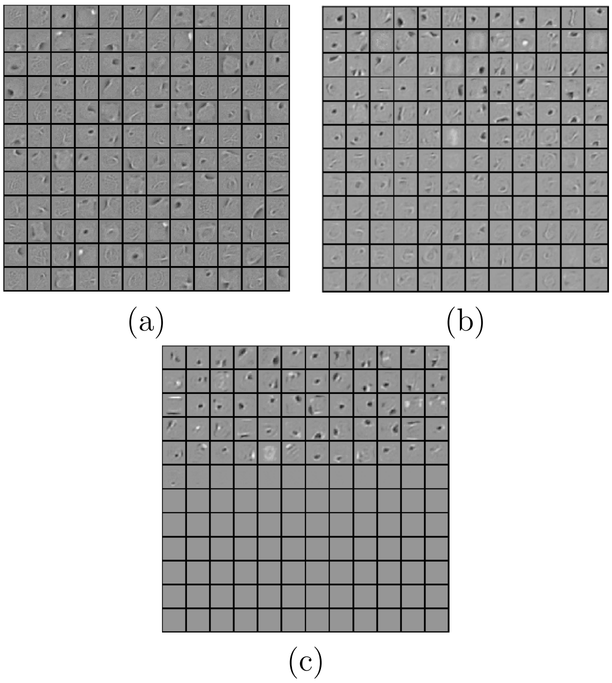


Figure 6.3: "The network's mind" considering MNIST dataset: comparing the filters obtained by (a) RBM fine-tuned with FA and trained with CD, (b) oRBM fine-tuned with FA and trained with PCD, and (c) iRBM fine-tuned with CS and trained with CD.

not proposed to outperform RBM, one can observe that more accurate models can be obtained by avoiding complex architectures. As a matter of fact, RBMs may be more

Table 6.3: Average time (minutes) for learning hyper-parameters considering MNIST dataset.

	RBM		oRBM		iRBM	
	CD	PCD	CD	PCD	CD	PCD
RS	13.07	14.10	20.26	19.72	16.64	19.01
BA	68.34	65.73	121.36	122.48	111.00	300.96
CS	**49.49**	50.29	92.36	92.84	100.56	349.06
FA	63.37	64.97	126.24	120.36	120.03	411.30
PSO	68.40	68.80	108.67	127.56	143.02	234.91

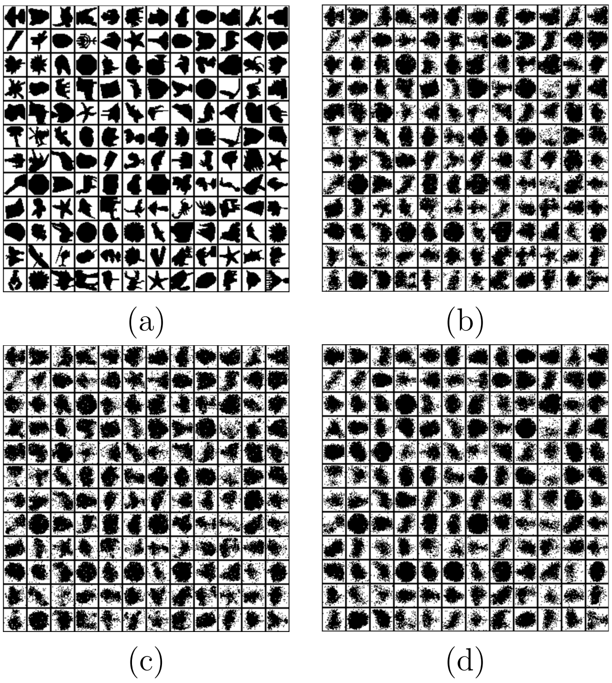
prone to overfit when one does not choose the number of hidden units properly.

Table 6.4: Average NLL values considering Caltech 101 Silhouettes dataset.

	RBM		oRBM		iRBM	
	CD	PCD	CD	PCD	CD	PCD
RS	384.30±29.94	432.38±140.15	267.42±28.39	386.03±94.26	274.36±33.99	424.30±187.62
BA	292.08±77.24	609.27±170.72	243.72±24.93	458.95±216.99	**229.32±32.14**	593.33±229.98
CS	349.60±47.13	455.83±104.28	267.82±29.60	448.20±126.03	255.15±18.67	579.29±254.97
FA	279.88±57.13	629.06±170.37	**237.85±23.63**	420.77±163.16	**218.36±28.54**	486.86±110.73
PSO	315.42±85.29	599.11±140.47	240.40±26.29	411.74±66.69	**237.83±37.83**	554.60±254.15

Figure 6.4 depicts some testing images reconstructed by RBM, oRBM and iRBM concerning Caltech 101 Silhouettes dataset. In this case, it is difficult to visualize a clear difference among the techniques. Also, Caltech dataset has much more classes than MNIST, thus resulting in poorer reconstructed images. The weights of the networks are displayed in Figure 6.5, in which a richer representation in the iRBM's weights can be observed. One can notice a considerable number of full-gray patches, which means they did not learn so much information from the training step.

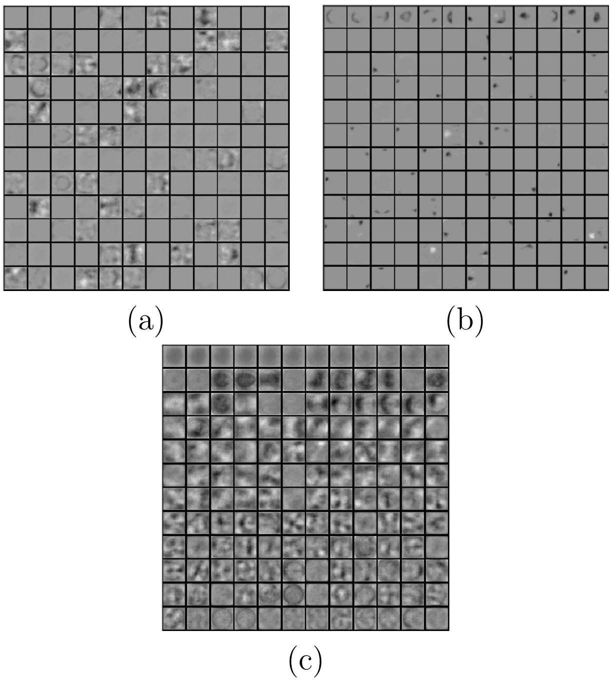
Table 6.5 presents the average computational load concerning Caltech 101 Silhouettes dataset. Considering the worst case, iRBM was around 14.75 times slower than RBM, which showed to be the fastest approach once again. As a matter of fact, both oRBM and iRBM tend to be faster than RBMs for the reconstruction step, since one has less hidden units for computation purposes.



(a) (b)

(c) (d)

Figure 6.4: Random (a) Caltech 101 Silhouettes testing images reconstructed by (b) RBM fine-tuned with FA and trained with CD, (c) oRBM fine-tuned with FA and trained with CD, and (d) iRBM fine-tuned with CS and trained with CD.



(a) (b)

(c)

Figure 6.5: "The network's mind" considering Caltech 101 Silhouettes dataset: comparing the filters obtained by (a) RBM fine-tuned with FA and trained with CD, (b) oRBM fine-tuned with FA and trained with CD, and (c) iRBM fine-tuned with FA and trained with CD.

6.6 Conclusions and Future Works

This chapter addressed the problem of iRBM fine-tuning by means of meta-heuristic techniques. Ordered and Infinity RBMs are very recent models that avoid choosing the

Table 6.5: Average time (minutes) for learning hyper-parameters considering Caltech 101 Silhouettes dataset.

	RBM		oRBM		iRBM	
	CD	PCD	CD	PCD	CD	PCD
RS	13.29	14.10	19.07	20.18	14.81	18.54
BA	20.20	17.68	74.44	54.45	108.00	128.25
CS	16.35	**15.61**	51.49	40.71	67.51	66.61
FA	18.52	17.71	72.42	56.66	133.48	128.63
PSO	19.05	18.66	74.50	50.61	97.79	230.21

number of hidden units, but at the price of introducing one more variable related to the penalty in adding one more hidden unit to the learning process. However, such parameter is way less sensitive to the number of hidden units, thus requiring less effort by the user and setting up the model.

Experiments over two public datasets concerning the task of binary image reconstruction using four meta-heuristic techniques showed they are suitable for hyper-parameter fine-tuning, being the Cuckoo Search the fastest technique, and FA one of the most accurate.

In regard to future works, we intend to investigate the suitability of deep versions of both oRBMs and iRBMs, as well as their fine-tuning by means of meta-heuristic techniques.

Chapter 7

Barrett's Esophagus Analysis Using Infinity Restricted Boltzmann Machines

This chapter presents the paper entitled Barrett's Esophagus Analysis Using Infinity Restricted Boltzmann Machines, submitted under an invitation from the Journal of Visual Communication and Image Representation as an extension from the idea presented in (PASSOS; PAPA, 2017a) applied to medical issues.

7.1 Introduction

The incidence of adenocarcinoma in patients with Barrett's esophagus (BE) faced a major increase in western populations in the last 10 years, explained by risk factors such as obesity and smoking (LAGERGREN; LAGERGREN, 2010; DENT, 2011; LEPAGE; RACHET; JOOSTE, 2008), and an expectation to rise in the next years. The bad prognosis of patients suffering from esophageal adenocarcinoma is related to its late diagnosis. Despite the dangerousness of the disease, when detected at the early stages the dysplastic tissue can be treated achieving very high rates of the disease remission (93% after 10 years, still presenting 5% of morbidity and 0% of mortality) (DENT, 2011; SHARMA et al., 2016; PHOA et al., 2016). Endoscopic resection (mucosal resection and submucosal dissection) and ablation techniques (radiofrequency ablation and cryoablation) appear to be promising methods developed for the management of BE, with the potential to reduce the adenocarcinoma risk in patients with dysplasia. However, limitations in the current methods for monitoring and evaluating the BE level highlighted the necessity to the design of additional tools to improve the detection of dysplasia (SHAHEEN et al., 2009; JOHNSON et al., 2005; OVERHOLT; PANJEHPOUR; HALBERG, 2003).

Many efforts were considered in the last years regarding machine learning and computer-aided diagnosis. Van der Sommen (SOMMEN et al., 2016), for instance, designed a system capable of automatically extract features for detecting and delineating early neoplastic lesions in Barrett's esophagus. Other works (JR. et al., 2017b; HASSAN; HAQUE, 2015) aimed to use features extracted from endoscopic images for the classification of Barrett's esophagus and adenocarcinoma. Furthermore, Mendel et al. (MENDEL et al., 2017) proposed a deep learning approach based on Convolutional Neural Networks in the context of BE analysis. Recently, Souza et al. (JR. et al., 2017a) conducted a study in which two approaches were introduced to distinguish between BE and adenocarcinoma: (i) the Optimum-Path Forest (OPF) (PAPA; FALCÃO; SUZUKI, 2009; PAPA et al., 2012) classifier; and (ii) the use of Bag-of-Visual-Words (BoVW) (CSURKA et al., 2004; PENG et al., 2016) using points-of-interest (PoIs) extracted from endoscopic images using Speed-Up Robust Features (SURF) (BAY et al., 2008) and Scale Invariant Feature Transform (SIFT) (LOWE, 2004) techniques (JR. et al., 2018) for the feature vector extraction.

Restricted Boltzmann Machines (RBMs) are nondeterministic neural networks composed of two layers of neurons, i.e., visible and hidden, whose main idea is to produce a probabilistic representation of a given input data in the hidden layer, such that the network is capable of reconstructing the data in the visible layer (LAROCHELLE et al., 2012; SCHMIDHUBER, 2015). The process is conducted using the minimization of the system's energy, analogous to the Maxwell-Boltzmann distribution law of thermodynamics. RBMs have been highlighted in the scientific community over the last years, as well as some variants concerning deep learning models, e.g., Deep Belief Networks (DBNs) (HINTON; OSINDERO; TEH, 2006a) and Deep Boltzmann Machines (DBMs) (SALAKHUTDINOV; HINTON, 2012b), due to their outstanding results in a number of domains, such as human motion (TAYLOR; HINTON; ROWEIS, 2006), classification (LAROCHELLE et al., 2012), spam (SILVA et al., 2016), anomaly detection (FIORE et al., 2013b), and collaborative filtering (SALAKHUTDINOV; MNIH; HINTON, 2007), just to cite a few.

However, one of the main concerns related to RBMs is associated with the number of hidden units, which is application-dependent and has a great impact on the final results. Montufar and Ay (MONTUFAR; AY, 2011) showed that an RBM with $2^{m-1}-1$ hidden units is a universal approximator, where m stands for the number of visible (input) units. Moreover, such a large scale representation may not be efficient in practice, which motivated researchers to study models that can automatically increase their capacity during learning.

Cotê and Larochelle (CÔTÉ; LAROCHELLE, 2016) proposed an extension of the RBM that does not require specifying the number of hidden units, and it can increase its capacity (i.e., number of hidden units) during training, hereinafter called infinite RBM (iRBM). The learning achieved by adding new units in the hidden layer, where each one is trained gradually from left to right. Effectively, the number of hidden units increases automatically to a capacity that is similar to the universal approximator (i.e., when the number of hidden units tends to infinite), though being much smaller. Such model is possible due to the following assumptions: (i) that a finite number of hidden units has non-zero weights and biases, and (ii) the parametrization of the per-unit energy penalty ($\boldsymbol{\beta}$) ensures the infinite sums during probability computation will converge. Since the role of this energy penalty is only to ensure the iRBM is properly defined, the penalty imposed in the energy function can be compensated by the learned parameters (i.e., weight decay). Therefore, we can remove one hyper-parameter from the project with the cost of introduction of a less sensible one, i.e., the number of hidden units is removed and an extra parameter is introduced in the model.

Despite the advantage that iRBM brought by removing the need to properly select the number of hidden neurons, it also came up with a shortcoming related to the slow convergence. Peng et al. (PENG; GAO; LI, 2017) attribute the problem to the initial correlation that is given by the ordering effect present in the iRBM, and proposed a solution by adding a probability of flipping the position of some neurons in the hidden layer while training, avoiding the dependency among each other. Additionally, they also proposed a mechanism to use the iRBM not only for binary image reconstruction but also for discriminative tasks, employing a “one-hot”vector representation of the sample's label together with the feature vector while training the model for further classification of the test set.

Regardless dropping out the hidden units that are usually required beforehand, iRBM still demands the selection of the remaining hyper-parameters, such as the learning rate, the weight decay, and the $\boldsymbol{\beta}$ hyper-parameter, which despite less sensitive than the number of hidden units, still requires a proper fine-tuning. To deal with this problem, Passos et al. (PASSOS; PAPA, 2017a) proposed to employ meta-heuristic optimization techniques to fine-tune the aforementioned hyper-parameters regarding binary image reconstruction since it has provided suitable results (PAPA et al., 2015b, 2015d; PAPA; SCHEIRER; COX, 2016a; ROSA et al., 2016b, 2015; RODRIGUES; YANG; PAPA, 2016). However, as far as we know, such techniques were never used to fine-tune iRBMs regarding classification tasks. In this paper, we propose to find suitable hyper-parameters concerning the discriminative iRBM model using eight meta-heuristic techniques: Particle Swarm Optimization

(PSO) (RODRIGUES et al., 2015), Bat Algorithm (BA) (YANG; GANDOMI, 2012), Cuckoo Search (CS) (YANG; DEB, 2010), Brain Storm Optimization (SHI, 2011), Firefly Algorithm (FA) (YANG, 2010), and the Harmony Search (HS) (GEEM, 2009). Although one can use any other optimization technique, we opted to use these mainly because they are well recognized in the literature, and they do not require computing derivatives as usually demanded by standard optimization techniques.

In this paper, we also introduce the infinity Restricted Boltzmann Machines in the context of automatic classification of Barrett's esophagus using information extracted with SURF and SIFT techniques in the "MICCAI 2015 Endovis Challenge"dataset. The experiments performed a comparison of the aforementioned meta-heuristic optimization techniques regarding iRBM meta-parameter fine-tuning to the task of Barrett's esophagus classification. Additionally, we also considered both linear and Radial Basis Function (RBF) Support Vector Machines (SVM) for comparison purposes.

Therefore, the main contributions of this paper are fourfold: (i) to introduce iRBM in the context of Barrett's esophagus recognition, (ii) to promote the scientific literature concerning iRBMs, (iii) to foster the scientific literature concerning Barrett's esophagus, and (iv) to deal with the problem of iRBM hyper-parameter optimization concerning discriminative tasks. The remainder of this paper is organized as follows. Sections 7.2 and 7.3 present theoretical background and the proposed fine-tuning process, respectively. Section 7.4 discusses the methodology and Section 7.5 presents the experimental results. Finally, Section 7.6 states conclusions and future works.

7.2 Theoretical Background

In this section, we briefly explain the theoretical background related to the discriminative Infinity Restricted Boltzmann Machines and the dinamic training strategy.

7.2.1 Discriminative Infinity Restricted Boltzmann Machines

Larochelle and Bengio introduced a discriminative version of the RBM (LAROCHELLE; BENGIO, 2008) to the task of classification, and Peng et al. (PENG; GAO; LI, 2017) adapted the idea to the iRBM domain. In order to couple the labels in the formulation, the energy function is redefined as follows:

$$\mathbf{E}(\mathbf{v},\mathbf{h},y,z) = -\sum_{i=1}^{m} a_i v_i - \mathbf{e}_y d - \sum_{j=1}^{z} b_j h_j - \sum_{j=1}^{z} h_j \left(\sum_{i=1}^{m} (v_i w_{ij}) + \mathbf{e}_y u_{yj} + \beta_j \right), \tag{7.1}$$

where d is the bias of the label vector, $\mathbf{e}_y = \left(1_{y=1}\right)_{i=1}^{C}$ stands for the so-called "one-hot"representation of label $y \in \{1,2,\ldots,C\}$, and u_{yj} is the element from matrix $\mathbf{U}$ connecting the jth hidden unit to $\mathbf{e}_y$.

The distribution over y given the energy function (Eq. 7.1) is given by:

$$P(h_j = 1|\mathbf{v},z) \quad = \quad \begin{cases} \phi\left(\sum_{i=1}^{m} w_{ij} v_i + u_{yj}\mathbf{e}_y + b_j\right) & \text{if } j \leq z \\ 0 & \text{otherwise,} \end{cases} \tag{7.2}$$

and

$$P(y|\mathbf{h},z) = exp\left(\sum_{j=1}^{z} h_j \left(\mathbf{e}_y u_{yj}\right) + d_y\right) \Bigg/ \sum_{y'} exp\left(\sum_{j=1}^{z} h_j \left(\mathbf{e}_{y'} u_{y'j}\right) + d_{y'}\right), \tag{7.3}$$

having $P(v_i = 1|\mathbf{h},z)$ defined exactly the same as in Equation 2.19. Figure 7.1 depicts the Discriminative iRBM.

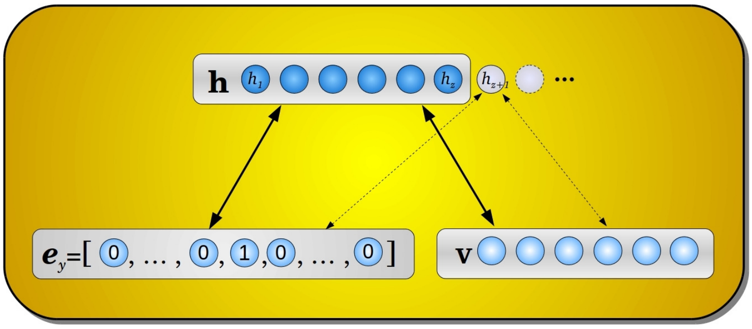


Figure 7.1: Discriminative iRBM. Both visible (v) and label ($\mathbf{e}_y$) layers are employed for training the model. A new hidden unit $\mathbf{h}_{z+1}$ is introduced in the model for learning purposes.

7.2.2 Dynamic Training Strategy

Despite the advantages achieved using iRBM, such as the absence of a hyper-parameter to be fine-tuned, i.e., the number of units in the hidden layer, it presents a shortcoming related to a slow convergence while training the network. The explanation concerns the time required by the filters to diverge from each other given an initial correlation imposed by the ordering effect intrinsic to iRBMs, where each newly added hidden unit suffers from the influence of the previously added ones, learning the features jointly and not by itself.

To cope with the issue, Peng et al. (PENG; GAO; LI, 2017) proposed to use the approximated gradient descent algorithm together with the dynamic training strategy, which assumes that changing the order of the hidden units at each gradient descent step and jointly training iRBMs with all possible orders it is possible to alleviate the bias inherited from the ordering effect. The model employs a variable Q_t, which controls the proportion of units regrouped at step t. Additionaly, $\tilde{\mathbf{o}}$ stands for the vector of indexes to be permuted given a probability distribution. The process is illustrated in Figure 7.2.

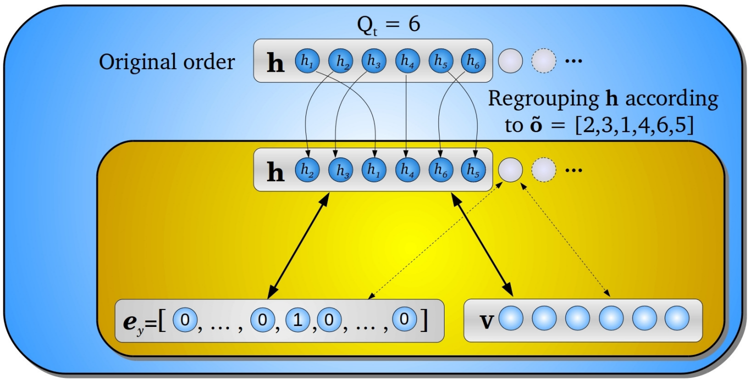

$Q_t = 6$
Original order
$\mathbf{h}$ h_1 h_2 h_3 h_4 h_5 h_6 ...
Regrouping $\mathbf{h}$ according to $\tilde{\mathbf{o}}$ = [2,3,1,4,6,5]
$\mathbf{h}$ h_2 h_3 h_1 h_4 h_6 h_5 ...
e_y=[0, ... , 0, 1, 0, ... , 0]
$\mathbf{v}$

Figure 7.2: Dynamic training strategy proposed by Peng et al. (PENG; GAO; LI, 2017), where Q_t Hidden units are permuted at time step t accordingly to the indexes defined in $\tilde{\mathbf{o}}$.

7.3 Infinity RBM Fine-Tuning as an Optimization Problem

The proposed approach requires the optimization of three hyper-parameters of the iRBM: (i) the learning rate η, (ii) the weight decay λ regularization parameter, and (iii) the β parameter. Notice the ADAGRAD stochastic gradient technique (DUCHI; HAZAN; SINGER, 2011) is employed as the learning rate. In this case, we have a per-dimension learning rate method, i.e., $\boldsymbol{\eta} \in \Re^n$, with $\varepsilon = 10^{-6}$ (CÔTÉ; LAROCHELLE, 2016). In a nutshell, meta-parameters η and β can be interpreted as an n-sized vector, where n stands for the current number of hidden units. This latter parameter stands for a small number to avoid numerical instabilities. Cotê and Larochelle (CÔTÉ; LAROCHELLE, 2016) claim that one can throw away the parameter n, thus replacing the RBM model by the iRBM one. However, $\boldsymbol{\beta}$ is still a hyper-parameter to be optimized. Notice the regularization parameter $\boldsymbol{\beta}$ is way less sensitive than the number of hidden units $\mathbf{n}$.

Figure 7.3 depicts the proposed approach to optimize the iRBM model, where the idea is to initialize all decision variables at random, and then the optimization algorithm takes place. In this work, we used the following ranges concerning the parameters: $\boldsymbol{\eta} \in [0{,}0001, 0{,}5]$, $\boldsymbol{\beta} \in [0{,}01, 1{,}5]$ and $\boldsymbol{\lambda} \in [0{,}00001, 0{,}01]$.

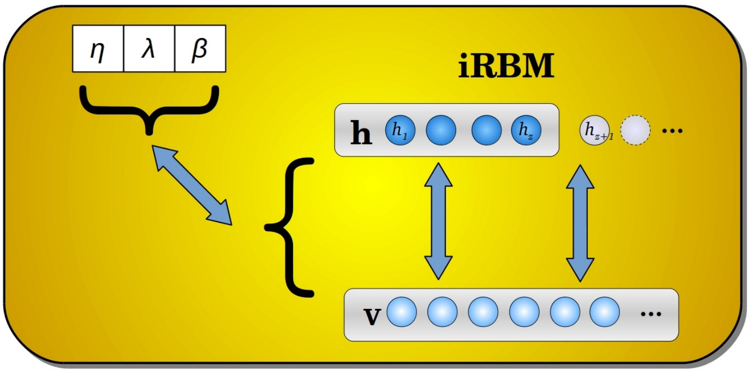


Figure 7.3: Proposed approach to model the iRBM fine-tuning problem as an optimization task.

In order to fulfill the requirements of any optimization technique, one shall design a fitness function to guide the search into for best solutions. In this paper, we used the average accuracy of the training set considering the task of classification as the fitness function. Furthermore, we present the mean results obtained over 20 runs to provide a statistical comparison.

In short, the optimization technique selects the set of hyper-parameters that maximizes the classification accuracy over the training set considering a set of features extracted from endoscopic images using BoVW over SIFT and SURF features as an input to the model. After learning the hyper-parameters, one can proceed to the classification step concerning the testing samples. Regarding this work, the following approaches are conducted: (i) the set of meta-parameters that best fits the model is selected using the validation set over a reduced number of 150 epochs for convergence purposes, and (ii) afterwards, the selected set of best meta-parameters are used to train the network using 1,500 epochs and to perform the classification over the test set.

7.4 Methodology

In this section, we present the methodology employed to evaluate the optimization techniques, the dataset, and the experimental setup.

7.4.1 Optimization Techniques

This work employs six metaheuristic techniques to the task of iRBM fine-tuning, i.e., PSO, BA, CS, BSO, HS, and FA, presented in Section 2.8.1.

Table 7.1 presents the parameters used for each aforementioned optimization technique, where five agents (initial solutions) were used for all optimization techniques during 10 iterations for convergence purposes. Notice these parameters were set empirically. In regard to PSO, w stands for the inertia weight, and c_1 and c_2 control the step size towards the best local and global solutions, respectively. With respect to BA, f_{min} and f_{max} bound the minimum and maximum frequency values, and A and r denote the loudness and pulse rate values, respectively. Regarding BSO, p_{gen} defines a probability whether a new solution will be generated by one or two other individuals, k stands for the number of clusters composed of similar ideas, and $p_{oneCluster}$ and $p_{twoCluster}$ stand for the probability of creating a new solution based on only one or two clusters, respectively. Parameters φ and τ are used to avoid the technique getting trapped from local optima. FA uses μ and γ, which stand for a random perturbation and the light absorption coefficient, respectively. Variable ς denotes the attractiveness of each firefly. Furthermore, Harmony Memory Considering Rate (HMRC) and Pitch Adjusting Rate (PAR) are used by HS for responsible creating new solutions based on previous experience of the music player and applying some disruption to the created solution in order to avoid local optima, respectively. Finally, CS uses Γ to compute the Lévy distribution, ζ for the switching probability (i.e., the probability of replacing the worst nests by new ones), and s for the step size.

7.4.2 Datasets

The information (i.e., features) were extracted from a dataset of images from Barrett's esophagus and adenocarcinoma called "MICCAI 2015", which was provided at the "MICCAI 2015 EndoVis Challenge"[1]. Such dataset is composed of 100 endoscopic images of the lower esophagus from 39 individuals, 22 presenting esophageal adenocarcinoma and

[1] https://endovissub-barrett.grand-challenge.org/

Table 7.1: Parameter configuration for each optimization technique.

Technique	Parameters
BA	$r = 0.5$, $A = 1.5$ $f_{min} = 0$, $f_{max} = 100$ $\varphi = 0.9$, $\tau = 0.9$
BSO	$p_{gen} = 0.4$, $k = 2$ $p_{oneCluster} = 0.8$, $p_{twoCluster} = 0.5$
CS	$\Gamma = 1.5$, $\zeta = 0.25$, $s = 0.8$
FA	$\gamma = 1$, $\varsigma = 1$, $\mu = 0.2$
HS	$HMCR = 0.7$, $\rho = 10$, $PAR = 0.7$
PSO	$c_1 = 1.7$, $c_2 = 1.7$, $w = 0.7$

17 diagnosed with early-stage Barrett's esophagus. For each patient, several endoscopic images were made available, ranging from one to eight. A total of 50 images showing cancerous tissue areas and 50 images showing dysplasia without cancer compose the dataset. The injured tissue observed in the cancerous images have been delineated by five different endoscopy experts. Figure 7.4 shows some dataset samples and their respective delineation performed by the experts.

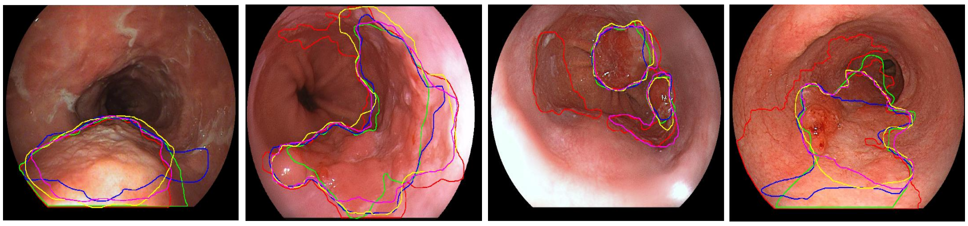


Figure 7.4: Some samples from the Barrett's Endovis 2015 Challenge (JR. et al., 2017b).

7.4.3 Experimental Setup

This work employs a cross-validation procedure with 20 runs to provide a statistical analysis using the Wilcoxon signed-rank test with a significance of 0.05 (WILCOXON, 1945). Regarding the task of meta-parameter optimization, we conducted the experiments over 150 epochs to find the meta-parameters that lead to the best classification accuracies regarding the validation set. Finally, the network was trained once again over 1,500 epochs using the best parameters found by each meta-heuristic technique to classify the testing set. The learning procedure was conducted using Persistent Contrastive Divergence (PCD) (TIELEMAN, 2008a) using three Gibbs sampling steps with mini-batches of size 5.

Finally, the codes used to reproduce the experiments are available on GitHub[2,3]. The experiments were conducted using an Ubuntu 16.04 Linux machine with 8Gb of RAM running an Intel Core™i5 − 2410*M* with a frequency of 2.30 GHz and a GPU GeForce® GT540*M* with 2GB. The source-codes run on top of Matlab and C for the iRBM and optimization approaches, respectively.

7.5 Experiments

This section describes the experiments as follows: Section 7.5.1 presents the image feature extraction using points-of-interest together with a Bag-of-Visual-Words schema, and Section 7.5.2 discusses the optimization steps as well as the time consumption regarding the role optimization process. Sections 7.5.3 and 7.5.4 present the procedures adopted while training and testing the model, respectively.

7.5.1 Feature Extraction

The points-of-interest were calculated using the Speed-Up Robust Features and the Scale-Invariant Feature Transform techniques, and then the feature vectors were calculated using Bag-of-Visual-Words. The SURF technique ensures scale and spatial invariance, seeking for maxima of the determinant of the Hessian matrix, demarcating specific key points which are explored in their local neighborhood resulting in a feature vector of size 64. The SIFT algorithm operates on image regions calculating features that are invariant to scaling and rotation. It seeks for the scale-space extrema detection evaluating the image scales (difference-of-Gaussian function) providing feature vectors of size 128. Finally, the BoVW technique uses points-of-interest from a set of reference images to generate a visual dictionary that is employed in the training and testing phases. For this work, we considered dictionaries with two different sizes: 500 and $1,000$ (JR. et al., 2017a). In order to compose such dictionaries, two well-known techniques were considered: (i) *k*-means and (ii) random selection. Figure 7.5 illustrates the feature vector calculation for the experiments.

[2] iRBM: https://github.com/Boltzxuann/RP-iRBM

[3] LibOPT (PAPA et al., 2017b): https://github.com/jppbsi/LibOPT

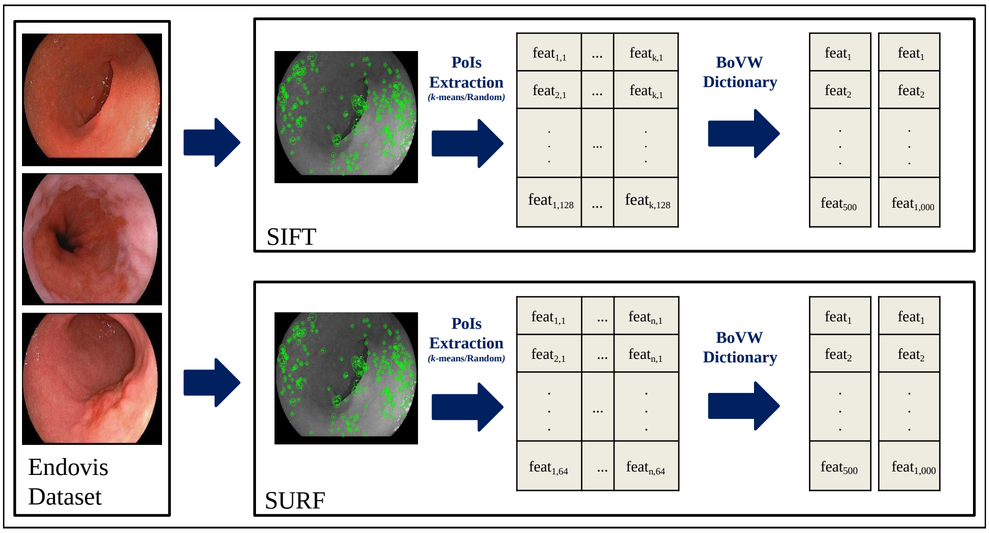

Endovis Dataset
SIFT
SURF
PoIs Extraction (k-means/Random)
BoVW Dictionary
feat$_{1,1}$ ... feat$_{k,1}$
feat$_{2,1}$ ... feat$_{k,1}$
feat$_{1,128}$ ... feat$_{k,128}$
feat$_{1,1}$ ... feat$_{n,1}$
feat$_{2,1}$ ... feat$_{n,1}$
feat$_{1,64}$ ... feat$_{n,64}$
feat$_1$
feat$_2$
feat$_{500}$
feat$_{1,000}$

Figure 7.5: Descriptor calculation for the experiments using SURF, SIFT and BoWV techniques (adapted from (JR. et al., 2017a)).

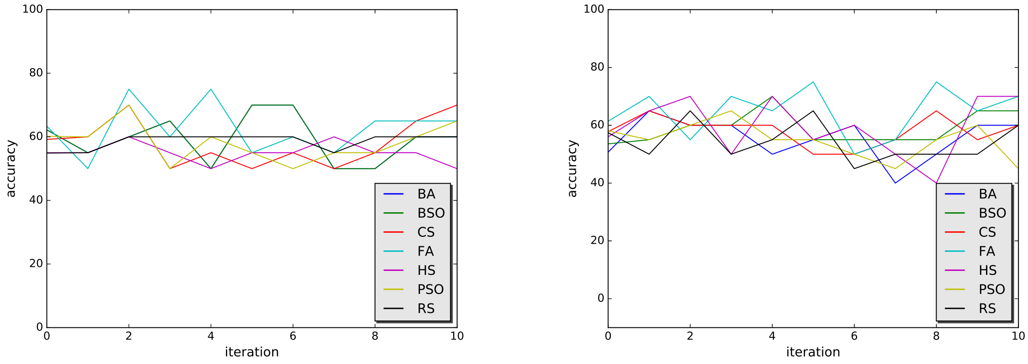
7.5.2 Optimization

Six meta-heuristic optimization techniques, i.e., BA, BSO, CS, FA, HS, and the PSO, were employed in this work to fine-tune the iRBM meta-parameters: learning rate η, weight decay λ, and the beta β. All techniques were initialized with five agents and executed during 10 iterations over 150 epochs. Additionally, we started the model using random variables and executed the iRBM for 15 runs over 150 epochs, hereinafter called Random Search (RS) for comparison purposes.

Figures 7.6 and 7.7 present the results obtained while fine-tuning the model over the validation sets regarding 500 and $1,000$ visual words, respectively. The most accurate iterations were selected for each technique for visualization purposes. Notice that iteration zero stands for the average of the five initial agents for the optimization techniques, as well as the first five runs for RS.

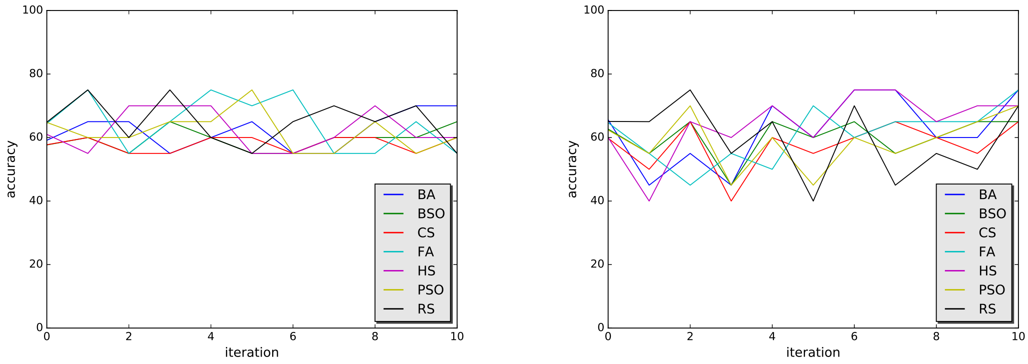
Despite the oscillatory behavior, one can notice that FA obtained the highest results for both SIFT and SURF techniques over 500 visual words, reaching 75% of accuracy during a few iterations (Figure 7.6). It also reached the best results at the end of the optimization steps, which reflects the best values obtained over the testing set.

Regarding $1,000$ visual words, Figure 7.7 depicts a behavior similar to Figure 7.6, with considerable oscillation and FA obtaining the best results, which once again reflects on the results of the test set, presented in Table 7.5. The Harmony Search and the Bat Algorithm behave similarly to FA, also achieving 75% over a few iterations. Additionally, the random search also obtained results statistically similar to the ones found by FA,



(a) (b)

Figure 7.6: Classification accuracies over the validation set during the meta-parameter optimization process concerning 500 visual words for SIFT (a) and SURF (b).



(a) (b)

Figure 7.7: Classification accuracies over the validation set during the meta-parameter optimization process concerning 1,000 visual words for SIFT (a) and SURF (b).

which can be explained by the short number of agents and iterations employed for the optimization convergence, i.e., 5 and 10, respectively. Furthermore, the random search presents an even more oscillatory behavior, as shown in Figure 7.7(b).

Tables 7.2 and 7.3 present the average execution time regarding 20 executions concerning 500 and 1,000 visual words, respectively. Clearly, HS and RS obtained the lowest execution time for both configurations. Since HS (and also RS) updates a single agent for each iteration and the remaining techniques update all the agents for each iteration, it is expected that HS to be faster. In a nutshell, for the configuration employed in this work with 5 agents over 10 iterations, HS and RS evaluate the fitness function 15 times, while the others evaluate 5 times (initialize each agent) and then update each one for 10 times, ending up in 55 executions. Notice CS also presents a small execution time, due to the solutions discarded without evaluation (eggs abandoned by the host bird).

		BA	BSO	CS	FA	HS	PSO	RS
k-means	SIFT	49.55	50.08	22.81	45.12	14.28	49.92	14.29
	SURF	49.96	51.01	23.42	46.38	14.17	50.44	14.50
Random	SIFT	44.93	51.97	24.05	45.29	14.45	51.19	14.99
	SURF	48.74	52.33	25.08	45.31	14.37	50.48	15.27

Table 7.2: Mean computational load (in minutes) of each technique applied to the BE and adenocarcinoma problem using 500 visual words for the feature vector calculation.

		BA	BSO	CS	FA	HS	PSO	RS
k-means	SIFT	50.45	53.27	23.84	47.17	14.10	48.76	14.34
	SURF	49.75	52.15	23.20	45.45	14.29	50.12	14.45
Random	SIFT	49.94	50.98	23.27	46.92	14.25	48.63	14.45
	SURF	49.99	51.64	24.12	45.15	14.29	50.73	14.38

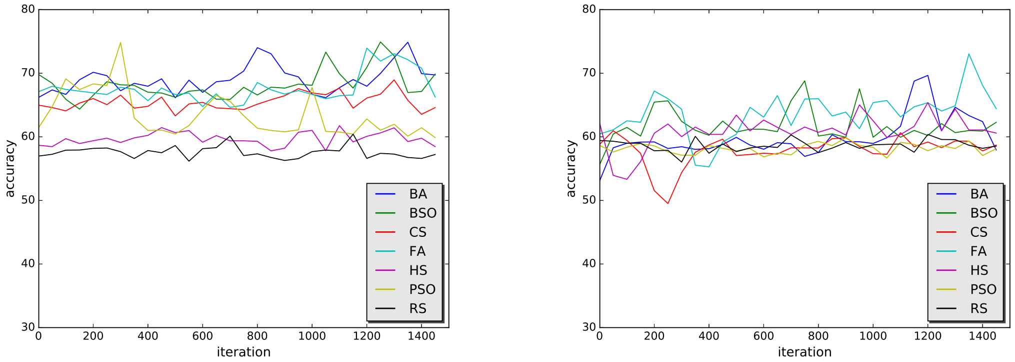
Table 7.3: Mean computational load (in minutes) of each technique applied to the BE and adenocarcinoma problem using 1000 visual words for the feature vector calculation.

7.5.3 Training

The experiments presented in this section employ the best meta-parameters obtained in Section 7.5.2 for each meta-heuristic optimization technique, i.e., the combination of learning rate, weight decay, and β that provided the best accuracies over the validation set during 150 epochs. The iRBM was trained once again, however using $1,500$ epochs.

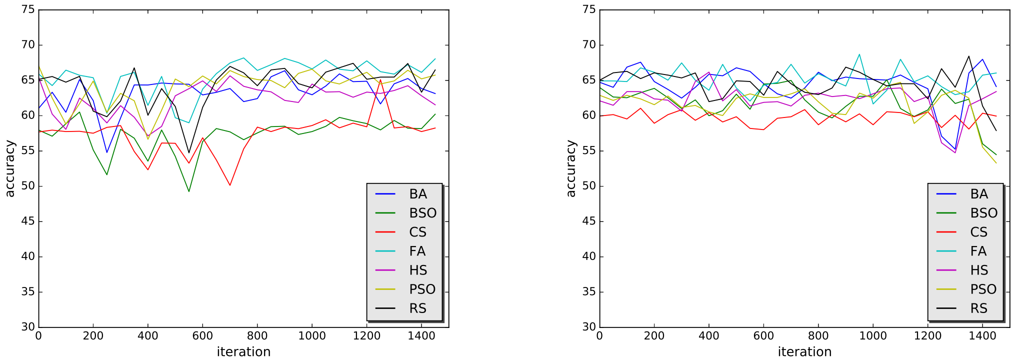
Figure 7.8 depicts the learning steps concerning SIFT and SURF features over the 500-sized dictionary. Despite the oscillatory behavior, which probably can be attributed to the dynamic training strategy (PENG; GAO; LI, 2017) described in Section 7.2.2, one can notice that FA, BSO, and BA interchanged the highest results regarding Figure 7.8(a), while the random search obtained the less accurate results. Concerning SURF, FA also presented the best results, as depicted in Figure 7.8(b). Furthermore, FA appears more inclined to a slight growth behavior than the others techniques, regardless the oscillations.

A similar behavior is observed in Figure 7.9, which concerns SIFT and SURF with a dictionary of size $1,000$. However, FA interchanges the top results with BA, HS, PSO and even the random search. In spite of such exchange, FA seems to stand for the best optimization technique overall.



(a) (b)

Figure 7.8: Classification accuracies during the training convergence process concerning a dictionary composed of 500 visual words for SIFT (a) and SURF (b).



(a) (b)

Figure 7.9: Classification accuracies during the training convergence process concerning a dictionary composed of 1,000 visual words for SIFT (a) and SURF (b).

7.5.4 Classification Step

The experiments conducted in this section are divided according to the feature extraction technique and visual dictionary sizes. Tables 7.4 and 7.5 present the mean accuracy and the standard deviation concerning the classification over the test set during 1,500 training epochs using 500 and 1,000 visual words, respectively. The training employs the combination of meta-parameters that provided the best accuracies over the validation set during the fine-tuning step, presented in Section 7.5.2. Additionally, results are compared against two versions of Support Vector Machines with RBF and linear kernels, as well as the well-known Bayes classifier. Moreover, the experiments were executed for 20 runs for statistical analysis using Wilcoxon signed-rank test (WILCOXON, 1945) with 0.05 of significance, being the most accurate results in bold.

Regarding Table 7.4, one can notice that iRBM fine-tuning with FA outperformed

	SIFT		SURF	
	k-means	Random	*k*-means	Random
iRBM-BA	**63.50±10.6184**	55.65±6.1846	61.60±7.0183	57.40±9.1311
iRBM-BSO	**64.85±9.3770**	51.85±2.9924	**62.70±8.5032**	54.65±3.2247
iRBM-CS	61.40±8.6628	57.45±7.7751	60.97±4.3011	55.90±3.7381
iRBM-FA	**66.15±11.5023**	49.95±2.0504	**66.35±4.7022**	58.80±4.1389
iRBM-HS	59.95±6.2954	55.75±8.1319	**65.35±6.4570**	56.00±5.2749
iRBM-PSO	**65.55±6.5370**	52.50±1.5003	**64.80±6.0156**	51.20±2.7254
iRBM-RS	58.45±5.0028	56.90±3.7025	60.20±6.45883	59.15±7.5520
SVM-RBF	**65.50±9.8467**	**65.60±10.4255**	**64.80±8.2496**	**63.40±11.2731**
SVM-Linear	56.80±4.5527	54.50±6.8608	58.60±6.5392	57.60±4.4458
Bayes	59.98±3.449	53.00±3.5192	56.86±3.1080	53.52±4.6362

Table 7.4: Best accuracy values for the "MICCAI 2015 Endovis Challenge"Dataset using 500 visual words.

all the other techniques, obtaining the most accurate classification results. Additionally, iRBM optimized with all meta-heuristic techniques, except CS and the random search, achieved similar results concerning the Wilcoxon test, as well as SVM-RBF. Such results may lead to two assumptions: (i) meta-heuristic optimization techniques are suitable for fine-tuning iRBM meta-parameters concerning classification tasks, as well as for reconstruction tasks (PASSOS; PAPA, 2017a), since the results obtained using such techniques outperformed the ones obtained with a random initialization of the weights, and (ii) iRBM is appropriate for classification tasks since it outperformed the results obtained by some well-established classifiers, such as SVM and Bayes.

	SIFT		SURF	
	k-means	Random	*k*-means	Random
iRBM-BA	60.81±7.4686	60.01±7.2947	**65.38±7.6846**	60.04±7.4913
iRBM-BSO	58.40±8.8101	60.61±6.7935	60.41±8.6898	58.79±7.5418
iRBM-CS	59.57±10.4727	59.43±9.2710	62.09±7.9337	58.71±8.9159
iRBM-FA	**66.01±9.6896**	62.21±6.3956	**67.00±8.1187**	60.4±9.2183
iRBM-HS	61.32±6.7106	61.08±11.1132	**65.38±6.4580**	59.46±7.1040
iRBM-PSO	62.84±6.8391	60.60±8.0681	59.80±7.8694	58.10±10.0872
iRBM-RS	**65.10±6.0221**	**63.79±6.1503**	**66.30 ±9.3325**	61.71±6.2307
SVM-RBF	**64.10±8.0761**	**63.70±7.0523**	**62.60±7.2304**	**62.10±6.3215**
SVM-Linear	**67.30±9.8649**	52.70±3.5027	**62.80±4.9553**	56.50±4.0050
Bayes	60.70±3.7278	54.37±3.0303	57.43±4.2186	56.86±6.2625

Table 7.5: Best accuracy values for the "MICCAI 2015 Endovis Challenge"Dataset using 1,000 visual words.

Concerning the results presented in Table 7.5, the most accurate results were obtained by SVM-Linear and once again the iRBM fine-tuned using the FA technique, which suggests the idea that FA is the most effective meta-heuristic optimization technique with respect to classification tasks using iRBM. Furthermore, similar results were obtained by SVM-RBF, as well as the iRBM using the HS, BA, and the RS as parameters fine-tuning. The explanation for finding competitive accuracies concerning a random initialization of the meta-parameters may be due to the short number of agents and iterations employed for meta-heuristic optimization techniques. Moreover, one can notice that the configuration using k-means obtained the best results regarding both SIFT and SURF, as well as in with both 500 and $1,000$ words, which suggests that dictionaries composed by employing k-means generate better features.

Considering the very best results obtained for all the techniques, i.e., iRBM, SVM-RBF, SVM-Linear, and Bayes, Tables 7.6 and 7.7 present the sensitivity (SE) and the specificity (SP) results concerning the configuration over 500 and $1,000$ words, respectively. Notice the best values are in bold.

	Accuracy	Sensitivity	Specificity
iRBM-FA	**66.35%**	**0.644**	0.687
SVM-RBF	65.60%	0.612	**0.706**
SVM-Linear	58.60%	0.582	0.593
Bayes	59.98%	0.593	0.605

Table 7.6: Mean SE and SP values for the selected best results obtained using dictionaries of 500 **words.**

	Accuracy	Sensitivity	Specificity
iRBM-FA	67.00%	**0.655**	0.692
SVM-RBF	64.10%	0.632	0.686
SVM-Linear	**67.30%**	0.583	**0.767**
Bayes	60.70%	0.581	0.638

Table 7.7: Mean SE and SP values for the selected best results obtained using dictionaries of $1,000$ **words.**

One can observe that iRBM obtained the best Sensitivity for both configurations, which indicates a higher rate of true positives correctly identified. Such outcome is particularly interesting for medical issues, since a correct identification of some illness, specially in early stages, may prevent the progress of the disease. Additionally, it can be also observed that using either configurations, i.e., dictionaries of 500 and $1,000$ words, does not impact in the final classification accuracy since both scenarios achieved similar results.

7.6 Conclusions and Future works

This work dealt with the problem of automatic Barrett's esophagus identification using infinity Restricted Boltzmann Machines for classification purposes. The approach employs SURF and SIFT techniques to extract the points-of-interest, which are used to build structural patterns in endoscopy images in association with a BoVW. Such descriptors were calculated over a set of images previously annotated by five experts, making the identification of malignant lesions available for classification. Additionally, experiments were conducted over two different dictionary configurations, i.e., 500 and 1,000 words.

From the experiments, we can conclude that: (i) infinity Restricted Boltzmann Machines are convenient for Barrett's esophagus identification task, since it outperformed SVM-Linear and SVM-RBF, as well as the Bayes classifier in one of the configurations, and achieved similar results concerning the other; (ii) meta-heuristic optimization techniques are suitable for iRBM meta-parameter optimization, since they outperformed a random search over an equal number of executions; (iii) the identification of Barrett's esophagus is not a trivial task, once all techniques obtained results under 70% of accuracy.

Based on the last assumption, we can also conclude that Barrett's esophagus identification requires more study and RBM-based approaches may offer an interesting direction in such context, once they are the base blocks for deeper architectures, i.e., Deep Belief Networks and Deep Boltzmann Machines, which are able of extracting deeper characteristics and correlations from data.

Considering future works, we intend to investigate the identification of Barrett's esophagus using RBM-based deeper architectures, as well as other deep learning techniques. Furthermore, we aim to consider the suitability of deeper versions of the iRBM.

Chapter 8

Conclusions

The present thesis was organized into eight chapters, described as follows: the introduction exposed the context of the research, as well as the motivation and main contribution to the proposed subject, while Chapter 2 briefly presented the theoretical background regarding the objective of the research. Chapters 3 and 4 presented a work published in the journal Neural Processing Letters (NPL) (PASSOS; PAPA, 2017c) entitled Temperature-Based Deep Boltzmann Machines, as well as the paper Deep Boltzmann Machines Using Adaptive Temperatures, presented at the 17th International Conference on Computer Analysis of Images and Patterns (CAIP) (PASSOS; COSTA; PAPA, 2017), respectively. The former introduced the temperature parameter into the DBM formulation, while the latter proposed to use the previously mentioned parameter in an adaptive fashion.

Chapter 5 presented the work submitted to the journal Applied Soft Computing (ASoC), which introduced the concepts of meta-heuristic parameters optimization into the DBM domain. Similarly, Chapter 6 employed the idea to the Infinity Restricted Boltzmann Machine(iRBM) context on a paper presented at the 30th Conference on Graphics, Patterns and Images (SIBGRAPI) (PASSOS; PAPA, 2017a). Moreover, Chapter 7 applied iRBM for Barret's Esophagus lesions detection. The latter was submitted to the Journal of Visual Communication and Image Representation (JVCIR) as an invited extension from (PASSOS; PAPA, 2017a).

The results obtained in the aforementioned chapters confirm the hypothesis of this works, evincing that both the application of meta-heuristic optimization algorithms to fine-tune the hyper-parameters, as well as the introduction of the temperature parameter into the RBM-based formulation, are suitable strategies concerning the enhancement of RBM-based models training process.

8.1 Works developed during the study period

Table 8.1 presents the works produced during the study period.

Name	Type	Qualis	Year	Status
Learning Parameters in Deep Belief Networks Through Firefly Algorithm (ROSA et al., 2016b)	Conference	B2	2016	Published
Deep Boltzmann Machines Using Adaptive Temperatures (PASSOS; COSTA; PAPA, 2017)	Conference	B1	2017	Published
Parkinson's Disease Identification Using Restricted Boltzmann Machines (PEREIRA et al., 2017)	Conference	B1	2017	Published
Fine-Tuning Infinity Restricted Boltzmann Machines (PASSOS; PAPA, 2017a)	Conference	B1	2017	Published
A Metaheuristic-Driven Approach to Fine-Tune Deep Boltzmann Machines	Journal	A1	2017	Submitted
Temperature-based Deep Boltzmann Machines (PASSOS; PAPA, 2017c)	Journal	A2	2018	Published
Parkinson Disease Identification Using Residual Networks and Optimum-Path Forest (PASSOS et al., 2018)	Conference	B1	2018	Published
Enhancing Brain Storm Optimization Through Optimum-Path Forest (AFONSO; PASSOS; PAPA, 2018)	Conference	B1	2018	Published
Fine Tuning Deep Boltzmann Machines Through Meta-Heuristic Approaches (PASSOS; RODRIGUES; PAPA, 2018)	Conference	B1	2018	Published
Intelligent Network Security Monitoring based on Optimum-Path Forest Clustering (GUIMARAES et al., 2018)	Journal	A1	2018	Published
Adaptive Improved Flower Pollination Algorithm for Global Optimization	Book Chapter	-	2018	Accepted
Barrett's Esophagus Analysis Using Infinity Restricted Boltzmann Machines	Journal	A2	2018	Submitted
Exudate Detection in Fundus Images Using Deeply-learnable Features (KHOJASTEH et al., 2019)	Journal	A2	2018	Published
Quaternion-Based Backtracking Search Optimization Algorithm	Conference	A1	2019	Submitted
Kaniadakis-Based Restricted Boltzmann Machines	Conference	A1	2019	Submitted

Table 8.1: Works developed during the study period

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

AFONSO, L. C.; PASSOS, L. A.; PAPA, J. a. P. Enhancing brain storm optimization through optimum-path forest. In: IEEE. *2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. [S.l.], 2018. p. 000183–000188.

BASTIEN, F. et al. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*, 2012.

BAY, H. et al. Speeded-up robust features (SURF). *Computer Visision and Image Understanding*, Elsevier Science Inc., New York, NY, USA, v. 110, n. 3, p. 346–359, 2008. ISSN 1077-3142.

BEKENSTEIN, J. D. Black holes and entropy. *Physical Review D*, APS, v. 7, n. 8, p. 2333, 1973.

Beraldo e Silva, L. et al. Statistical mechanics of self-gravitating systems: Mixing as a criterion for indistinguishability. *Physical Review D*, APS, v. 90, n. 12, p. 123004, 2014.

BOUREAU, Y.-l.; CUN, Y. L. et al. Sparse feature learning for deep belief networks. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2008. p. 1185–1192.

BRAKEL, P.; DIELEMAN, S.; SCHRAUWEN, B. Training restricted boltzmann machines with multi-tempering: Harnessing parallelization. In: VILLA, A. E. P. et al. (Ed.). *Artificial Neural Networks and Machine Learning*. [S.l.]: Springer Berlin Heidelberg, 2012, (Lecture Notes in Computer Science, v. 7553). p. 92–99. ISBN 978-3-642-33265-4.

CARREIRA-PERPIÑÁN, M. A.; HINTON, G. E. On Contrastive Divergence Learning. In: COWELL, R. G.; GHAHRAMANI, Z. (Ed.). *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*. [S.l.], 2005. p. 33–40.

CIVICIOGLU, P. Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematics and Computation*, Elsevier, v. 219, n. 15, p. 8121–8144, 2013.

CÔTÉ, M.-A.; LAROCHELLE, H. An infinite restricted boltzmann machine. *Neural computation*, MIT Press, 2016.

CSURKA, G. et al. Visual categorization with bags of keypoints. In: *Proceedings of the Workshop on Statistical Learning in Computer Vision*. [S.l.: s.n.], 2004. p. 1–22.

DENT, J. Barrett's esophagus: a historical perspective, an update on core practicalities and predictions on future evolutions of management. *Journal of Gastroenterology and Hepatology*, v. 26, p. 11–30, 2011.

DONG, C. et al. Learning a deep convolutional network for image super-resolution. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2014. p. 184–199.

DUCHI, J.; HAZAN, E.; SINGER, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, v. 12, n. Jul, p. 2121–2159, 2011.

DUONG, C.-N. et al. 2015 ieee conference on beyond principal components: Deep boltzmann machines for face modeling. In: *Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2015. (CVPR '15), p. 4786–4794.

FIORE, U. et al. Network anomaly detection with the restricted boltzmann machine. *Neurocomputing*, Elsevier, v. 122, p. 13–23, 2013.

FIORE, U. et al. Network anomaly detection with the restricted boltzmann machine. *Neurocomputing*, v. 122, p. 13–23, 2013. Advances in cognitive and ubiquitous computingSelected papers from the Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2012).

GADJIEV, B.; PROGULOVA, T. Origin of generalized entropies and generalized statistical mechanics for superstatistical multifractal systems. In: *International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*. [S.l.: s.n.], 2015. v. 1641, p. 595–602.

GEEM, Z. W. *Music-Inspired Harmony Search Algorithm: Theory and Applications*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2009. ISBN 364200184X, 9783642001840.

GOH, H. et al. Computer vision – eccv 2012: 12th european conference on computer vision, florence, italy, october 7-13, 2012, proceedings, part v. In: ______. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. cap. Unsupervised and Supervised Visual Codes with Restricted Boltzmann Machines, p. 298–311. ISBN 978-3-642-33715-4.

GOMPERTZ, B. On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies. *Philosophical transactions of the Royal Society of London*, JSTOR, v. 115, p. 513–583, 1825.

GORDON, B. L. Maxwell–boltzmann statistics and the metaphysics of modality. *Synthese*, Springer, v. 133, n. 3, p. 393–417, 2002.

GUIMARAES, R. R. et al. Intelligent network security monitoring based on optimum-path forest clustering. *IEEE Network*, IEEE, 2018.

HASSAN, A. R.; HAQUE, M. A. Computer-aided gastrointestinal hemorrhage detection in wireless capsule endoscopy videos. *Computers in Biology and Medicine*, v. 122, p. 341–353, 2015.

HINTON, G. Training products of experts by minimizing contrastive divergence. *Neural Computation*, MIT Press, Cambridge, MA, USA, v. 14, n. 8, p. 1771–1800, 2002. ISSN 0899-7667.

HINTON, G.; SALAKHUTDINOV, R. Reducing the dimensionality of data with neural networks. *Science*, v. 313, n. 5786, p. 504 – 507, 2006.

HINTON, G.; SALAKHUTDINOV, R. Discovering binary codes for documents by learning deep generative models. *Topics in Cognitive Science*, Wiley Online Library, v. 3, n. 1, p. 74–91, 2011.

HINTON, G. E. Neural networks: Tricks of the trade: Second edition. In: MONTAVON, G.; ORR, G. B.; MÜLLER, K.-R. (Ed.). Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. cap. A Practical Guide to Training Restricted Boltzmann Machines, p. 599–619.

HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A fast learning algorithm for deep belief nets. *Neural Computation*, MIT Press, Cambridge, MA, USA, v. 18, n. 7, p. 1527–1554, 2006. ISSN 0899-7667.

HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A fast learning algorithm for deep belief nets. *Neural computation*, MIT Press, v. 18, n. 7, p. 1527–1554, 2006.

HINTON, G. E.; SALAKHUTDINOV, R. R. Replicated softmax: an undirected topic model. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2009. p. 1607–1614.

ITO, M.; KOMATSU, H. Representation of angles embedded within contour stimuli in area v2 of macaque monkeys. *The Journal of neuroscience*, Soc Neuroscience, v. 24, n. 13, p. 3313–3324, 2004.

JOHNSON, M. H. et al. Cryoablation of barrett's esophagus: a pilot study. *Gastrointestinal Endoscopy*, v. 62, p. 842–848, 2005.

JR., L. A. de S. et al. A survey on barrett's esophagus analysis using machine learning. *Computers in Biology and Medicine*, v. 96, p. 203–213, 2018.

JR., L. A. S. et al. Barrett's esophagus identification using optimum-path forest. In: *30th SIBGRAPI Conference on Graphics, Patterns and Images*. [S.l.: s.n.], 2017. p. 308–314.

JR., L. A. S. et al. Barrett's esophagus analysis using SURF features. In: MAIER-HEIN GEB. FRITZSCHE, K. H. et al. (Ed.). *Bildverarbeitung für die Medizin 2017*. Berlin, Heidelberg: Springer, 2017. p. 141–146.

JUNIOR, L. A. P.; PAPA, J. P. A meta-heuristic-driven approach to fine-tune deep boltzmann machines. *arXiv*, 2016.

KENNEDY, J. Particle swarm optimization. In: *Encyclopedia of machine learning*. [S.l.]: Springer, 2011. p. 760–766.

KHOJASTEH, P. et al. Exudate detection in fundus images using deeply-learnable features. *Computers in biology and medicine*, Elsevier, v. 104, p. 62–69, 2019.

LAGERGREN, J.; LAGERGREN, P. Oesophageal cancer. *BMJ*, BMJ Publishing Group Ltd, v. 341, 2010. ISSN 0959-8138.

LAROCHELLE, H.; BENGIO, Y. Classification using discriminative restricted boltzmann machines. In: ACM. *Proceedings of the 25th international conference on Machine learning*. [S.l.], 2008. p. 536–543.

LAROCHELLE, H. et al. An empirical evaluation of deep architectures on problems with many factors of variation. In: ACM. *Proceedings of the 24th international conference on Machine learning*. [S.l.], 2007. p. 473–480.

LAROCHELLE, H. et al. Learning algorithms for the classification restricted boltzmann machine. *Journal of Machine Learning Research*, v. 13, n. 1, p. 643–669, 2012.

LECUN, Y.; BENGIO, Y.; HINTON, G. E. Deep learning. *Nature*, v. 521, p. 436–444, 2015.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998.

LEE, H.; EKANADHAM, C.; NG, A. Y. Sparse deep belief net model for visual area v2. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2008. p. 873–880.

LEPAGE, C.; RACHET, B.; JOOSTE, V. Continuing rapid increase in esophageal adenocarcinoma in england and wales. *The American Journal of Gastroenterology*, v. 103, p. 2694–2699, 2008.

LI, G. et al. Temperature based restricted boltzmann machines. *Scientific reports*, Nature Publishing Group, v. 6, 2016.

LI, G. et al. Temperature based restricted Boltzmann machines. *Scientific Reports*, v. 6, p. 1–12, 2016.

LOPES, N.; RIBEIRO, B. Towards adaptive learning with improved convergence of deep belief networks on graphics processing units. *Pattern Recognition*, Elsevier, v. 47, n. 1, p. 114–127, 2014.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 60, n. 2, p. 91–110, 2004. ISSN 0920-5691.

LY, D. L.; PAPROTSKI, V.; YEN, D. Neural networks on gpus: Restricted boltzmann machines. *see http://www. eecg. toronto. edu/moshovos/CUDA08/doku. php*, 2008.

MAHDAVI, M.; FESANGHARY, M.; DAMANGIR, E. An improved harmony search algorithm for solving optimization problems. *Applied mathematics and computation*, Elsevier, v. 188, n. 2, p. 1567–1579, 2007.

MENDEL, R. et al. Barrett's esophagus analysis using convolutional neural networks. In: . Berlin, Heidelberg: Springer, 2017. p. 80–85.

MENDES, G. et al. Nonlinear kramers equation associated with nonextensive statistical mechanics. *Physical Review E*, APS, v. 91, n. 5, p. 052106, 2015.

MONTUFAR, G.; AY, N. Refinements of universal approximation results for deep belief networks and restricted boltzmann machines. *Neural Computation*, MIT Press, v. 23, n. 5, p. 1306–1319, 2011.

NAIR, V.; HINTON, G. E. Implicit mixtures of restricted boltzmann machines. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2009. p. 1145–1152.

NIVEN, R. K. Exact maxwell–boltzmann, bose–einstein and fermi–dirac statistics. *Physics Letters A*, Elsevier, v. 342, n. 4, p. 286–293, 2005.

OVERHOLT, B. F.; PANJEHPOUR, M.; HALBERG, D. L. Photodynamic therapy for barrett's esophagus with dysplasia and/or early stage carcinoma: long-term results. *Gastrointestinal Endoscopy*, v. 58, p. 183–188, 2003.

PAPA, J. et al. On the harmony search using quaternions. In: SPRINGER. *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*. [S.l.], 2016. p. 126–137.

PAPA, J. P. et al. Efficient supervised optimum-path forest classification for large datasets. *Pattern Recognition*, Elsevier Science Inc., New York, NY, USA, v. 45, n. 1, p. 512–520, 2012.

PAPA, J. P.; FALCÃO, A. X.; SUZUKI, C. T. N. Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, John Wiley & Sons, Inc., New York, NY, USA, v. 19, n. 2, p. 120–131, 2009. ISSN 0899-9457.

PAPA, J. P. et al. On the model selection of bernoulli restricted boltzmann machines through harmony search. In: ACM. *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*. [S.l.], 2015. p. 1449–1450.

PAPA, J. P. et al. On the model selection of bernoulli restricted boltzmann machines through harmony search. In: *Proceedings of the Genetic and Evolutionary Computation Conference*. New York, USA: ACM, 2015. (GECCO '15), p. 1449–1450.

PAPA, J. P. et al. Model selection for discriminative restricted boltzmann machines through meta-heuristic techniques. *Journal of Computational Science*, Elsevier, v. 9, p. 14–18, 2015.

PAPA, J. P. et al. Model selection for discriminative restricted boltzmann machines through meta-heuristic techniques. *Journal of Computational Science*, v. 9, p. 14–18, 2015. ISSN 1877-7503.

PAPA, J. P. et al. Quaternion-based deep belief networks fine-tuning. *Applied Soft Computing*, Elsevier, v. 60, p. 328–335, 2017.

PAPA, J. P. et al. Libopt: An open-source platform for fast prototyping soft optimization techniques. *arXiv preprint arXiv:1704.05174*, 2017.

PAPA, J. P.; ROSA, G. H.; YANG, X.-S. Quaternion-driven deep belief networks fine-tuning. *Applied Soft Computing*, 2016. (submitted).

PAPA, J. P.; SCHEIRER, W.; COX, D. D. Fine-tuning deep belief networks using harmony search. *Applied Soft Computing*, v. 46, p. 875–885, 2016.

PAPA, J. P.; SCHEIRER, W.; COX, D. D. Fine-tuning deep belief networks using harmony search. *Applied Soft Computing*, v. 46, p. 875–885, 2016.

PASSOS, L. A.; COSTA, K. A.; PAPA, J. P. Deep boltzmann machines using adaptive temperatures. In: SPRINGER. *International Conference on Computer Analysis of Images and Patterns*. [S.l.], 2017. p. 172–183.

PASSOS, L. A.; PAPA, J. P. Fine-tuning infinity restricted boltzmann machines. In: LAGE, M. et al. (Ed.). *Electronic Proceedings of the 30th Conference on Graphics, Patterns and Images (SIBGRAPI'17)*. Niterói, RJ, Brazil: [s.n.], 2017. Disponível em: <http://sibgrapi2017.ic.uff.br/>.

PASSOS, L. A.; PAPA, J. P. Fine-tuning infinity restricted boltzmann machines. In: IEEE. *Graphics, Patterns and Images (SIBGRAPI), 2017 30th SIBGRAPI Conference on*. [S.l.], 2017. p. 63–70.

PASSOS, L. A.; PAPA, J. P. Temperature-based deep boltzmann machines. *Neural Processing Letters*, Springer, p. 1–13, 2017.

PASSOS, L. A. et al. Parkinson disease identification using residual networks and optimum-path forest. In: IEEE. *2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. [S.l.], 2018. p. 000325–000330.

PASSOS, L. A.; RODRIGUES, D. R.; PAPA, J. P. Fine tuning deep boltzmann machines through meta-heuristic approaches. In: IEEE. *2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. [S.l.], 2018. p. 000419–000424.

PENG, X.; GAO, X.; LI, X. On better training the infinite restricted boltzmann machines. *arXiv preprint arXiv:1709.03239*, 2017.

PENG, X. et al. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *Computer Vision and Image Understanding*, v. 150, p. 109–125, 2016.

PEREIRA, C. R. et al. Parkinson's disease identification using restricted boltzmann machines. In: SPRINGER. *International Conference on Computer Analysis of Images and Patterns*. [S.l.], 2017. p. 70–80.

PHOA, K. N. et al. Multimodality endoscopic eradication for neoplastic barrett oesophagus: results of an european multicentre study (euro-ii). *Gut*, BMJ Publishing Group, v. 65, n. 4, p. 555–562, 2016.

RAINA, R.; MADHAVAN, A.; NG, A. Y. Large-scale deep unsupervised learning using graphics processors. In: ACM. *Proceedings of the 26th annual international conference on machine learning*. [S.l.], 2009. p. 873–880.

RANZATO, M.; BOUREAU, Y.; CUN, Y. Sparse feature learning for deep belief networks. In: PLATT, J. et al. (Ed.). *Advances in Neural Information Processing Systems 20*. [S.l.]: Curran Associates, Inc., 2008. p. 1185–1192.

RODRIGUES, D.; YANG, X. S.; PAPA, J. P. Fine-tuning deep belief networks using cuckoo search. In: YANG, X. S.; PAPA, J. P. (Ed.). *Bio-Inspired Computation and Applications in Image Processing*. [S.l.]: Academic Press, 2016. p. 47–59.

RODRIGUES, D. et al. Recent advances in swarm intelligence and evolutionary computation. In: _____. Cham: Springer International Publishing, 2015. cap. Binary Flower Pollination Algorithm and Its Application to Feature Selection, p. 85–100. ISBN 978-3-319-13826-8.

ROSA, G. et al. Learning parameters in deep belief networks through firefly algorithm. In: SPRINGER. *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*. [S.l.], 2016. p. 138–149.

ROSA, G. H. et al. Learning parameters in deep belief networks through firefly algorithm. In: _____. *Artificial Neural Networks in Pattern Recognition: 7th IAPR TC3 Workshop, ANNPR*. Cham: Springer International Publishing, 2016. p. 138–149.

ROSA, G. H. et al. Fine-tuning convolutional neural networks using harmony search. In: PARDO, A.; KITTLER, J. (Ed.). *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. [S.l.]: Springer International Publishing, 2015, (Lecture Notes in Computer Science, v. 9423). p. 683–690. 20th Iberoamerican Congress on Pattern Recognition.

RRNYI, A. On measures of entropy and information. 1961.

RUANGKANOKMAS, P.; ACHALAKUL, T.; AKKARAJITSAKUL, K. Deep belief networks with feature selection for sentiment classification. In: *7th International Conference on Intelligent Systems, Modelling and Simulation*. [S.l.: s.n.], 2016.

SALAKHUTDINOV, R.; HINTON, G. Deep boltzmann machines. In: *Artificial Intelligence and Statistics*. [S.l.: s.n.], 2009. p. 448–455.

SALAKHUTDINOV, R.; HINTON, G. Semantic hashing. *International Journal of Approximate Reasoning*, Elsevier, v. 50, n. 7, p. 969–978, 2009.

SALAKHUTDINOV, R.; HINTON, G. E. An efficient learning procedure for deep boltzmann machines. *Neural Computation*, v. 24, n. 8, p. 1967–2006, 2012.

SALAKHUTDINOV, R.; HINTON, G. E. An efficient learning procedure for deep boltzmann machines. *Neural Computation*, MIT Press, Cambridge, MA, USA, v. 24, n. 8, p. 1967–2006, 2012. ISSN 0899-7667.

SALAKHUTDINOV, R.; MNIH, A.; HINTON, G. Restricted boltzmann machines for collaborative filtering. In: ACM. *Proceedings of the 24th international conference on Machine learning*. [S.l.], 2007. p. 791–798.

SALAKHUTDINOV, R.; MURRAY, I. On the quantitative analysis of deep belief networks. In: ACM. *Proceedings of the 25th international conference on Machine learning*. [S.l.], 2008. p. 872–879.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural Networks*, Elsevier, v. 61, p. 85–117, 2015.

SHAHEEN, N. J. et al. Radiofrequency ablation in barrett's esophagus with dysplasia. *The New England Journal of Medicine*, v. 360, p. 2277–2288, 2009.

SHARMA, P. et al. Development and validation of a classification system to identify high-grade dysplasia and esophageal adenocarcinoma in barrett's esophagus using narrow-band imaging. *Gastroenterology*, v. 150, n. 3, p. 591 – 598, 2016. ISSN 0016-5085.

SHI, Y. Brain storm optimization algorithm. In: *Proceedings of the Second International Conference on Advances in Swarm Intelligence - Volume Part I*. Berlin, Heidelberg: Springer, 2011. (ICSI'11), p. 303–309.

SHIM, J. W.; GATIGNOL, R. Robust thermal boundary conditions applicable to a wall along which temperature varies in lattice-gas cellular automata. *Physical Review E*, APS, v. 81, n. 4, p. 046703, 2010.

SILVA, L. A. et al. Learning spam features using restricted boltzmann machines. *IADIS International Journal on Computer Science and Information Systems*, v. 11, n. 1, p. 99–114, 2016.

SMOLENSKY, P. *Information processing in dynamical systems: Foundations of harmony theory*. [S.l.], 1986.

SMOLENSKY, P. Parallel distributed processing: Explorations in the microstructure of cognition. In: RUMELHART, D. E.; MCCLELLAND, J. L.; GROUP, C. P. R. (Ed.). Cambridge, MA, USA: MIT Press, 1986. v. 1, cap. Information Processing in Dynamical Systems: Foundations of Harmony Theory, p. 194–281. ISBN 0-262-68053-X.

SOHN, K.; LEE, H.; YAN, X. Learning structured output representation using deep conditional generative models. In: CORTES, C. et al. (Ed.). *Advances in Neural Information Processing Systems 28*. [S.l.]: Curran Associates, Inc., 2015. p. 3465–3473.

SOMMEN, F. van der et al. Computer-aided detection of early neoplastic lesions in barrett's esophagus. *Endoscopy*, © Georg Thieme Verlag KG, v. 48, n. 07, p. 617–624, 2016.

SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, v. 15, n. 1, p. 1929–1958, 2014.

SRIVASTAVA, N. et al. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, JMLR.org, v. 15, n. 1, p. 1929–1958, jan. 2014. ISSN 1532-4435.

SRIVASTAVA, N.; SALAKHUTDINOV, R. R. Multimodal learning with deep boltzmann machines. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2012. p. 2222–2230.

SWERSKY, K. et al. Cardinality restricted boltzmann machines. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2012. p. 3293–3301.

TAIGMAN, Y. et al. Deepface: Closing the gap to human-level performance in face verification. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 1701–1708.

TAYLOR, G. W.; HINTON, G. E.; ROWEIS, S. T. Modeling human motion using binary latent variables. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2006. p. 1345–1352.

TIELEMAN, T. Training restricted Boltzmann machines using approximations to the likelihood gradient. In: *Proceedings of the 25th International Conference on Machine Learning*. New York, NY, USA: ACM, 2008. (ICML '08), p. 1064–1071. ISBN 978-1-60558-205-4.

TIELEMAN, T. Training restricted boltzmann machines using approximations to the likelihood gradient. In: *Proceedings of the 25th International Conference on Machine Learning*. New York, USA: ACM, 2008. p. 1064–1071.

TIELEMAN, T.; HINTON, G. E. Using fast weights to improve persistent contrastive divergence. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. New York, NY, USA: ACM, 2009. (ICML '09), p. 1033–1040. ISBN 978-1-60558-516-1.

TOMCZAK, J. M.; GONCZAREK, A. Learning invariant features using subspace restricted boltzmann machine. *Neural Processing Letters*, p. 1–10, 2016.

WAN, L. et al. Regularization of neural networks using dropconnect. In: DASGUPTA, S.; MCALLESTER, D. (Ed.). *Proceedings of the 30th International Conference on Machine learning*. [S.l.]: JMLR Workshop and Conference Proceedings, 2013. (ICML '13, 3), p. 1058–1066.

WANG, Y. et al. Differential evolution based on covariance matrix learning and bimodal distribution parameter setting. *Applied Soft Computing*, Elsevier, v. 18, p. 232–247, 2014.

WELLING, M.; ROSEN-ZVI, M.; HINTON, G. E. Exponential family harmoniums with an application to information retrieval. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2004. p. 1481–1488.

WICHT, B.; FISCHER, A.; HENNEBERT, J. On cpu performance optimization of restricted boltzmann machine and convolutional rbm. In: SPRINGER. *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*. [S.l.], 2016. p. 163–174.

WILCOXON, F. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, International Biometric Society, v. 1, n. 6, p. 80–83, dez. 1945. ISSN 00994987. Disponível em: <http://dx.doi.org/10.2307/3001968>.

XU, J.; LI, H.; ZHOU, S. Improving mixing rate with tempered transition for learning restricted boltzmann machines. *Neurocomputing*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 139, p. 328–335, 2014. ISSN 0925-2312.

YANG, X.-S. Firefly algorithm, stochastic test functions and design optimisation. *International Journal Bio-Inspired Computing*, Inderscience Publishers, v. 2, n. 2, p. 78–84, 2010.

YANG, X.-S.; DEB, S. Cuckoo search via lévy flights. In: IEEE. *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*. [S.l.], 2009. p. 210–214.

YANG, X.-S.; DEB, S. Engineering optimisation by cuckoo search. *International Journal of Mathematical Modelling and Numerical Optimisation*, Inderscience Publishers, v. 1, n. 4, p. 330–343, 2010.

YANG, X.-S.; GANDOMI, A. H. Bat algorithm: a novel approach for global engineering optimization. *Engineering Computations*, v. 29, n. 5, p. 464–483, 2012.

YU, X.; LIU, J.; LI, H. An adaptive inertia weight particle swarm optimization algorithm for iir digital filter. In: IEEE. *Artificial Intelligence and Computational Intelligence, 2009. AICI'09. International Conference on.* [S.l.], 2009. v. 1, p. 114–118.

ZHANG, J.; SANDERSON, A. C. Jade: adaptive differential evolution with optional external archive. *IEEE Transactions on evolutionary computation*, IEEE, v. 13, n. 5, p. 945–958, 2009.

Glossary

AIWPSO – *Adaptive Inertia Weight Particle Swarm Optimization*

ASoC – *Applied Soft Computing*

BA – *Bat Algorithm*

BE – *Barrett's esophagus*

BSA – *Backtracking Search Optimization Algorithm*

BSO – *Brain Storm Optimization*

BoVW – *Bag-of-Visual-Words*

CAIP – *International Conference on Computer Analysis of Images and Patterns*

CNN – *Convolutional Neural Networks*

CS – *Cuckoo Search*

CoBiDE – *Differential Evolution Based on Covariance Matrix Learning and Bimodal Distribution Parameter Setting Algorithm*

DBM – *Deep Boltzmann Machine*

DBN – *Deep Belief Network*

DL – *Deep Learning*

FA – *Firefly Algorithm*

HS – *Harmony Search*

IHS – *Improved Harmony Search*

JADE – *Adaptive Differential Evolution*

JVCIR – *Journal of Visual Communication and Image Representation*

MSE – *Mean Squared Error*

NPL – *Neural Processing Letters*

OPF – *Optimum-Path Forest*

PCD – *Persistent Contrastive Divergence*

PSO – *Particle Swarm Optimization*

PoIs – *Points-of-interest*

RBF – *Radial Basis Function*

RBM – *Restricted Boltzmann Machines*

SESM – *Sparse Encoding Symmetric Machine*

SIBGRAPI – *Conference on Graphics, Patterns and Images*

SIFT – *Scale Invariant Feature Transform*

SURF – *Speed-Up Robust Features*

SVM – *Support Vector Machines*

TRBM – *Temperature-based Restrict Boltzmann machine*