Neural Process Lett (2018) 48:95–107
https://doi.org/10.1007/s11063-017-9707-2

# Temperature-Based Deep Boltzmann Machines

**Leandro Aparecido Passos Jr.**[1] · **João Paulo Papa**[2]

Published online: 8 September 2017
© Springer Science+Business Media, LLC 2017

**Abstract** Deep learning techniques have been paramount in the last years, mainly due to their outstanding results in a number of applications, that range from speech recognition to face-based user identification. Despite other techniques employed for such purposes, Deep Boltzmann Machines (DBMs) are among the most used ones, which are composed of layers of Restricted Boltzmann Machines stacked on top of each other. In this work, we evaluate the concept of temperature in DBMs, which play a key role in Boltzmann-related distributions, but it has never been considered in this context up to date. Therefore, the main contribution of this paper is to take into account this information, as well as the impact of replacing a standard Sigmoid function by another one and to evaluate their influence in DBMs considering the task of binary image reconstruction. We expect this work can foster future research considering the usage of different temperatures during learning in DBMs.

**Keywords** Deep Learning · Deep Boltzmann Machines · Machine learning

## 1 Introduction

Deep learning techniques have attracted considerable attention in the last years due to their outstanding results in a number of applications [3,5,30], since such techniques possess an intrinsic ability to learn different information at each level of a hierarchy of layers [15]. Restricted Boltzmann Machines (RBMs) [11], 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

✉ João Paulo Papa
papa@fc.unesp.br

Leandro Aparecido Passos Jr.
leandropassosjr@gmail.com

1 Department of Computing, Federal University of São Carlos, São Carlos, Brazil

2 Department of Computing, São Paulo State University, Bauru, Brazil

Belief Networks (DBNs) [12] or the Deep Boltzmann Machines (DBMs) [26], 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 [14,27]. RBMs were firstly idealized under the name of "Harmonium" by Smolensky in 1986 [29], and some years later renamed to RBM by Hinton et. al. [10]. 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 [8,9,19–21,33].

Roughly speaking, 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. [16] 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 [2,4,17], idealized by Wolfgang Boltzmann. In fact, a Maxwell-Boltzmann distribution [7,18,28] 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. [16] 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 [1,23].

Since DBMs are a natural extension of RBMs and DBNs, we believe the temperature can also play an important role in these models. Roughly speaking, 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 [6], 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 paper is organized as follows. Sect. 2 presents the theoretical background related to DBMs and the proposed temperature-based approach, and Sect. 3 describes the methodology adopted in this work. The experimental results are discussed in Sect. 4, and conclusions and future works are stated in Sect. 5.

## 2 Theoretical Background

In this section, we briefly explain the theoretical background related to RBMs and DBMs.

### 2.1 Restricted Boltzmann Machines

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 $h_j$.

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 energy function of a Restricted Boltzmann Machine is given by:

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

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

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

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

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:

$$P(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}. \tag{3}$$

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:

$$P(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{m} P(v_i|\mathbf{h}), \tag{4}$$

and

$$P(\mathbf{h}|\mathbf{v}) = \prod_{j=1}^{n} P(h_j|\mathbf{v}), \tag{5}$$

where

$$P(v_i = 1|\mathbf{h}) = \phi\left(\sum_{j=1}^{n} w_{ij} h_j + a_i\right), \tag{6}$$

and

$$P(h_j = 1|\mathbf{v}) = \phi\left(\sum_{i=1}^{m} w_{ij} v_i + b_j\right). \tag{7}$$

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

Let $\theta = (W, a, b)$ 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 all the available training data $\mathcal{V}$, as follows:

$$\arg\max_{\Theta} \prod_{\mathbf{v}\in\mathcal{V}} P(\mathbf{v}). \tag{8}$$

One can solve the aforementioned equation using the following derivatives over the matrix of weights $\mathbf{W}$, and biases $\mathbf{a}$ and $\mathbf{b}$ at iteration $t$ as follows:

$$\mathbf{W}^{t+1} = \mathbf{W}^t + \underbrace{\eta(P(\mathbf{h}|\mathbf{v})\mathbf{v}^T - P(\tilde{\mathbf{h}}|\tilde{\mathbf{v}})\tilde{\mathbf{v}}^T) + \Phi}_{=\Delta\mathbf{W}^t}, \tag{9}$$

$$\mathbf{a}^{t+1} = \mathbf{a}^t + \underbrace{\eta(\mathbf{v} - \tilde{\mathbf{v}}) + \alpha\Delta\mathbf{a}^{t-1}}_{=\Delta\mathbf{a}^t} \tag{10}$$

and

$$\mathbf{b}^{t+1} = \mathbf{b}^t + \underbrace{\eta(P(\mathbf{h}|\mathbf{v}) - P(\tilde{\mathbf{h}}|\tilde{\mathbf{v}})) + \alpha\Delta\mathbf{b}^{t-1}}_{=\Delta\mathbf{b}^t}, \tag{11}$$

where $\eta$ stands for the learning rate and $\alpha$ denotes the momentum. Notice the terms $P(\tilde{\mathbf{h}}|\tilde{\mathbf{v}})$ and $\tilde{\mathbf{v}}$ can be obtained by means of the Contrastive Divergence [10] technique[1], which basically ends up performing Gibbs sampling using the training data as the visible units. Roughly speaking, Eqs. 9, 10 and 10 employ the well-known Gradient Descent as the optimization algorithm. The additional term $\Phi$ in Eq. 9 is used to control the values of matrix $\mathbf{W}$ during the convergence process, and it is formulated as follows:

$$\Phi = -\lambda\mathbf{W}^t + \alpha\Delta\mathbf{W}^{t-1}, \tag{12}$$

where $\lambda$ stands for the weight decay.

### 2.2 Deep Boltzmann Machines

Learning more complex and internal representations of the data can be accomplished by using stacked RBMs, such as DBNs and DBMs. In this paper, we are interested in the DBM formulation, which is slightly different from DBN one. Suppose we have a DBM with two layers, where $\mathbf{h}^1$ and $\mathbf{h}^2$ stand for the hidden units at the first and second layer, respectively.

The energy of a DBM can be computed as follows:

$$E(\mathbf{v}, \mathbf{h}^1, \mathbf{h}^2) = -\sum_{i=1}^{m^1}\sum_{j=1}^{n^1} v_i h_j^1 w_{ij}^1 - \sum_{i=1}^{m^2}\sum_{j=1}^{n^2} h_i^1 h_j^2 w_{ij}^2, \tag{13}$$

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:

$$P(\mathbf{v}) = \frac{1}{Z}\sum_{\mathbf{h}^1,\mathbf{h}^2} e^{-E(\mathbf{v},\mathbf{h}^1,\mathbf{h}^2)}. \tag{14}$$

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

$$P(v_i = 1|\mathbf{h}^1) = \phi\left(\sum_{j=1}^{n^1} w_{ij}^1 h_j^1\right), \tag{15}$$

[1] Notice that $\tilde{\mathbf{v}}$ and $\tilde{\mathbf{h}}$ are obtained by sampling from $\mathbf{h}$ and $\tilde{\mathbf{v}}$, respectively.

$$P(h_z^2 = 1|\mathbf{h}^1) = \phi\left(\sum_{i=1}^{m^2} w_{iz}^2 h_i^1\right), \tag{16}$$

and

$$P\left(h_j^1 = 1|\mathbf{v}, \mathbf{h}^2\right) = \phi\left(\sum_{i=1}^{m^1} w_{ij}^1 v_i + \sum_{z=1}^{n^2} w_{jz}^2 h_z^2\right). \tag{17}$$

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

$$P(\mathbf{v}) = \sum_{\mathbf{h}^1} P(\mathbf{h}^1) P(\mathbf{v}|\mathbf{h}^1), \tag{18}$$

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 $P(\mathbf{h}^1) = \sum_{\mathbf{h}^2} P(\mathbf{h}^1, \mathbf{h}^2)$. Roughly speaking, using such procedure, the conditional probabilities given by Eq. 15–16, and Contrastive Divergence, one can learn DBM parameters one layer at a time [26].

### 2.3 Temperature-Based Deep Boltzmann Machines

Li et. al. [16] 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 Eq. 2, which can be rewritten as follows:

$$P(\mathbf{v}, \mathbf{h}; T) = \frac{1}{Z} e^{\frac{-E(\mathbf{v},\mathbf{h})}{T}}. \tag{19}$$

As such, when $T = 1$ the aforementioned equation degenerates to Eq. 2. Therefore, the probability of a given sample $\mathbf{v}$ given by Eq. 14 can be rewritten considering now the temperature:

$$P(\mathbf{v}; T) = \frac{1}{Z} \sum_{\mathbf{h}^1, \mathbf{h}^2} e^{\frac{-E(\mathbf{v},\mathbf{h}^1,\mathbf{h}^2)}{T}}. \tag{20}$$

In addition, Eq. 6 can be rewritten in order to accommodate the temperature parameter as follows:

$$P(h_j = 1|\mathbf{v}) = \phi\left(\frac{\sum_{i=1}^{m} w_{ij} v_i}{T}\right). \tag{21}$$

Notice the temperature parameter does not affect the conditional probability of the input units (Eq. 5).

In order to apply the very same idea to DBMs, the conditional probabilities over the two hidden layers given by Eqs. 16 and 16 can be derived and expressed using the following formulation, respectively:

$$P\left(h_z^2 = 1|\mathbf{h}^1\right) = \phi\left(\frac{\sum_{i=1}^{m^2} w_{iz}^2 h_i^1}{T}\right), \tag{22}$$

and

$$P\left(h_j^1 = 1|\mathbf{v}, \mathbf{h}^2\right) = \phi\left(\frac{\sum_{i=1}^{m^1} w_{ij}^1 v_i + \sum_{z=1}^{n^2} w_{jz}^2 h_z^2}{T}\right). \tag{23}$$

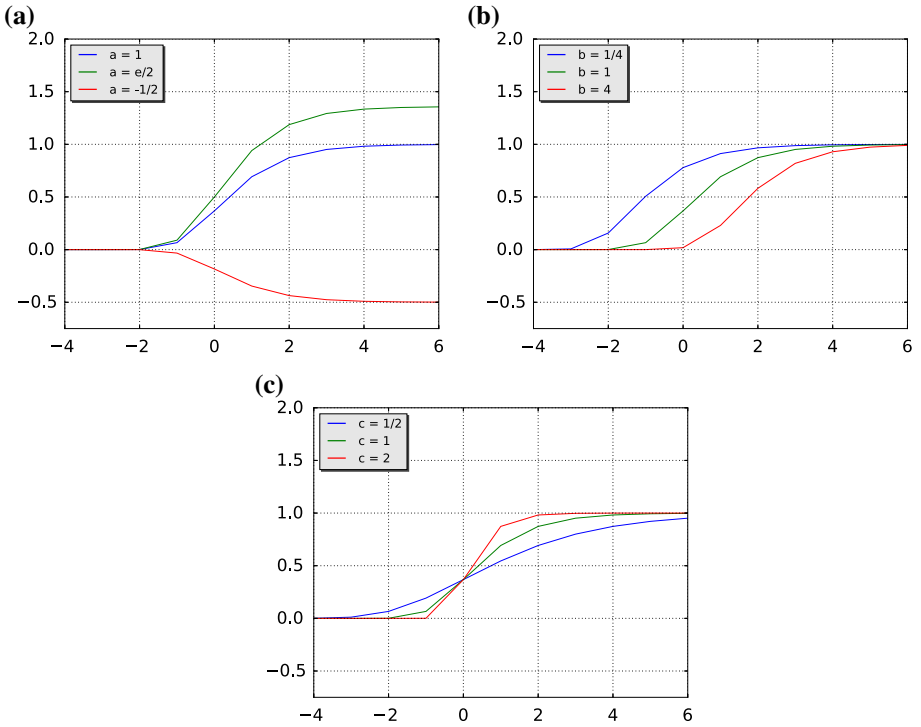


**Fig. 1** Gompertz curves variating theirs parameters for **a**, **b** and **c**, respectively. Note the remaining parameters are fixed to 1

### 2.4 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:

$$f(t) = ae^{-be^{-ct}}, \tag{24}$$

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 1 depicts the behavior of the function concerning its parameters.

## 3 Methodology

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

### 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, as described below:

- 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 silhouettes of images from 101 classes with resolution of $28 \times 28$. We have used only the training and test sets, since our optimization model aims at minimizing the MSE error over the training set.
- Semeion Handwritten Digit Dataset[6]: it is formed by 1593 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 $16 \times 16$ in a grayscale of 256 values and then each pixel was binarized.

### 3.2 Experimental Setup

We employed a 3-layered architecture for all datasets as follows: $i$-500-500-2000, where $i$ stands for the number of pixels used as input for each dataset, i.e. 196 ($14 \times 14$ images), 784 ($28 \times 28$ images) and 256 ($16 \times 16$ images) considering MNIST, Caltech 101 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 2000 neurons.[7] 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.[8]

In order to provide a statistical analysis by means of the Wilcoxon signed-rank test with significance of 0.05 [35], 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) [13]. Since $a$ controls the bounds of the function, and we are dealing with a binary reconstruction problem, we set $a = 1$.

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

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

[3] The images are originally available in grayscale with resolution of $28 \times 28$, but they were reduced to $14 \times 14$ images.

[4] The original training set was reduced to 2% of its former size, which corresponds to 1200 images.

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

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

[7] Similar architectures have been commonly employed in the literature [12,16,24,25,34].

[8] Notice all parameters and architectures have been empirically chosen [21].

**Table 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 |

both DBMs and DBNs with two different algorithms[9,10]: Contrastive Divergence (CD) [10] and Persistent Contrastive Divergence (PCD) [32].

## 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 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 minimum squared error (MSE) over the test set as the measure for comparison purposes.

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. [16], 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. [22], 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 [31].

We have observed the lower the temperature values, the higher the probability of turning "on" hidden units (Eq. 23), 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 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 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 1, some of the better results were obtained using $T = 0.5$ (Fig. 2b), with $T = 1$ (Fig. 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 Fig. 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 (Fig. 3).

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

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

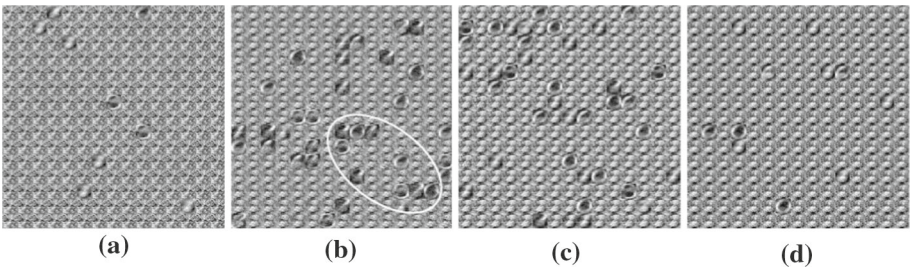


**Fig. 2** Effect of different temperatures by means of DBM-PCD considering Semeion Handwritten Digit dataset with respect to the connection weights of the first hidden layer for: **a** $T = 0.1$, **b** $T = 0.5$, **c** $T = 1.0$, **d** $T = 2.0$

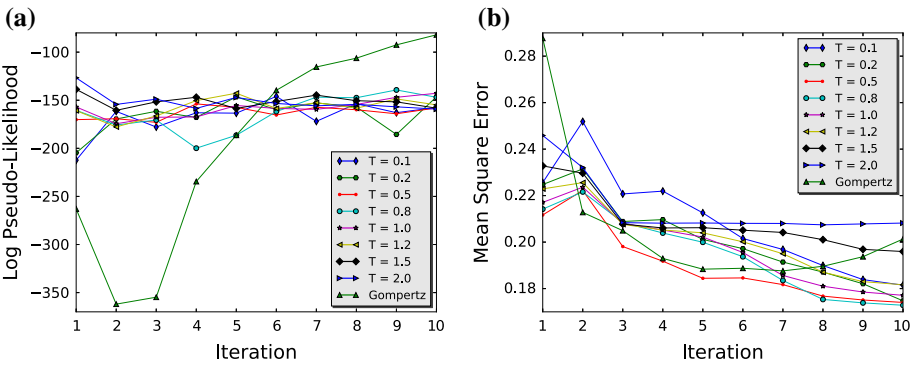


**Fig. 3** Evolution of the: **a** logarithm of the pseudo-likelihood, and **b** mean squared error by means of DBM-PCD considering Semeion Handwritten Digit dataset

**Table 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 |

Table 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—Eqs. 13 and 14). Figure 4 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 (Fig. 4d), since most of them have similar information encoded.

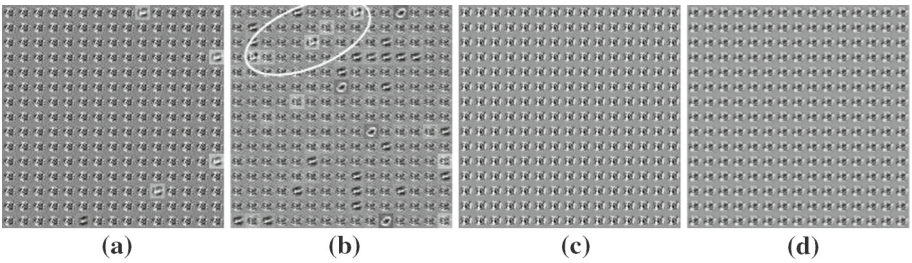


**Fig. 4** Effect of different temperatures by means of DBM-PCD considering MNIST dataset with respect to the connection weights of the first hidden layer for: **a** $T = 0.1$, **b** $T = 0.5$, **c** $T = 1.0$, **d** $T = 2.0$

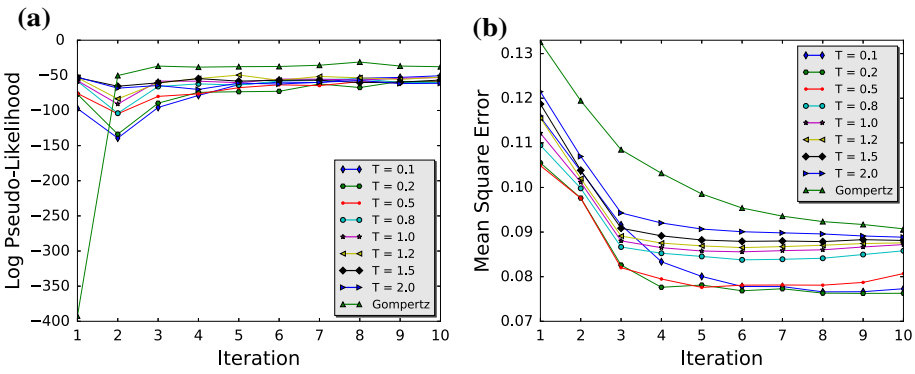


**Fig. 5** Evolution of the: **a** logarithm of the pseudo-likelihood and **b** mean squared error by means of DBM-PCD considering MNIST dataset

Figure 5 depicts the evolution of two distinct measures to monitor the learning process of a DBM over the MNIST dataset at the first layer. The first one (Fig. 7a) is the logarithm of the pseudo-likelihood (PL) of Eq. 20, where the larger its value, the more similar the reconstructed data is concerning its original version. The second measure (Fig. 7b) concerns the mean squared error of the reconstruction data over the training set, where the lower values are the best ones. One can observe the lower temperatures obtained the largest PL values (e.g. $T = 0.1$ and $T = 0.5$) and the lowest errors ($T = 0.1$ and $T = 0.2$).

Table 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 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 Fig. 6. Curiously, the neurons' response at the lower temperatures (Fig. 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 paper, 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 (Fig. 7). 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

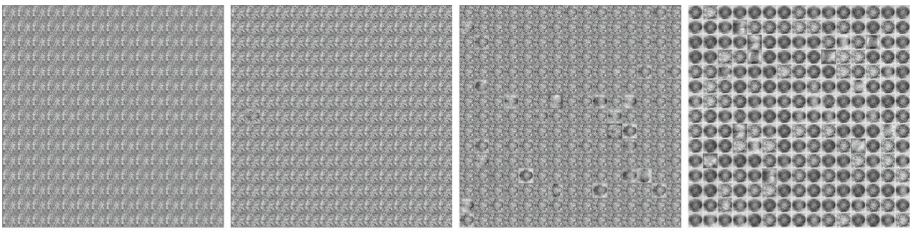


**Fig. 6** Effect of different temperatures by means of DBM-PCD considering Caltech 101 Silhouettes dataset with respect to the connection weights of the first hidden layer for: **a** $T = 0.1$, **b** $T = 0.5$, **c** $T = 1.0$, **d** $T = 2.0$

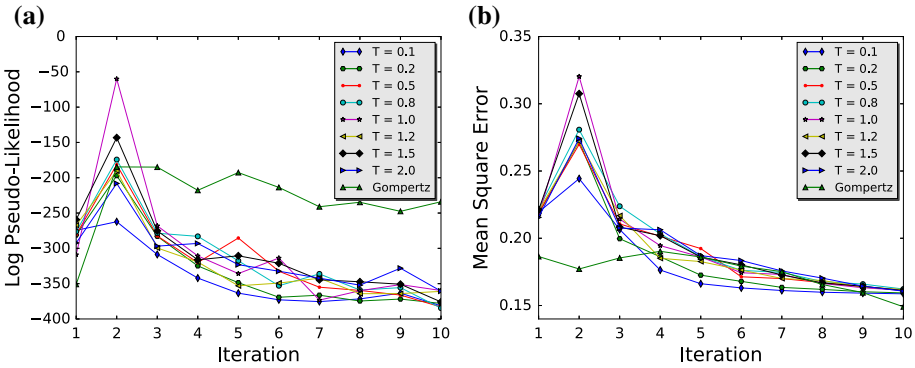


**Fig. 7** Evolution of log pseudo-likelihood and mean squared error for **a** and **b**, respectively, by means of DBM-PCD considering Caltech 101 Silhouettes dataset

**Table 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 |

probability smaller then 50% when the input value is negative. However, such behavior can not be observed in the Gompertz function, where most of it 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.

## 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 Boltzmann

Machines [16], 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. [16], 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.

**Acknowledgements** The authors would like to thank FAPESP Grants #2014/16250-9, #2014/12236-1, and #2016/19403-6, Capes and CNPq Grant #306166/2014-3.

## References

1. Bekenstein JD (1973) Black holes and entropy. Phys Rev D 7(8):2333
2. Beraldo e Silva L, Lima M, Sodré L, Perez J (2014) Statistical mechanics of self-gravitating systems: mixing as a criterion for indistinguishability. Phys Rev D 90(12):123004
3. Duong CN, Luu K, Quach KG, Bui TD (2015) Beyond principal components: deep Boltzmann machines for face modeling. In: 2015 IEEE conference on computer vision and pattern recognition, CVPR'15, pp 4786–4794
4. Gadjiev B, Progulova T (2015) 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, vol 1641, pp 595–602
5. Goh H, Thome N, Cord M, Lim JH (2012) Unsupervised and Supervised Visual Codes with Restricted Boltzmann Machines. In: Proceedings of computer vision—ECCV 2012: 12th European conference on computer vision, Florence, Italy, October 7–13, 2012, Part V. Springer, Berlin, pp 298–311. doi:10.1007/978-3-642-33715-4_22
6. Gompertz B (1825) 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. Philos Trans R Soc Lond 115:513–583
7. Gordon BL (2002) Maxwell–Boltzmann statistics and the metaphysics of modality. Synthese 133(3):393–417
8. Hinton G, Salakhutdinov R (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
9. Hinton G, Salakhutdinov R (2011) Discovering binary codes for documents by learning deep generative models. Top Cogn Sci 3(1):74–91
10. Hinton GE (2002) Training products of experts by minimizing contrastive divergence. Neural Comput 14(8):1771–1800
11. Hinton GE (2012) A practical guide to training restricted Boltzmann machines. In: Montavon G, Orr GB, Müller K-R (eds) Neural networks: tricks of the trade, 2nd edn. Springer, Berlin, pp 599–619. doi:10.1007/978-3-642-35289-8_32
12. Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554
13. Kennedy J (2011) Particle swarm optimization. In: Sammut C, Webb GI (eds) Encyclopedia of machine learning. Springer, pp 760–766
14. Larochelle H, Mandel M, Pascanu R, Bengio Y (2012) Learning algorithms for the classification restricted Boltzmann machine. J Mach Learn Res 13(1):643–669
15. LeCun Y, Bengio Y, Hinton GE (2015) Deep learning. Nature 521:436–444
16. Li G, Deng L, Xu Y, Wen C, Wang W, Pei J, Shi L (2016) Temperature based restricted Boltzmann machines. Sci Rep 6(19):133

17. Mendes G, Ribeiro M, Mendes R, Lenzi E, Nobre F (2015) Nonlinear Kramers equation associated with nonextensive statistical mechanics. Phys Rev E 91(5):052–106
18. Niven RK (2005) Exact Maxwell–Boltzmann, Bose–Einstein and fermi-dirac statistics. Phys Lett A 342(4):286–293
19. Papa JP, Rosa GH, Costa KAP, Marana AN, Scheirer W, Cox DD (2015) On the model selection of Bernoulli restricted Boltzmann machines through harmony search. In: Proceedings of the genetic and evolutionary computation conference, GECCO'15. ACM, New York, pp 1449–1450
20. Papa JP, Rosa GH, Marana AN, Scheirer W, Cox DD (2015) Model selection for discriminative restricted boltzmann machines through meta-heuristic techniques. J Comput Sci 9:14–18
21. Papa JP, Scheirer W, Cox DD (2016) Fine-tuning deep belief networks using harmony search. Appl Soft Comput 46:875–885
22. Ranzato M, Boureau Y, Cun Y (2008) Sparse feature learning for deep belief networks. In: Platt J, Koller D, Singer Y, Roweis S (eds) Advances in neural information processing systems, vol 20. Curran Associates Inc., Red Hook, pp 1185–1192
23. Rényi, A. (1961) On measures of entropy and information. In: Neyman J (ed) Proceedings IV Berkeley symposium on mathematical statistics and probability, vol 1. pp 547–561
24. Salakhutdinov R, Hinton G (2009) Semantic hashing. Int J Approx Reason 50(7):969–978
25. Salakhutdinov R, Hinton GE (2009) Deep Boltzmann machines. In: Neil L, Mark R (eds) AISTATS, vol 1. Microtome Publishing, Brookline, MA, pp 3
26. Salakhutdinov R, Hinton GE (2012) An efficient learning procedure for deep Boltzmann machines. Neural Comput 24(8):1967–2006
27. Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117
28. Shim JW, Gatignol R (2010) Robust thermal boundary conditions applicable to a wall along which temperature varies in lattice-gas cellular automata. Phys Rev E 81(4):046–703
29. Smolensky P (1986) Information processing in dynamical systems: foundations of harmony theory. In: Technical reports, DTIC Document
30. Sohn K, Lee H, Yan X (2015) Learning structured output representation using deep conditional generative models. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R (eds) Advances in neural information processing systems, vol 28. Curran Associates Inc., Red Hook, pp 3465–3473
31. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
32. Tieleman T (2008) Training restricted Boltzmann machines using approximations to the likelihood gradient. In: Proceedings of the 25th international conference on machine learning, ICML '08. ACM, New York, pp 1064–1071 (2008)
33. Tomczak JM, Gonczarek A (2016) Learning invariant features using subspace restricted Boltzmann machine. Neural Process Lett 9896:1–10
34. Wicht B, Fischer A, Hennebert J (2016) On CPU performance optimization of restricted Boltzmann machine and convolutional rbm. In: IAPR workshop on artificial neural networks in pattern recognition. Springer, pp 163–174 (2016)
35. Wilcoxon F (1945) Individual comparisons by ranking methods. Biom Bull 1(6):80–83. doi:10.2307/3001968