Energy-based Dropout in Restricted Boltzmann Machines: Why not go random

Mateus Roder, *Member, IEEE*, Gustavo H. de Rosa, *Member, IEEE*, Victor Hugo C. de Albuquerque, *Senior Member, IEEE*, André L. D. Rossi, *Member, IEEE*, and João P. Papa, *Senior Member, IEEE*

Abstract—Deep learning architectures have been widely fostered throughout the last years, being used in a wide range of applications, such as object recognition, image reconstruction, and signal processing. Nevertheless, such models suffer from a common problem known as overfitting, which limits the network from predicting unseen data effectively. Regularization approaches arise in an attempt to address such a shortcoming. Among them, one can refer to the well-known Dropout, which tackles the problem by randomly shutting down a set of neurons and their connections according to a certain probability. Therefore, this approach does not consider any additional knowledge to decide which units should be disconnected. In this paper, we propose an energy-based Dropout (E-Dropout) that makes conscious decisions whether a neuron should be dropped or not. Specifically, we design this regularization method by correlating neurons and the model's energy as an importance level for further applying it to energy-based models, such as Restricted Boltzmann Machines (RBMs). The experimental results over several benchmark datasets revealed the proposed approach's suitability compared to the traditional Dropout and the standard RBMs.

Index Terms—Machine learning, Restricted Boltzmann Machines, Regularization, Dropout, Energy-based Dropout

I. INTRODUCTION

Machine learning (ML) techniques have been broadly investigated to create authentic representations of the real world. Recently, deep learning has emerged as a significant area in ML [1], since its techniques have achieved outstanding results and established several hallmarks in a wide range of applications, such as image classification, object detection, and speech recognition, to cite a few.

Restricted Boltzmann Machines (RBMs) [2] attracted considerable attention in the past years, mainly due to their simplicity, high-level parallelism, and comprehensive representation capacity. Such models stand for stochastic neural networks based on energy principles and guided by physical laws. Usually, these networks learn in an unsupervised fashion [3] and are applied in various problems, e.g., image reconstruction, collaborative filtering, and feature extraction.

Machine learning algorithms are commonly trained according to an error metric called loss function (training error). Nevertheless, their biggest challenge lies in achieving a low generalization error (testing error). Whenever there is a high

Mateus Roder, Gustavo H. de Rosa, André L. D. Rossi, and João P. Papa are with the São Paulo State University, Brazil and Victor Hugo C. de Albuquerque is with ARMTEC Tecnologia em Robótica, Fortaleza/CE, Brazil. (email: {mateus.roder, gustavo.rosa, andre.rossi, joao.papa}@unesp.br, victor.albuquerque@ieee.org).

discrepancy between training and testing errors, the model expects to "memorize" the training data, losing its generalization capacity and leading to reduced recognition rates when confronted with new data. One can acknowledge such a problem as overfitting.

Numerous attempts have been engaged in order to lessen the overfitting problem in classification tasks, such as early-stopping training or even introducing regularization methods such as soft-weight sharing [4], L1 [5], and L2 [6], DropConnect [7], among others. Alternatively, the best way to employ a regularization method would be to average the predictions of all possible parameter configurations, weighing the possibilities and checking out which would perform better. Nevertheless, such a methodology demands a cumbersome computational effort, only feasible for pitiful or non-complex models [8].

Some years ago, an regularization approach known as Dropout was proposed by Srivastava et al. [9] and aimed to turn off learning neurons using a random Bernoulli distribution. In other words, neurons and their outgoing and incoming connections are temporarily removed from the network according to a probability, allowing the evaluation of distinct sub-architectures and providing more robust training knowledge. Although it seems a straightforward method, the problem lies in that neurons are randomly dropped based only on a probability value (p), not taking advantage of valuable information related to the model itself. Also, the p value have to be carefully chosen, since high probabilities of shutting off neurons may negatively impact the learning process.

Therefore, we aim to address such a problem through an energy-based Dropout, which creates a relationship between the system's neurons and its energy, removing standard Dropout's hyper-parameter (p) and the aleatory behavior while feeding in more robust information about the learning process itself.

In a nutshell, the main contributions of this paper are threefold: (i) to introduce a new type of regularization based on the model's energy, (ii) to introduce an energy-based Dropout in the context of RBMs, and (iii) to fill the lack of research regarding Dropout-based regularizations in RBMs. The remainder of this paper is organized as follows. Section II presents some studies and theoretical background concerning Dropout. Section III explains the energy-based Dropout, while Section IV presents the central concepts of RBM, Dropout RBM, and energy-based Dropout RBM. Section V discusses the experimental setup employed in this work, while Section VI presents the experimental results. Finally, Section VII

states conclusions and future works.

II. BACKGROUND AND RELATED WORK

Dropout is a probability-based method [9] that decides whether a set of neurons should be dropped or not. This section presents the main concepts regarding such an approach and studies concerning such a regularization method.

A. Related Works

Only a few recent studies have addressed the RBMs' overfitting problem with Dropout-based regularization. For instance, Wang et al. [10] have introduced a fast version of Dropout, but not aiming RBMs as their primary focus. The proposed approach is employed in classification and regression tasks and works by sampling from a Gaussian approximation instead of applying the Monte Carlo "optimization".

Ba et al. [11] proposed an adaptive Dropout for training deep neural networks, which is achieved by computing local expectations of binary dropout variables and by calculating derivatives using backpropagation and stochastic gradient descent. The experiments showed that the method achieved low misclassification rates in the MNIST and NORB datasets, highly competitive with CNNs.

Su et al. [12] introduced a Dropout-based RBM considering field-programmable gate arrays, enabling improved implementation and hardware efficiency. Additionally, Wang et al. [13] presented an extensive review of different regularization methods in the context of RBMs, such as weight decay, network pruning, and Dropconnect. Although all these methods have obtained state-of-the-art results in some applications, their main drawback concerns setting up parameters.

Tomczak [14] employed different regularization methods for RBMs to improve their classification and generalization performance. In the experiments, the application of the considered regularization techniques did not result in any improvement. Nevertheless, when combining the information-theoretic regularization and the reconstruction cost, the proposed approach improved the log-probabilities.

In summary, RBM-related works show that when the main task is classification, such technique takes little advantages from the Dropout regularizer. On the other hand, it may boost the unsupervised learning, increasing the log-probabilities, and providing robustness data reconstruction. Considering that an RBM has a simple architecture, connections can quickly saturate, thus forcing the latent space to learn only the more prominent features from the data, which cause difficulties in data generalization and generation. It is interesting to employ an advanced regularization method, as the proposed approach, such that the energy associated with the latent representation indicates which hidden neurons need to be off to encourage others to learn more.

B. Dropout Regularization

Dropout is a robust regularization method with a low computational cost that evaluates countless sub-architectures by randomly dropping out some neurons along the training process. Such a heuristic inhibits units from learning their neighbors' mistakes or "memorizing" the input data, been widely employed for classification tasks. Figure 1 illustrates examples of both standard and Dropout network architectures.
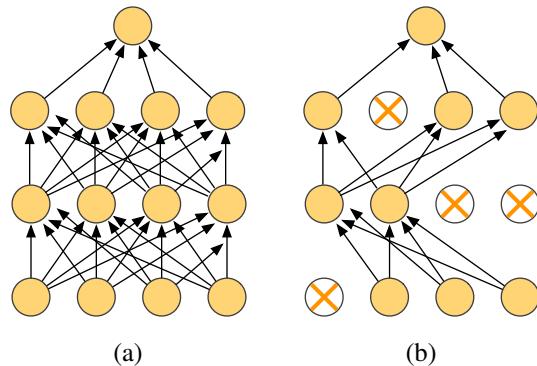

Fig. 1. Examples of: (a) standard network architecture and (b) a Dropout network architecture.

Furthermore, it is straightforward to elucidate the mathematical foundations of Dropout. Let r be a vector of n neurons of a specific layer L, where each variable r_i, $i = \{1, 2, \ldots, n\}$, assumes the value 0 (zero) with probability p, regardless of other variables r_j, $j = \{1, 2, \ldots, n\}$, where $i \neq j$. If $r_i = 0$, the ith unit from the layer L is temporarily switched-off alongside with its connections, while the unit is held when $r_i = 1$.

Notice the probability p is sampled directly from a Bernoulli distribution [9], as follows:

$$r_i \sim Bernoulli(p), \forall i = \{1, 2, \ldots, n\}. \tag{1}$$

Besides, such a probability value is re-sampled for every batch during training.

Let γ be the network activation function and $\boldsymbol{W}^L \in \Re^{m \times n}$ the weight matrix in a specific layer L. The activation vector $\boldsymbol{y}^L \in \Re^n$ can be formulated as follows:

$$\boldsymbol{y}^L = \gamma(\boldsymbol{W}^L \boldsymbol{x}^L), \tag{2}$$

where $\boldsymbol{x}^L \in \Re^m$ is the input from layer L.

In order to consider the dropout of neurons in this layer, the previous equation can be extended to the following:

$$\boldsymbol{y}^L = \boldsymbol{r} * \gamma(\boldsymbol{W}^L \boldsymbol{x}^L), \tag{3}$$

where $*$ stands for the point-wise operator.

Notably, the Dropout regularization provides training based on all possible 2^n sub-networks, as neurons are randomly shut down according to a probability p. Nevertheless, at the inference time (testing step), the weight matrix $\boldsymbol{W}^L$ needs to be re-scaled with p in order to consider all possible sub-networks, as follows:

$$\tilde{\boldsymbol{W}}^L = p\boldsymbol{W}^L. \tag{4}$$

III. ENERGY-BASED DROPOUT

In this section, we present the proposed approach denoted as energy-based Dropout (E-Dropout), which establishes a straightforward relationship between hidden neurons and the system's energy, hereinafter denoted "Importance Level" ($\mathcal{I}$). The idea is to take advantage of the model's behavior for further enabling a more conscious decision whether a set of neurons should be dropped or not.

Let $\mathcal{I}^L \in \Re^n$ be the Importance Level of the hidden neurons at a specific layer L, which directly correlates the hidden probabilities with the RBM total energy. One can define $\mathcal{I}^L$ as follows:

$$\mathcal{I}^L = \frac{\left(\dfrac{P_{tr}(\boldsymbol{x}^L = 1)}{P_i(\boldsymbol{x}^L = 1)}\right)}{|\Delta E|}, \quad (5)$$

where $P_{tr}(\boldsymbol{x}^L = 1)$ represents the probability of activating hidden neurons in layer L after the training procedure, and $P_i(\boldsymbol{x}^L = 1)$ stands for the activation probability of the hidden neurons in layer L given the input data $\boldsymbol{x}$ only, i.e., before training. Finally, $|\Delta E|$ represents the absolute value of the system's energy variation, i.e., the energy after training subtracted from the initial energy measured.

The main intuition behind such a relationship derives from the RBM's energy, in which the hidden configuration participate directly to the total energy, as shown in Equation 8 in the next section. The idea is to represent a gain or loss in information by applying a ratio between the pre- and post-neurons activation. Looking towards Equation 5, one can observe an innovative way to model the relationship between neuron probability and the system's energy. In short, the meaning of a hidden neuron in the model is proportional to its importance level.

After computing $\mathcal{I}^{\mathcal{L}}$ for each hidden neuron, it is possible to obtain the Dropout mask $\boldsymbol{s}$ by comparing it with a uniformly distributed random vector as follows:

$$\boldsymbol{s} = \begin{cases} 1, & \text{if } \mathcal{I}^L < \boldsymbol{u} \\ 0, & \text{otherwise}, \end{cases} \quad (6)$$

where $\boldsymbol{u} \in \Re^n$ is a uniformly distributed random vector, i.e., $\boldsymbol{u} \in [0, 1)$. Furthermore, one can calculate the activation vector $\boldsymbol{y}^L$ as follows:

$$\boldsymbol{y}^L = \boldsymbol{s} * \gamma(\boldsymbol{W}^L \boldsymbol{x}^L). \quad (7)$$

It is crucial to highlight that neurons tend to increase or decrease their importance level during the learning process based on the information acquired from the data distribution, where a neuron is less likely to be dropped out when its importance assumes a higher value. Additionally, when the system's energy is close to zero (more accurate data distribution learning), the energy-based Dropout allows a continuous drop out of neurons to learn additional information. Finally, during the inference phase, it is unnecessary to re-scale the weight matrix.

IV. RESTRICTED BOLTZMANN MACHINES

Restricted Boltzmann Machines [15] are stochastic neural networks that deal with unlabeled data efficiently. In other words, RBMs are a suitable approach for unsupervised problems such as image reconstruction, feature extraction, pre-training deep networks, and collaborative filtering.

Such networks are modeled as bipartite graphs and parametrized by physical concepts like energy and entropy. Thereby, RBMs have a simple architecture with two binary-valued layers: the visible layer $\boldsymbol{v}$ with m units, and the hidden layer $\boldsymbol{h}$ with n units. Each connection between a visible v_i and a hidden unit h_j is weighted by w_{ij}. The weight matrix $\boldsymbol{W}_{m \times n}$ retains the knowledge of the network[1]. Figure 2 shows the standard architecture of an RBM.
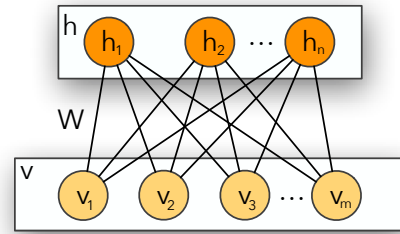

Fig. 2. The standard RBM architecture.

While the visible layer handles the data, the hidden layer performs the feature extraction by detecting patterns and learning the data distribution in a probabilistic manner. Equation 8 describes the energy function of an RBM, where $\boldsymbol{a} \in \Re^m$ and $\boldsymbol{b} \in \Re^n$ stand for the biases of visible and hidden units, respectively:

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{n} b_j h_j - \sum_{i=1}^{m} \sum_{j=1}^{n} v_i h_j w_{ij}. \quad (8)$$

In addition, the joint probability of an arrangement $(\boldsymbol{v}, \boldsymbol{h})$ can be modeled as follows:

$$P(\boldsymbol{v}, \boldsymbol{h}) = \frac{e^{-E(\boldsymbol{v}, \boldsymbol{h})}}{Z}, \quad (9)$$

where Z is the partition function, which is a normalization term for the probability over all possible visible and hidden states. Moreover, the marginal probability of an input vector is represented as follows:

$$P(\boldsymbol{v}) = \frac{\sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v}, \boldsymbol{h})}}{Z}. \quad (10)$$

As in bipartite graph and in an undirected model, the activations for both units (visible and hidden) are mutually independent. Therefore, the formulation of their conditional probabilities is straightforward, being defined by as follows:

---

[1]Since RBMs have one hidden layer only, we omitted the layer index L.

$$P(\boldsymbol{v}|\boldsymbol{h}) = \prod_{i=1}^{m} P(v_i|\boldsymbol{h}), \tag{11}$$

and

$$P(\boldsymbol{h}|\boldsymbol{v}) = \prod_{j=1}^{n} P(h_j|\boldsymbol{v}), \tag{12}$$

where $P(\boldsymbol{v}|\boldsymbol{h})$ and $P(\boldsymbol{h}|\boldsymbol{v})$ represent the probability of the visible layer given the hidden states and the probability of the hidden layer given the visible states, respectively.

From Equations 11 and 12, we can derive the probability of a single active visible neuron i given the hidden states, and the probability of a single active hidden neuron j given the visible states, as follows:

$$P(v_i = 1|\boldsymbol{h}) = \sigma\left(\sum_{j=1}^{n} w_{ij} h_j + a_i\right), \tag{13}$$

and

$$P(h_j = 1|\boldsymbol{v}) = \sigma\left(\sum_{i=1}^{m} w_{ij} v_i + b_j\right), \tag{14}$$

where $\sigma(\cdot)$ stands for the logistic-sigmoid function.

Essentially, an RBM learns a set of parameters $\theta = (\boldsymbol{W}, \boldsymbol{a}, \boldsymbol{b})$ during the training process. Such task can be modeled as an optimization problem aiming to maximize the product of data probabilities for all training set $\mathcal{V}$, as follows:

$$\arg\max_{\Theta} \prod_{\boldsymbol{v} \in \mathcal{V}} P(\boldsymbol{v}). \tag{15}$$

Such a problem is commonly treated by applying the negative of the logarithm function, known as the Negative Log-Likelihood (NLL), which represents the approximation of the reconstructed data regarding the original data distribution. Therefore, it is possible to take the partial derivatives of $\boldsymbol{W}$, $\boldsymbol{a}$ and $\boldsymbol{b}$ at iteration t. Equations 16, 17 and 18 describe the update rules for this set of parameters:

$$\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} + \eta(\boldsymbol{v}P(\boldsymbol{h}|\boldsymbol{v}) - \tilde{\boldsymbol{v}}P(\tilde{\boldsymbol{h}}|\tilde{\boldsymbol{v}})), \tag{16}$$

$$\boldsymbol{a}^{(t+1)} = \boldsymbol{a}^{(t)} + (\boldsymbol{v} - \tilde{\boldsymbol{v}}), \tag{17}$$

and

$$\boldsymbol{b}^{(t+1)} = \boldsymbol{b}^{(t)} + (P(\boldsymbol{h}|\boldsymbol{v}) - P(\tilde{\boldsymbol{h}}|\tilde{\boldsymbol{v}})), \tag{18}$$

where η is the learning rate, $\tilde{\boldsymbol{v}}$ stands for the reconstructed input data given $\boldsymbol{h}$, and $\tilde{\boldsymbol{h}}$ represents an estimation of the hidden vector $\boldsymbol{h}$ given $\tilde{\boldsymbol{v}}$.

Hinton et al. [15] proposed one of the most efficient ways to train an RBM and estimate the visible and hidden layers, known as the Contrastive Divergence (CD). Such an approach uses Gibbs sampling to infer the neurons' states, initializing the visible units with the training data.

A. Dropout RBMs

Considering the concepts mentioned above, a Dropout RBM can be formulated as a simple RBM extended with one binary random vector $\boldsymbol{r} \in \{0, 1\}^n$. In this new formulation, $\boldsymbol{r}$ stands for the activation or dropout of the neurons in the hidden layer, where each variable r_i determines whether the neuron h_i is going to be dropped out or not. Figure 3 illustrates such an idea, in which the hidden unit h_2 is shutoff.
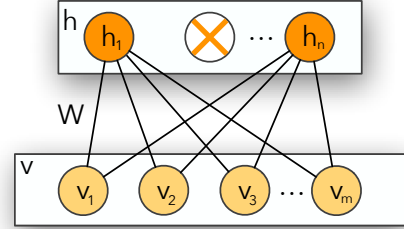

Fig. 3. The Dropout-based RBM architecture.

Notice that $\boldsymbol{r}$ is re-sampled for every mini-batch during learning. As units were dropped from the hidden layer, Equation 14 can be rewritten as follows:

$$P(h_j = 1|\boldsymbol{r}, \boldsymbol{v}) = \begin{cases} 0, & \text{if } r_j = 0 \\ \sigma\left(\sum_{i=1}^{m} W_{ij} v_i + b_j\right), & \text{otherwise.} \end{cases} \tag{19}$$

Therefore, a Dropout RBM can be understood as a blend of several RBMs, each one using different subsets of their hidden layers. As we are training the model with different subsets, the weight matrix $\boldsymbol{W}$ needs to be scaled at testing time, being multiplied by p in order to adjust its weights (Equation 4).

B. E-Dropout RBMs

As aforementioned in Section III, one can use Equation 5 to calculate the importance level $\mathcal{I}$ of the hidden neurons. Nevertheless, when dealing with an E-Dropout RBM, as the system's energy approximates to zero, $\mathcal{I}$ tends to overflow with large values. Therefore, it is necessary to re-scale $\mathcal{I}$ between 0 and 1 as follows:

$$\mathcal{I} = \frac{\mathcal{I}}{\max\{\mathcal{I}\}}. \tag{20}$$

After computing $\mathcal{I}$, one can use Equation 6 to calculate the Dropout mask $\boldsymbol{s}$. Therefore, Equation 14 can be rewritten as follows:

$$P(h_j = 1|\boldsymbol{s}, \boldsymbol{v}) = \begin{cases} 0, & \text{if } s_j = 0 \\ \sigma\left(\sum_{i=1}^{m} W_{ij} v_i + b_j\right), & \text{otherwise.} \end{cases} \tag{21}$$

Furthermore, it is worth using mini-batches while training the network, which can be accomplished by calculating Equation 20 for every sample in the mini-batch followed by its average.

V. Experiments

In this section, we present the methodological setup used to evaluate the E-Dropout considering RBMs[2] in the task of binary image reconstruction. Besides, we compare the proposed method against a standard-Dropout, RBMs without Dropout, among others, and describe the employed datasets and the experimental setup.

A. Modeling E-Dropout RBMs

As aforementioned in Section III, the energy-based Dropout uses Equation 5 to calculate an importance level $\mathcal{I}$ for each neuron. Additionally, it computes the dropout mask s using Equation 6. Finally, it uses s in the same way as the standard Dropout method. Note that we consider the very same fundamental concepts presented in Section IV.

B. Datasets

Three well-known image datasets were employed throughout the experiments:

- MNIST[3] [17]: set of 28×28 grayscale images of handwritten digits (0-9), i.e., 10 classes. 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;
- Fashion-MNIST[4] [18]: set of 28×28 grayscale images of clothing objects. The original version contains a training set with $60,000$ images from 10 distinct objects (t-shirt, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot), and a test set with $10,000$ images;
- Kuzushiji-MNIST[5] [19]: set of 28×28 grayscale images of hiragana characters. The original version contains a training set with $60,000$ images from 10 previously selected hiragana characters, and a test set with $10,000$ images.

C. Experimental Setup

Concerning the experimental setup, we employed five different RBM architectures, in which the main difference lies in the regularization method. In this case, RBM does not employ Dropout, the "Weight" RBM (W-RBM) with L2 regularization employing a penalty of $5 \cdot 10^{-3}$ (the mean value of the ranges proposed by Hinton [2]), the "standard-dropout" RBM (D-RBM) uses the traditional Dropout, the "DropConnect" RBM (DC-RBM) uses the traditional DropConnect, and the "E-Dropout" RBM (E-RBM) employs the proposed energy-based Dropout. Additionally, when considering the "standard-dropout" and the "DropConnect", we used $p = 0.5$ as stated by Srivastava et al. [9] and Wan et al. [7], respectively.

Since the learning rate and the number of hidden neurons are important hyperparameters of an RBM, we fixed each RBM according to Table I, in which four different models have been considered, i.e., M_a, M_b, M_c, M_d. To provide more shreds

of evidence of the E-Dropout suitability, we employed four distinct architectures, differing only in the number of hidden neurons and learning rates.

Notice that three out of four architectures have $1,024$ hidden neurons. The reason is that RBMs with more feature detector units have more chances to learn unimportant information from the data distribution. Moreover, we decreased the learning rate to verify the E-Dropout ability to improve significantly when the network learns slowly.

Furthermore, we have considered 50 epochs for the RBM learning procedure with mini-batches of size 256, while all RBMs were trained using the Contrastive Divergence algorithm with $k = 1$ (CD-1).

TABLE I
RBM HYPERPARAMETERS CONFIGURATION.

Parameter	$\mathbf{M_a}$	$\mathbf{M_b}$	$\mathbf{M_c}$	$\mathbf{M_d}$
n (hidden neurons)	512	1,024	1,024	1,024
η (learning rate)	0.1	0.1	0.03	0.01

Two distinct metrics assessed the performance of the models on the test set, i.e., the Mean Squared Error (MSE) and the Structural Similarity Index (SSIM) [20]. The former is often known as the reconstruction error, which encodes the quality of the pixels reconstructed by the RBMs. In contrast, the latter provides a more efficient analysis of the image structure itself, which compares the quality between the original and reconstructed images.

To provide robust statistical analysis and acknowledge that the experiments' results are independent and continuous over a particular dependent variable (e.g., number of observations), we identified the Wilcoxon signed-rank test [21] satisfied our obligations. It is a non-parametric hypothesis test used to compare two or more related observations (in our case, repeated measurements of the MSE and SSIM values) to assess whether there are statistically significant differences between them or not. Therefore, we evaluated different RBM models with distinct Dropout methods ten times to mitigate the RBMs' stochastic nature for every dataset and architecture. Notice the statistical evaluation considers each model at once.

Finally, all the experiments were run in a desktop computer with 16 Gb of RAM ($2,400$MHz clock), an AMD processor containing six cores with 3 GHz of a clock, and a video card (GPU) GTX 1060 with 6 Gb of memory.

VI. Experimental Results

This section presents the experimental results concerning the E-Dropout RBM, D-RBM, W-RBM, and RBM, considering three well-known literature datasets.

A. MNIST

Considering the MNIST dataset, Table II exhibits the mean reconstruction errors and their respective standard deviation over the testing set, where the best results are in bold according to the Wilcoxon signed-rank test. Considering model M_b and

M_c, the E-RBM could not obtain better results as the RBM, while for models M_a and M_d, we can highlight that E-RBM was more accurate than RBM, W-RBM, D-RBM, and DC-RBM. Furthermore, these achievements evidence that E-Dropout is less sensitive to different learning rates.

Table III exhibits the mean SSIM and their respective standard deviation over all experiments, where the best results are in bold according to the Wilcoxon signed-rank test. Considering model M_b and M_c, the E-RBM obtained, statistically, the same results as the RBM, while for models M_a and M_d, the E-RBM was significantly better than the RBM, W-RBM, D-RBM, and DC-RBM. It is interesting to note that the proposed approach overpass the other regularizers methods for all models, also for the architecture with $1,024$ hidden neurons and the lowest learning rate, the E-Dropout supported a 2.5% performance improvement on SSIM, in front of an RBM (the second-best model).

In summary, one can notice that E-RBM performed better than all the baselines with regularization. Additionally, Figure 4 depicts the mean reconstruction error over the training set only for models that employ Dropout regularization and the naive version (RBM) since such comparison stands for the work focus and more curves generate visually unattractive graphics. One can observe that most of the RBM models achieved better results than the D-RBM considering the same number of hidden neurons and learning rate. Nevertheless, for the models M_a and M_b, the E-RBM achieved better reconstruction errors, besides converging faster at the first iterations.
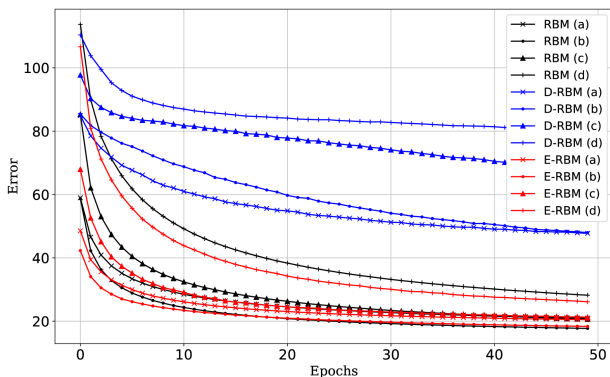

Fig. 4. Mean reconstruction error over the MNIST training set.

Figure 5 depicts the mean SSIM over the testing set for both Dropout methods and the RBM naive version regarding all models. One crucial point to highlight is that all RBM and E-RBM models achieved better results than the D-RBM ones, probably due to the latter "constant" neurons shutdown. Moreover, the E-Dropout achieved the best SSIM considering models M_a, M_b, and M_c, thus fostering the proposed regularization technique.

B. Fashion-MNIST

Regarding the Fashion-MNIST dataset, Table IV exhibits the mean reconstruction errors and their respective standard deviation over all experiments. Considering the E-RBM, it is
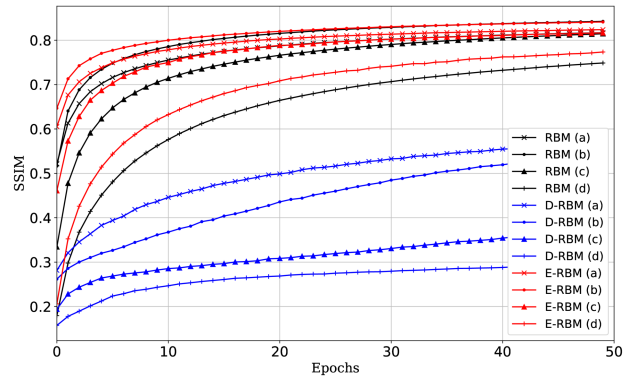

Fig. 5. Mean structural similarity index over the MNIST testing set.

clear its superiority regarding the baselines, once it achieved the lowest errors overall RBM architectures. We can highlight the performance on model M_d, which was 5.77% better than standard RBM.

Table V exhibits the results concerning the SSIM measure, been the best ones according to the Wilcoxon's signed-rank test in bold. We can observe that the E-RBM achieved better results than other baselines for models M_b and M_c (similar to DC-RBM). Surprisingly, the DC-RBM achieved better results regarding models M_a, M_c, and M_d, which was unexpected since such behavior was not observed in Table IV.

Additionally, Figure 6 depicts the mean reconstruction error for all architectures that employ Dropout and its naive version (RBM). One can note that the standard Dropout technique disturbed the RBM learning step. Moreover, the E-RBM achieved the best results in all models and the lowest reconstruction errors.
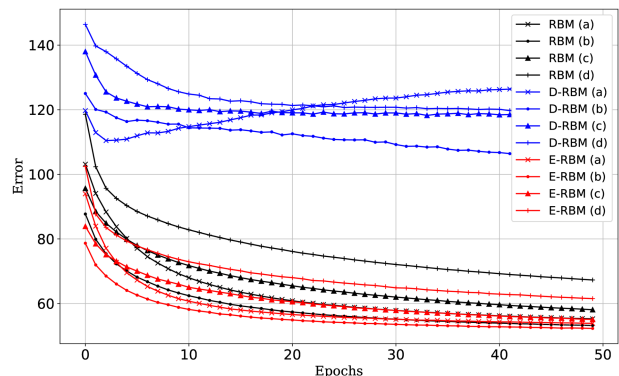

Fig. 6. Mean reconstruction error over Fashion-MNIST training set.

In the same manner, but assessing the models' performance by the SSIM measure (Figure 7), we can confirm that D-RBM was not able to be competitive against the energy-based Dropout and the RBM. Besides, E-RBM obtained better results than the RBMs without Dropout, strengthening its capability to increase the network's learning power.

C. Kuzushiji-MNIST

Considering the Kuzushiji-MNIST dataset, from Table VI, one can see that the E-RBM achieved the lowest mean

TABLE II
MEAN RECONSTRUCTION ERRORS AND THEIR RESPECTIVE STANDARD DEVIATION ON MNIST TESTING DATASET.

Technique	M_a	M_b	M_c	M_d
RBM [15]	20.968 ± 0.059	$\mathbf{17.728 \pm 0.040}$	$\mathbf{20.644 \pm 0.039}$	28.214 ± 0.068
W-RBM [2]	35.460 ± 0.126	31.753 ± 0.125	33.235 ± 0.062	39.583 ± 0.046
D-RBM [9]	47.757 ± 0.242	47.968 ± 0.267	67.572 ± 0.0339	80.155 ± 0.244
DC-RBM [7]	26.470 ± 0.134	25.595 ± 0.235	28.437 ± 0.142	35.856 ± 0.146
E-RBM	$\mathbf{20.511 \pm 0.061}$	18.369 ± 0.072	21.277 ± 0.050	$\mathbf{26.14 \pm 0.174}$

TABLE III
MEAN SSIM VALUES AND THEIR RESPECTIVE STANDARD DEVIATION ON MNIST TESTING DATASET.

Technique	M_a	M_b	M_c	M_d
RBM [15]	0.8170 ± 0.001	$\mathbf{0.8410 \pm 0.001}$	$\mathbf{0.8150 \pm 0.000}$	0.7490 ± 0.001
W-RBM [2]	0.7243 ± 0.001	0.7500 ± 0.001	0.7367 ± 0.001	0.6884 ± 0.001
D-RBM [9]	0.5690 ± 0.002	0.5460 ± 0.003	0.3750 ± 0.002	0.2950 ± 0.001
DC-RBM [7]	0.7684 ± 0.001	0.7659 ± 0.002	0.7468 ± 0.001	0.6934 ± 0.001
E-RBM	$\mathbf{0.8240 \pm 0.000}$	$\mathbf{0.8410 \pm 0.000}$	$\mathbf{0.8160 \pm 0.001}$	$\mathbf{0.7740 \pm 0.002}$

TABLE IV
MEAN RECONSTRUCTION ERRORS AND THEIR RESPECTIVE STANDARD DEVIATION ON FASHION-MNIST TESTING SET.

Technique	M_a	M_b	M_c	M_d
RBM [15]	55.258 ± 0.097	53.204 ± 0.075	58.077 ± 0.066	67.293 ± 0.070
W-RBM [2]	66.195 ± 0.238	59.660 ± 0.152	62.769 ± 0.115	73.732 ± 0.114
D-RBM [9]	127.76 ± 0.694	104.78 ± 0.722	118.52 ± 0.782	119.95 ± 0.497
DC-RBM [7]	71.161 ± 0.105	68.983 ± 0.146	73.101 ± 0.190	80.538 ± 0.205
E-RBM	$\mathbf{53.858 \pm 0.180}$	$\mathbf{52.288 \pm 0.095}$	$\mathbf{55.064 \pm 0.091}$	$\mathbf{61.52 \pm 0.085}$

TABLE V
MEAN SSIM VALUES AND THEIR RESPECTIVE STANDARD DEVIATION ON FASHION-MNIST TESTING SET.

Technique	M_a	M_b	M_c	M_d
RBM [15]	0.5630 ± 0.001	0.5940 ± 0.000	0.5410 ± 0.000	0.4760 ± 0.000
W-RBM [2]	0.5295 ± 0.001	0.5608 ± 0.001	0.5463 ± 0.001	0.4913 ± 0.001
D-RBM [9]	0.2010 ± 0.002	0.2570 ± 0.002	0.2220 ± 0.002	0.2130 ± 0.001
DC-RBM [7]	$\mathbf{0.5791 \pm 0.001}$	0.5856 ± 0.001	$\mathbf{0.5729 \pm 0.001}$	$\mathbf{0.5405 \pm 0.001}$
E-RBM	0.5760 ± 0.001	$\mathbf{0.6130 \pm 0.001}$	$\mathbf{0.5710 \pm 0.001}$	0.5150 ± 0.001

reconstruction errors for settings M_a and M_d. For the model M_b, RBM was the best in front of the employed architectures, while for M_c, the difference between E-RBM and RBM was not significant. On the other hand, for model M_d, the E-RBM had a performance improvement of 3.32% compared to RBM.

Additionally, Table VII exhibits the results for the SSIM metric. One can see that E-RBM kept the same previous behavior, meaning that for models M_a, and M_d, it achieved better results than the other baselines, while for model M_b RBM and E-RBM have no statistical difference.

Moreover, Figure 8 depicts the mean reconstruction error over the training set for all the Dropout-based models and its naive version, respectively. In this particular dataset, the MSE was considerably lower than Fashion-MNIST's ones, even though its digits seem more complex and have fewer details than Fashion-MNIST objects.

TABLE VI
MEAN RECONSTRUCTION ERROR AND THEIR RESPECTIVE STANDARD DEVIATION ON KUZUSHIJI-MNIST.

Technique	M_a	M_b	M_c	M_d
RBM [15]	46.470 ± 0.121	$\mathbf{37.587 \pm 0.064}$	$\mathbf{43.38 \pm 0.070}$	58.262 ± 0.062
W-RBM [2]	76.839 ± 0.139	67.470 ± 0.139	70.537 ± 0.059	83.548 ± 0.061
D-RBM [9]	89.810 ± 0.249	80.475 ± 0.207	93.291 ± 0.189	109.436 ± 0.085
DC-RBM [7]	60.703 ± 0.149	53.330 ± 0.219	58.538 ± 0.141	74.056 ± 0.186
E-RBM	$\mathbf{44.853 \pm 0.133}$	38.511 ± 0.086	$\mathbf{43.544 \pm 0.074}$	$\mathbf{54.937 \pm 0.143}$

TABLE VII
MEAN SSIM AND THEIR RESPECTIVE STANDARD DEVIATION ON KUZUSHIJI-MNIST.

Technique	M_a	M_b	M_c	M_d
RBM [15]	0.6910 ± 0.001	$\mathbf{0.7480 \pm 0.001}$	0.7040 ± 0.000	0.6060 ± 0.001
W-RBM [2]	0.5675 ± 0.001	0.6135 ± 0.001	0.5948 ± 0.001	0.5262 ± 0.001
D-RBM [9]	0.3890 ± 0.002	0.4290 ± 0.002	0.2970 ± 0.002	0.2040 ± 0.000
DC-RBM [7]	0.6561 ± 0.001	0.6703 ± 0.001	0.6577 ± 0.001	0.5933 ± 0.001
E-RBM	$\mathbf{0.7040 \pm 0.001}$	$\mathbf{0.7480 \pm 0.000}$	$\mathbf{0.7150 \pm 0.000}$	$\mathbf{0.6370 \pm 0.000}$

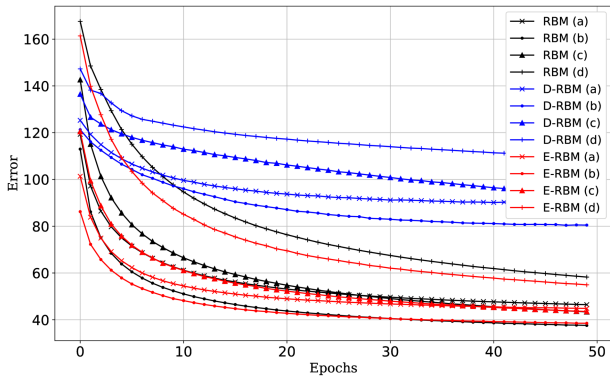Fig. 7. Mean structural similarity index over Fashion-MNIST testing set.

Fig. 8. Mean reconstruction error over Kuzushiji-MNIST's training set.

Furthermore, Figure 9 exhibits the mean SSIM over the testing set considering the same approach that Figure 8. In

this particular dataset, the E-RBM achieved almost the same performance as the RBM, for all configurations of n (number of hidden neurons) and η (learning rate).
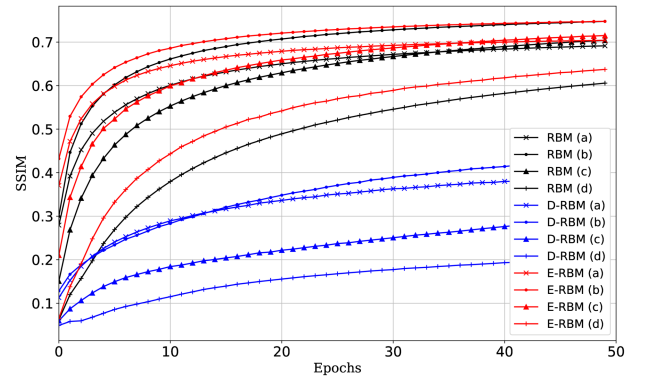

Fig. 9. Mean structural similarity index over Kuzushiji-MNIST's testing set.

D. Dropout Overall Discussion

In addition to the reconstruction error and the visual quality of the reconstructed image assessed by SSIM, in this section, we analyze the behavior of E-Dropout related to the number of neurons dropped out over the training epochs and the weights learned by the E-RBM, D-RBM, and RBM models since it is the only ones employing neuron deactivation, in addition the original RBM.

The third architecture (M_a) is used here to illustrate how the E-Dropout affects the number of neurons turned off in the training process for the three datasets, as shown in Figures 10, 11 and 12. For clarity, D-RBM tends to turn off $60,000$ ($n * p * 60,000\_images/mini - batch$) neurons on

every epoch, while E-RBM considers the neurons activation and the system energy batch-by-batch, and therefore, does not have a "mean value".
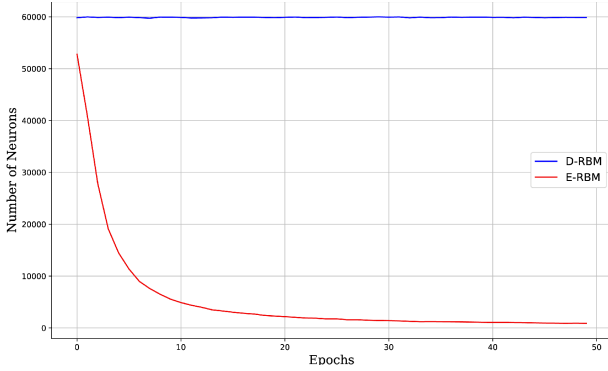

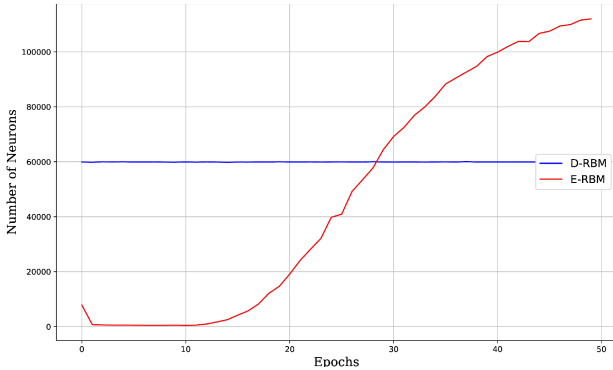Fig. 10. Mean number of neurons dropped over MNIST's training set.

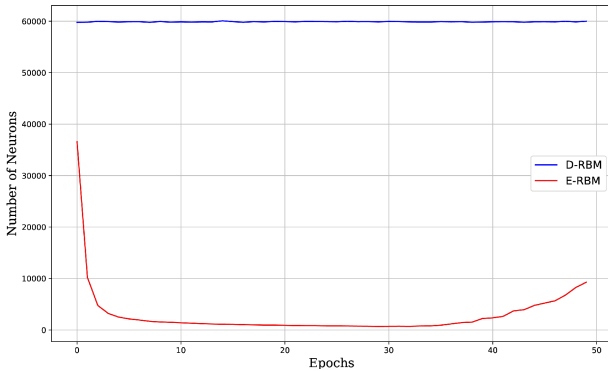Fig. 11. Mean number of neurons dropped over Fashion-MNIST's training set.

Fig. 12. Mean number of neurons dropped over Kuzushiji-MNIST's training set.

These results indicate that E-Dropout behavior depends on the dataset since it considers the relationship between neurons activation and the system's energy derived from the data itself.

Considering the MNIST dataset, the E-Dropout starts by almost turning off the same amount of neurons that the standard Dropout, and slowly decrease this value over the epochs. For the Fashion-MNIST dataset, the E-Dropout starts with almost all neurons and starts dropping them out similar to a sigmoid function shape. On the other hand, considering the

Kuzushiji-MNIST dataset, the E-Dropout starts by turning off approximately $37,000$ neurons and rapidly decreasing these values, while increasing the number of dropped out neurons in the last epochs.
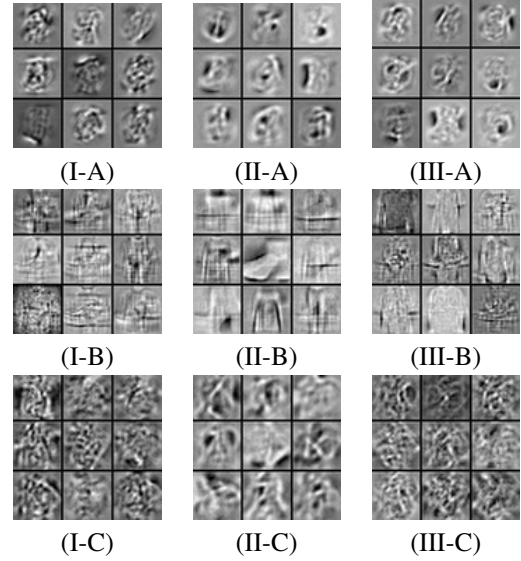

Fig. 13. M_c - MNIST subset of learned weights: (I-A) RBM, (II-A) D-RBM and (III-A) E-RBM. Fashion-MNIST subset of learned weights: (I-B) RBM, (II-B) D-RBM and (III-B) E-RBM. Kuzushiji-MNIST subset of learned weights: (I-C) RBM, (II-C) D-RBM and (III-C) E-RBM.

Regarding the weights learned by the models, Figure 13 depict a subset of model M_c for MNIST (I-A, II-A, III-A), Fashion-MNIST (I-B, II-B, III-B), and Kuzushiji-MNIST (I-C, II-C, III-C) datasets, respectively. Overall, the D-RBM provides some sparsity but less "clear" weights, representing any images' details. The RBM provides a fair representation of these details, and even though, in some cases, it is clear that its weights are less informative. Finally, the E-RBM portrays more accurate images representation, mainly the high-frequency ones, such as the inner drawings.

Additionally, one can establish a parallel with the temperature regularization effect showed by [22] and [23], in which low temperatures forces de connections to small values, providing network sparsity at the step that improves the lower bound in the learning process. Such behavior is interesting since the E-RBM was encouraged to prevent co-adaptations selectively.

Furthermore, Figures 14, 15 and 16 depict a subset of model M_c reconstructed images over MNIST, Fashion-MNIST, and Kuzushiji-MNIST datasets, respectively.
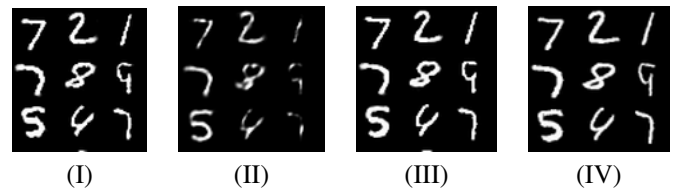

Fig. 14. M_c - MNIST subset of reconstructed images: (I) RBM, (II) D-RBM, (III) E-RBM and (IV) Original.

Finally, Table VIII shows the computational burden over all methods and architectures, regarding 50 epoch of training. It
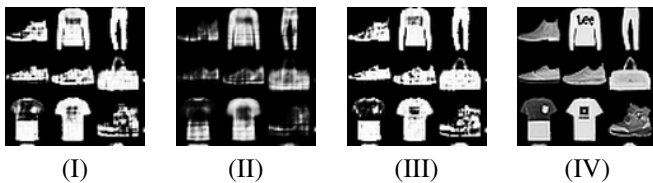
Fig. 15. M_c - Fashion-MNIST subset of reconstructed images: (I) RBM, (II) D-RBM, (III) E-RBM and (IV) Original.
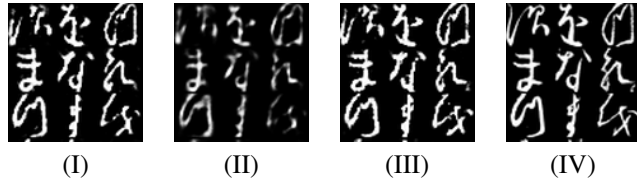

Fig. 16. M_c - Kuzushiji-MNIST subset of reconstructed images: (I) RBM, (II) D-RBM, (III) E-RBM and (IV) Original.

is essential to highlight that all datasets consume almost the same computational load due to the same characteristics, and, for that, it was summarized in one general table. The mean and standard deviation are from the ten repetitions taken from the experiments.

TABLE VIII
MEAN TIME IN SECONDS AND THEIR RESPECTIVE STANDARD DEVIATION.

Technique	M_a	M_b	M_c	M_d
RBM [15]	250 ± 3	250 ± 3	250 ± 3	250 ± 3
W-RBM [2]	250 ± 3	250 ± 3	250 ± 3	250 ± 3
D-RBM [9]	250 ± 3	250 ± 3	250 ± 3	250 ± 3
DC-RBM [7]	1050 ± 3	1100 ± 3	1100 ± 3	1100 ± 3
E-RBM	300 ± 3	300 ± 4	300 ± 4	300 ± 4

Table VIII shows that E-RBM has a little more computational load than RBM, W-RBM, and D-RBM, but considering a high number of training epochs. On the other hand, the DC-RBM was the more power-consume model since the Drop-Connect needs to sample a weight mask for every instance on the mini-batch. In summary, the improvement achieved by the E-RBM in the image reconstruction task depicted in previous sections overcome the slightly worst performance in processing time against the simpler baselines.

VII. CONCLUSION

This article proposed a new regularization method, known as energy-based Dropout, an enhanced parameterless version of the traditional Dropout. Based on physical principles, it creates a direct correlation between the system's energy and its hidden neurons, denoted as Importance Level ($\mathcal{I}$). Furthermore, as Restricted Boltzmann Machines are also physical-based neural networks, they were considered the perfect architecture to validate the proposed approach.

The energy-based Dropout was validated in Restricted Boltzmann Machines through a binary image reconstruction

task. Three well-known literature, datasets, MNIST, Fashion-MNIST, and Kuzushiji-MNIST, were employed to validate the proposed approach. Considering the experimental results discussed in the paper, one can observe that the energy-based Dropout proved to be a suitable regularization technique, obtaining significantly better SSIM rates than its counterpart Dropout in all three datasets. Additionally, when comparing the energy-based Dropout to the standard RBM, it outperformed the latter in two out of three datasets, being slightly worse in the one that it could not achieve the best result. Moreover, it is possible to perceive that the weights learned by the energy-Dropout approach were able to recognize different patterns and high-frequency details, besides had less sharp edges when compared to the standard RBM and Dropout-based RBM.

When comparing all the employed techniques, more demanding tasks benefit more from the energy-based Dropout than easier ones, i.e., tasks with higher reconstruction errors seem to achieved the best result when using the energy-based Dropout. Moreover, when comparing the proposed method and the standard one, the proposed regularization obtained significantly better results, reinforcing its capacity to improve RBMs' learning procedure.

Regarding future works, we aim at expanding some concepts of the energy-Dropout regularization technique to the classification task and other suitable machine learning algorithms, such as Deep Belief Networks (DBNs) and Deep Boltzmann Machines (DBMs).

REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[2] G. E. Hinton, "A practical guide to training restricted boltzmann machines," in *Neural Networks: Tricks of the Trade*, ser. Lecture Notes in Computer Science, G. Montavon, G. Orr, and K.-R. Müller, Eds. Springer Berlin Heidelberg, 2012, vol. 7700, pp. 599–619.

[3] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[4] S. J. Nowlan and G. E. Hinton, "Simplifying neural networks by soft weight-sharing," *Neural Computation*, vol. 4, no. 4, pp. 473–493, July 1992.

[5] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[6] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[7] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *International conference on machine learning*, 2013, pp. 1058–1066.

[8] H. Y. Xiong, Y. Barash, and B. J. Frey, "Bayesian prediction of tissue-regulated splicing using rna sequence and cellular context," *Bioinformatics*, vol. 27, no. 18, pp. 2554–2562, Sep. 2011.

[9] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[10] S. Wang and C. Manning, "Fast dropout training," in *Proceedings of the 30th International Conference on Machine Learning*, 2013, pp. 118–126.

[11] L. J. Ba and B. Frey, "Adaptive dropout for training deep neural networks," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13.   Curran Associates Inc., 2013, pp. 3084–3092.

[12] J. Su, D. B. Thomas, and P. Y. K. Cheung, "Increasing network size and training throughput of fpga restricted boltzmann machines using dropout," in *2016 IEEE 24th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, May 2016, pp. 48–51.

[13] B. Wang and D. Klabjan, "Regularization for unsupervised deep neural nets," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[14] J. M. Tomczak, "Learning informative features from restricted boltzmann machines," *Neural Processing Letters*, vol. 44, no. 3, pp. 735–750, 2016.

[15] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[16] M. Roder, G. H. de Rosa, and J. P. Papa, "Learnergy: Energy-based machine learners," 2020.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[18] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.

[19] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha. (2018) Deep learning for classical japanese literature.

[20] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *Trans. Img. Proc.*, vol. 13, no. 4, pp. 600–612, 2004. [Online]. Available: http://dx.doi.org/10.1109/TIP.2003.819861

[21] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[22] L. A. Passos and J. P. Papa, "Temperature-based deep boltzmann machines," *Neural Processing Letters*, vol. 48, no. 1, pp. 95–107, 2018.

[23] G. Li, L. Deng, Y. Xu, C. Wen, W. Wang, J. Pei, and L. Shi, "Temperature based restricted boltzmann machines," *Scientific reports*, vol. 6, p. 19133, 2016.

Gustavo Henrique de Rosa is a Bachelor in Computer Science at São Paulo State University (UNESP), FC/Bauru (2016), and a former Scientific Initiation FAPESP's scholarship holder with an internship at the Harvard University, focusing on image processing formulations, pattern recognition, pattern classification, machine learning algorithms, and meta-heuristic optimization. Master of Science in Computer Science at São Paulo State University, IBILCE/Rio Preto (2018), and a former Master of Science FAPESP's scholarship holder with an internship at the University of Virginia, focusing on deep learning and meta-heuristic optimization. Currently, as a Ph.D. student in Computer Science at São Paulo State University, FC/Bauru, and a Ph.D. FAPESP's scholarship holder, focusing on natural language processing and adversarial learning. Email: gustavo.rosa@unesp.br
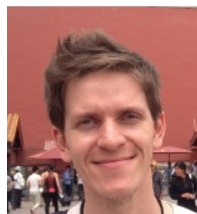
Victor Hugo C. de Albuquerque [M'17, SM'19] is a professor and senior researcher at the ARMTEC Tecnologia em Robótica, Brazil. He has a Ph.D in Mechanical Engineering from the Federal University of Paraíba (UFPB, 2010), an MSc in Teleinformatics Engineering from the Federal University of Ceará (UFC, 2007), and he graduated in Mechatronics Engineering at the Federal Center of Technological Education of Ceará (CEFETCE, 2006). He is a specialist, mainly, in IoT, Machine/Deep Learning, Pattern Recognition, Robotic.

André Rossi André L.D. Rossi received his B.Sc. degree in Computer Science from the Universidade Estadual de Londrina (UEL), Brazil, and his M.Sc. and Ph.D. degrees in Computer Science from University of São Paulo (USP), Brazil. André Rossi is currently an Associate Professor at the São Paulo State University (UNESP), Brazil. His main research interests are Machine Learning and Computer Vision. Email: andre.rossi@unesp.br

João P. Papa [SM'17] received his B.Sc. in Information Systems from the São Paulo State University (UNESP), SP, Brazil. In 2005, he received his M.Sc. in Computer Science from the Federal University of São Carlos, SP, Brazil. In 2008, he received his Ph.D. in Computer Science from the University of Campinas, SP, Brazil. During 2008-2009, he had worked as a post-doctorate researcher at the same institute, and during 2014-2015 he worked as a visiting scholar at Harvard University. He has been a Professor at the Computer Science Department, São Paulo, State University, since 2009. He was also the recipient of the Alexander von Humboldt research fellowship in 2017. Email: joao.papa@unesp.br

Mateus Roder Mateus Roder is a Bachelor in Manufacturing Engineering at São Paulo State University (UNESP), Itapeva-SP (2018), and a former Scientific Initiation FAPESP's scholarship holder, focusing on image processing and classification, machine learning algorithms, and meta-heuristic optimization. Currently, is a student and FAPESP's scholarship holder in Master of Computer Science at São Paulo State University, FC/Bauru, focusing on restricted Boltzmann machines and deep learning. Email: mateus.roder@unesp.br