Mask Selective Regularization for Restricted Boltzmann Machines

Jordi Mansanet, Alberto Albiol, Roberto Paredes, Antonio Albiol

Universitat Politècnica de València, 46022 València, Spain

Abstract

In the present work, we propose to deal with two important issues regarding to the RBM's learning capabilities. First, the topology of the input space, and second, the sparseness of the RBM obtained. One problem of RBMs is that they do not take advantage of the topology of the input space. In order to alleviate this lack, we propose to use a surrogate of the mutual information of the input representation space to build a set of binary masks. This approach is general and not only applicable to images, thus it can be extended to other layers in the standard layer-by-layer unsupervised learning. On the other hand, we propose a selective application of two different regularization terms, L_1 and L_2, in order to ensure the sparseness of the representation and the generalization capabilities. Additionally, another interesting capability of our approach is the adaptation of the topology of the network during the learning phase by means of selecting the best set of binary masks that fit the current weights configuration. The performance of these new ideas is assessed with a set of experiments on different well-known corpus.

Keywords: Restricted Boltzmann Machine, Deep Belief Networks, Regularization

1. Introduction

Representation learning tries to convert data into a form that makes it easier to extract useful information when building classifiers [1]. Among the different approaches for learning representations, this paper focuses on deep learning methods. Deep learning methods work by stacking several layers of non-linear transformations with the objective of yielding more abstract and useful representations.

Restricted Boltzmann Machines (RBM) are powerful generative graphical models that are used in unsupervised learning for modeling data distributions. Recently, RBMs have become very popular particularly since they were proposed to initialize the layers of Deep Belief Networks (DBN) [2]. RBMs model statistical dependencies of observed variables by introducing binary latent variables, which are assumed to be independent given the observed variables.

In the present work, we propose to deal with two important issues regarding to the RBM's learning capabilities. First, the topology of the input space, and second, the sparseness of the RBM obtained. Regarding the first point, one problem of RBMs is that they do not take advantage of the topology of the input space. For instance, in the case of images they model long-range dependencies that are known to be weak [3]. As a result, DBNs initialized with these RBMs are known to be non robust to noise that is not seen in the training set [4]. In order to capture the topology of the input space, we propose to evaluate the mutual information of the components of the input representation space (pixels in case of images). This mutual information is a measure of the dependence expressed in the joint distribution of two input components relative to the joint distribution of these two components under the assumption of independence. However, since mutual information is very difficult to obtain for continuous variables it is common to approximate it by the Pearson's correlation coefficient [5, 6]. In this work, we propose to use this surrogate measure to obtain a weighted vector for each input component w.r.t to the others. These weighted vectors will be considered to obtain regularization masks for the RBMs in the learning process.

Regarding to the second point, sparseness is a recent concept introduced to increase the efficiency and robustness of neural networks. There exist two variants of sparseness [7]: in *sparse activity* only a small fraction of the neurons are active when an input pattern is presented, while in *sparse connectivity* each neuron is connected to only a limited set of neurons. Interestingly, both kind of sparseness have a strong biological inspiration since similar properties have been observed in mammalian brains [8].

Following these two main ideas, we propose the use of a new regularization scheme to train RBMs based on a selective L_2–L_1 regularization approach. This selective approach depends on a set of binary masks derived from a surrogate measure of the mutual information of the input space, as mentioned above. This combination of selective regularization and binary masks enforces sparse connectivity and improves the robustness of DBNs to noise. One key advantage of our approach is that the learning of the topology of the network (the binary mask selection) is included in the training process. Moreover, the definition of the binary masks is general, so they can be used with any kind of data (not only images) and also in higher layers of the DBN where the topology is usually unknown. To prove the validity of our hypothesis we have performed several classification experiments on well-know databases: MNIST, USPS, 20-Newsgroups and CIFAR-10.

2. Related Work

The study of the robustness of deep learning structures is a problem that has attracted the attention of many researchers lately [9, 10, 11]. One way to introduce robustness against noise is to artificially corrupt training data [12, 13]. However, the noise distribution of test data can be unknown during training. Another way to increase the robustness of deep structures is to introduce sparsity in the activation of hidden units [14, 15, 16]. Interestingly, sparse representations are not very useful as generative models, although they have proved to be very successful for unsupervised feature learning [17].

Sparsity can also be introduced in the connectivity of the neurons. Sparse connectivity was introduced many years ago as one of the fundamental ideas used in convolutional neural networks, where local receptive fields connect a small subset of image pixels [18]. Similarly, sparse connectivity has been also introduced in RBMs that model images [4, 19] where the impact area of a hidden unit is restricted to a small patch of the visible image. In contrast to convolutional networks, the weights are not shared between different hidden units. In [4] the robustness of the sparsely connected RBMs was validated using test sets that were corrupted with different noise types. In this paper, we follow a similar methodology to validate the robustness of our regularization scheme against noise.

Although sparse connectivity has shown to be useful for feature learning, its use has been restricted to image data where there is prior information about the topology [20]. However, how to extend its use to other data types remains an ongoing question. In the case of deep architectures, it is still unclear how to extend sparse connectivity to hidden layers where the topology is also unknown. In the present work we aim at providing an efficient solution to this problem.

3. Restricted Boltzmann Machines

A training set of samples can be modeled using a two-layer network called Restricted Boltzmann Machine (RBM). Each dimension of the sample correspond to a "visible" unit. The visible layer is connected to the "hidden" units, which correspond to binary feature detectors. An RBM is an energy model with a function given by:

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

where v_i,h_j are the binary states of visible unit i and hidden unit j, a_i,b_j are their biases and w_{ij} is the weight of the connection between them. The model assigns a probability to every possible pair of visible and hidden vectors through this energy function:

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

where the "partition function", Z, is given by summing over all possible pairs of visible and hidden vectors:

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

The probability that the network assigns to a visible vector, $\mathbf{v}$, is given by summing over all possible hidden vectors:

$$p(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \tag{4}$$

Let $\mathcal{L}\left(\theta, \mathcal{D}\right)$ be the log-likelihood of the data defined as:

$$\mathcal{L}\left(\theta, \mathcal{D}\right) = \sum_{\mathbf{x}_i \in \mathcal{D}} \log p\left(\mathbf{x}_i\right) \tag{5}$$

where θ are the parameters of the model and $\mathbf{x}_i \in R^d$ is a sample of the training set $\mathcal{D}$.

During the training process, the parameters of the model are adjusted so that the log-likehood of the training data is maximized. For this goal, we can perform stochastic gradient descent on the negative log-likelihood function. Therefore, the loss function to be minimized is:

$$\ell\left(\theta, \mathcal{D}\right) = -\mathcal{L}\left(\theta, \mathcal{D}\right) \tag{6}$$

To minimize the loss function, it is necessary to estimate the gradient with respect to the model parameters. An estimation of this gradient can be obtained using a fast learning procedure called Contrastive Divergence (CD) [21].

4. Mask Selective Regularization for RBM

4.1. Introduction

Regularization is an important step in any optimization process to prevent overfitting by penalizing complex solutions. From a Bayesian point of view, regularization can be seen as a way to introduce prior distributions on model parameters. In the context of neural networks, the simplest regularization method is weight-decay, that controls the growth of parameters. Weight-decay adds an extra penalty term to the loss function of Eq. 6:

$$\ell\left(\theta, \mathcal{D}\right) = -\mathcal{L}\left(\theta, \mathcal{D}\right) + \lambda_2 \|W\|_2 \tag{7}$$

where $W \in \mathbb{R}^{d \times n}$ is the matrix of weights w_{ij} that connects visible and hidden units, λ_2 is the regularization coefficient, and d and n are the size of the visible and hidden layers, respectively. Weight decay penalizes large weights in W. There are some reasons for using weight-decay in an RBM: improve generalization to new data, make the receptive fields of the hidden units smoother and more interpretable by shrinking useless weights, etc [22]. However, the L_2 norm does not force zeros in the weights values. This causes the RBM model to be non-robust against noise. The problem is that although the weights of a standard RBM are spatially localized, they are not zero outside its influence area. So that, the overall contribution of all the connections with small weight values adversely affects the hidden activation units if there is any noise in the visible layer.

An alternative approach to reduce the effect of noise is to obtain a sparsely connected RBM, so that each hidden feature is connected to a few visible units. This goal can also be accomplished using regularization with a L_1 norm. In this case the loss function is given by:

$$\ell\left(\theta, \mathcal{D}\right) = -\mathcal{L}\left(\theta, \mathcal{D}\right) + \lambda_1 \|W\|_1 \tag{8}$$

where λ_1 is the regularization coefficient. This loss function often causes many of the weights to become exactly zero whilst allowing a few of them to grow quite large. The features obtained using the L_1 regularization are strongly localized which eases the interpretation. However, the L_1 norm is not very common in the literature. In our opinion, L_1 may remove too many connections and the remaining may contain large weights which is a known problem for generalization.

One of the most well-known approaches in order to combine both regularization terms is the Elastic-Net (EN) [23]. It is important to note that EN regularization was originally devoted for linear regression, a lasso convex problem, proposed by Zou and Hastie. The authors transform the naive elastic net problem into an equivalent lasso problem on augmented data. Moreover, the naive elastic net can perform an automatic variable selection in a fashion similar to the lasso. The EN approach applied to the RBM optimization function will lead to the following expression:

$$\ell\left(\theta, \mathcal{D}\right) = -\mathcal{L}\left(\theta, \mathcal{D}\right) + \lambda_1 \|W\|_1 + \lambda_2 \|W\|_2 \tag{9}$$

Unfortunately, all the advantages introduced by the authors for the lasso convex problem can not be extended to the RBMs due to the non-convexity and the nature of the optimization problem.

4.2. A Loss function combining L_2–L_1 regularization

As mentioned above, each of the norms L_1 and L_2 have its own advantages and disadvantages. It would be desirable to define a new regularization scheme that could combine the advantages of each norm into a single framework. To accomplish this, we propose to adaptively split the set of weights into two disjoints sets, so that L_1 regularization is used in one set and L_2 in the other. We define a new loss function given by:

$$\ell\left(\theta, \mathcal{D}\right) = -\mathcal{L}\left(\theta, \mathcal{D}\right) + \lambda_1 \left\|W \circ \hat{R}\right\|_1 + \lambda_2 \left\|W \circ R\right\|_2 \tag{10}$$

where R is a $d \times n$ binary mask and $\hat{R}$ is its complementary mask. Since W and R are multiplied point-wise, the *ones* in R represent the elements in W where the L_2 regularization is applied. On the other hand, the ones in $\hat{R}$ correspond to the elements in W where the L_1 regularization is used. Our loss function enforces many elements of W to be exactly zero (L_1 norm) and, at the same time, avoids weights to grow too large where the L_2 norm is applied. This type of regularization will be called Mask Selective Regularization (MSR) in the sequel.

A significant difference between EN and this approach, is that in MSR either L1 or L2 regularization is applied to each parameter, whereas in EN both regularizations are applied together to all the parameters. Typically, EN is unfruitful to combine L1 and L2 in cases where λ_1 and λ_2 have different magnitude orders. For instance, a large λ_1 dominates the minimization towards an sparse solution while a large λ_2 towards a shrinkage solution. However, this undesirable effect is avoided with the selective regularization of our approach because λ_1 and λ_2 values are applied separately over different weights.

4.3. Binary regularization mask

The first problem of this approach is how to build the binary matrix R. This matrix should be set according to the topology of the data, so that weak dependencies between visible units will be regularized using the L_1 norm and stronger dependencies using L_2. The regularization mask R is built by selecting a subset of binary vectors from a set $\mathcal{M}$. The set of a binary masks $\mathcal{M}$ is formed by d different vectors $\mathbf{m_i} \in R^d$, where each $\mathbf{m_i}$ is related to a different visible unit. Therefore, there are as many unique binary masks in the $\mathcal{M}$ set as number of visible units. Our goal is to obtain a mask set that sweeps the most influence areas of each dimension in the training set. We propose to use the Pearson's correlation coefficient [5, 6] as a surrogate of the mutual information to obtain these masks as mentioned before. Therefore, each binary element m_{ij} of the vector $\mathbf{m_i}$ would be 1 if the correlation between visible units i and j is high. The correlation coefficient is defined by:

$$corr\left(i,j\right) = \frac{cov\left(i,j\right)}{\sigma_i \sigma_j} = \frac{E\left[\left(i-\mu_i\right)\left(j-\mu_j\right)\right]}{\sigma_i \sigma_j} \tag{11}$$

where i and j are random variables that represent the visible units with expected values μ_i and μ_j and standard deviations σ_i and σ_j respectively.

To build the binary mask, we set to 1 the c components of $\mathbf{m_i}$ which respective highest correlation values and 0 for the rest. In order to decide the c parameter we propose to keep the correlation ratio above a certain threshold α:

$$\frac{\sum_{j \in S} \left|corr\left(i,j\right)\right|}{\sum_{j=1}^{d} \left|corr\left(i,j\right)\right|} > \alpha \tag{12}$$

where S is the set of indexes of the highest c correlation values.

In order to visually assess the effect of the parameter α, Figure 1 shows the effect of different values on a regularization mask associated to a visible unit. In the case of images, small values of α generates masks that are localized around the visible unit. Larger values of α capture more complex topologies and, in the case of images, can relate areas that are not spatially connected.

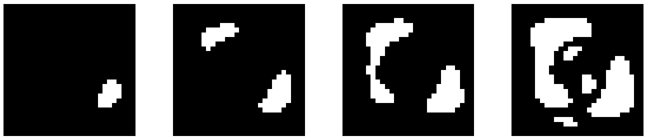


Figure 1: Binary masks for $\alpha \in \{0.2, 0.4, 0.6, 0.8\}$ for the MNIST dataset. All masks are associated to the same visible unit.

Figure 2 shows a few examples of regularization masks obtained from the MNIST training set using $\alpha = 0.7$. Each mask intends to capture the strongest dependencies for each visible unit where we would like to learn useful localized features detectors.

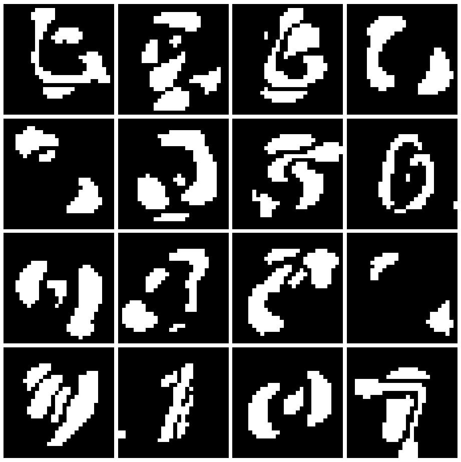


Figure 2: Few examples of binary regularization masks in $\mathcal{M}$ for the MNIST dataset.

As mentioned above, the regularization mask R is built by means of selecting a subset of binary vectors from the set of binary masks $\mathcal{M}$. It is important to note that this selection changes along the training process. Therefore the topology of the network, sparseness induced by R, changes adaptively selecting those masks that better explain the distribution of the input data, as it will be explained in Section 4.5. Moreover, it is important to mention that, unlike other methods that force sparse connections by hand assuming the image layout of the samples [3, 4], our method does not use any knowledge about the geometry of the data. In fact, we use our method to specify sparse connections not only in the first RBM of the network, but also in upper layers where you can not assume that the data follows any particular layout.

4.4. Topology selection and convergence

As mentioned before, the matrix R is a $d \times n$ matrix formed by binary vectors from the set $\mathcal{M}$. This matrix R is used to selectively apply a different regularization term over the weight matrix W. Let $\mathbf{w}_j$ and $\mathbf{r}_j$ be the j^{th} column of the matrix W and R, respectively. Each $\mathbf{w}_j$ maps the visible units to the j^{th} hidden feature. Following our MSR approach, the elements of $\mathbf{w}_j$ can be regularized using either a L_2 or L_1 norm according to the binary values of $\mathbf{r}_j$.

Instead of fixing the subset of binary vectors from $\mathcal{M}$ to be included in R, we propose to build R dynamically. In general, we could have the same binary mask $\mathbf{m_i}$ several times repeated in R, obviously it will happen when $n > d$. This effect is desirable because some local areas would require several hidden units in order to explain all the variability that appears in that portion of the representation space. Hence, each $\mathbf{r}_j$ is selected among a finite set of masks $\mathcal{M}$ and can be changed during the optimization process.

The selection of each $\mathbf{r}_j$ at each update time is done using an energy criterion that depends on the current weight values $\mathbf{w}_j$. The energy criterion is defined by:

$$\mathbf{r}_j = \max_{\mathbf{m}_k \in \mathcal{M}} E_{in}^k - E_{out}^k \tag{13}$$

$$E_{in}^k = \frac{\mathbf{m}_k^\intercal \mathbf{w}_j^2}{|\mathbf{m}_k|_1} \tag{14}$$

$$E_{out}^k = \frac{\hat{\mathbf{m}}_k^\intercal \mathbf{w}_j^2}{|\hat{\mathbf{m}}_k|_1} \tag{15}$$

where $\hat{\mathbf{m}}_k$ is the complementary of $\mathbf{m}_k$. The E_{in}^k value is the "positive energy" of the mask $\mathbf{m}_k$ with the feature $\mathbf{w}_j$ averaged by the mask's area. The intuitive idea is that given a feature vector $\mathbf{w}_j$, E_{in}^k will be high if the large weights of $\mathbf{w}_j$ are under the mask $\mathbf{m}_k$. Similarly, E_{out}^k value is the "negative energy". This term penalizes the case where large weights of $\mathbf{w}_j$ are not under $\mathbf{m}_k$. Therefore, after the selection, $\mathbf{r}_j$ is going to be the binary mask $\mathbf{m}_k \in \mathcal{M}$ that best covers the high values of $\mathbf{w}_j$. For these large weights the L_2 regularization will be applied. For the rest of weights, the L_1 regularization will be applied to enforce real zeros.

As we have said before, mask selection occurs in each epoch of the optimization process and the mask selected can change. In order to ensure the convergence of the selected mask by each feature, we propose to decrease linearly a probability assigned to the capacity of change the current mask. Hence the chance for changing the mask in very initial iterations is high, but tends to zero for the last iterations.

4.5. MSR Algorithm

The MSR algorithm entails to select the regularization matrices as described in previous section and compute the update equations derived from the loss function in Eq. 10. It is important to mention that the update of R is performed only once at every epoch, instead of at every batch. This is important because it allows the weights to settle to stable configurations and greatly reduces the computational cost. The binding between regularization masks and weights is quite changing at the beginning of the training process. At early epochs, since the values of the weights are randomly initialized, there is not a clear topology and the binary masks that best fit the weights varies significantly. As the process evolves, the correspondence freezes and each feature detector gets associated with a regularization mask. Figure 3 shows some features learned with the MSR agorithm for the MNIST dataset. The binary mask selected for each feature is overlayed in red.

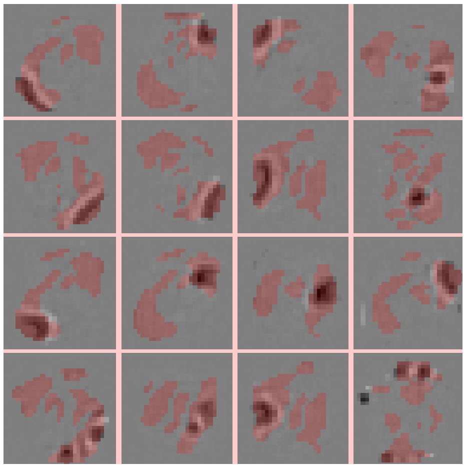


Figure 3: Learned features for the MNIST dataset along with their corresponding binary masks overlayed in red color.

Once the regularization masks are selected the derivative of the the loss function of Eq. 10 yields a very simple learning rule, following the CD-1 approach:

$$\begin{aligned}\Delta w_{ij} = \quad & \epsilon\big(\langle v_i h_j\rangle_{data} - \langle v_i h_j\rangle_{model} \\ & -\lambda_1 sgn\left(w_{ij}\cdot\hat{r}_{ij}\right) - 2\lambda_2\left(w_{ij}\cdot r_{ij}\right)\big)\end{aligned} \tag{16}$$

where w_{ij} are the weights of the RBM connections, and the angle brackets are used to denote expectations under the distribution specified by the subscript. Note that the last two terms are the derivatives of the L_1 and L_2 regularization terms, respectively, where $sgn(x)$ is the *signum* function of x.

One key point of our algorithm is that it is not made any assumption a priori about which regularization masks should be used. This selection is guided only by the topology of the data. In fact, we have observed that there exist masks in $\mathcal{M}$ that are never used while others can be reused several times in different feature detectors. It only depends on the complex dependencies of the visible data. Another interesting point is that our regularization serves as a guide to indicate where coefficients should be enforced to be zero. However, there exists situations where higher order relationships between visible units are not captured by the binary masks. In this case, the feature may have non-zero values in the L_1 regularized units outside the mask. The algorithm allows this phenomenon. This fact is illustrated in the bottom-right feature shown in Fig. 3, where large weights lay outside the mask (overlaid in red).

In order to show the differences among the different regularization techniques, Figure 4 shows the histograms of weight coefficients in the first layer on the MNIST dataset using different regularizations. The histogram without regularization is also included in the figure for completeness. It can be seen that the MSR inherits properties from both L_2 and L_1 regularizations. It forces many coefficients to be zero as in L_1 and the number of large coefficients is similar to L_2. Although the EN histogram in this case is very similar to the MSR, the results presented in next section will show the advantage of our MSR approach.

5. Experiments

In this section we present the evaluation carried out. We propose an evaluation over different datasets, images and non-images. First, two popular handwritten digit datasets: MNIST[1] and US Postal Service (USPS)[2] are considered. Second, in order to show the capabilities of the MSR to model the topology of the input space on non-images problems, we carried out experiments beyond image datasets. We run experiments with the well-known 20-Newsgroups text

[1] MNIST dataset is available here: http://yann.lecun.com/exdb/mnist/.

[2] USPS dataset is available in Matlab format here: http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html.

classification dataset[3]. Finally, we have used the CIFAR-10[4] to evaluate our algorithm in a more complex scenario that contains natural color images.

All of these experiments have a common framework based in [2]. Our goal is to train a DBN using a layer-by-layer pre-training method. Each layer of the network is trained as an RBM with the CD-1 algorithm in a completely unsupervised way. Finally, the entire network is fine-tuned discriminatively with back-propagation adding a "softmaxed" output layer and using the label information of the training samples. The fine-tuning process stops when the average cross-entropy error on the training data fall bellow a pre-specified threshold. To fix the threshold value, we fine-tune the network using only a subset of the training samples and using the remaining examples as a validation set. The cross-entropy threshold value is fixed with the fewest classification error on the validation set.

The results obtained with our approach (MSR model) will be compared against other regularization schemes: the No REG model is trained without any regularization penalty, whereas L_1 and L_2 are models trained including L_1 or L_2 regularization restrictions, respectively. The EN model uses both L_1 and L_2 at the same time. The regularization parameters λ_1 and λ_2, as well as the parameter α in our model, were chosen by model selection. For L_1 and L_2 methods, this model selection performs a search over λ_1 and λ_2 amog the values 0.0001, 0.001 and 0.01, fixing each value according to the best performance on a validation set in each case, respectively. For the EN method the grid search is performed jointly over λ_1 and λ_2. Finally, for our MSR method the grid search includes the α parameter in the range $0.3 - 0.8$ with a step size of 0.1. It is important to make it clear that the MSR algorithm is only applied during the pre-training phase, as well as the others regularization techniques. Once the models have been pre-trained, the discriminative fine-tuning procedure is the

[3] 20-Newsgroups dataset is available in Matlab format here: http://qwone.com/∼jason/20Newsgroups/.

[4] CIFAR-10 dataset is available here: http://www.cs.toronto.edu/∼kriz/cifar.html.

same in all cases. Also, it is important to highlight that the MSR approach is applied not only in the first layer but also in upper layers where we can not assume that the data has any particular geometry or layout. Actually, some extra experiments on the MNIST dataset have been performed to show the advantage of using MSR not only in the first layer, but also in upper layers of the DBN model. Note that the weights between the last layer and the output layer (that represents the labels) are not pre-trained.

In order to assess the robustness of the MSR approach we have run some extra experiments over a noisy version of the test set. To this end, for the digit datasets inspired by [4], the original test partition has been corrupted with three kinds of noise to reflect some possible sources of error. The first source of noise is a random noise where 10% of the pixels are activated. The second one introduces a border of two pixels wide to the images. Finally, the third one simulates a block occlusion by adding a square to the images in a random location. The area of that square is set to be one sixteenth of the total area of the image. We can see an example of each kind of noise in Figure 5. It is important to mention that the models are trained and validated in absence of noise in all cases. Similar experiments over a noisy test set are performed on the 20-Newsgroups text classification dataset where the noise is obtained modifying the word counts.

5.1. MNIST

The MNIST dataset has a training set of 60000 examples, and a test set of 10000 examples. The size of each image is 28×28 pixels and the pixel values were normalized to the range [0,1]. For these experiments we have used the same $784 - 500 - 500 - 2000 - 10$ deep network used by Hinton [2], which achieves 1.14% error on the test set. Each RBM has been trained for 50 epochs. Weights were initialized with small random values sampled from a normal distribution with zero mean and standard deviation of 0.1. The learning rate value was set to 0.1 for both weights and biases. The regularization terms λ_1 and λ_2 were fixed to 0.0001 for all the models, whereas α is fixed to 0.7 for our model. The

Table 1: Error rate (%) on the MNIST test set for different noise sources.

Model	Clean	Random	Border	Block
No reg	1.11	1.39	87.61	16.46
L_2	1.07	1.20	79.86	16.24
L_1	1.11	1.21	**2.82**	12.83
EN	1.16	1.24	3.37	13.17
MSR	**1.05**	**1.13**	2.89	**12.44**

results are given in Table 1. First, we observe that our model MSR achieves the best result in the noisy free case. In fact, we obtain a 1.05% error rate in the classical MNIST task, which is comparable to other results published for the permutation-invariant MNIST task. [2, 24, 4]. On the other hand, despite the fact that the No REG approach obtains reasonable good results on the clean version of the test set, the performance of this approach drops drastically when the noise appears in the test samples. However, MSR also outperforms in almost all cases for noisy test samples. Note that L_1 deals quite well with the Border case in this database where the digits are very clear. These results confirm the fact that regularization is a required procedure in order to ensure the generalization of the learned models.

Table 2: Effect of applying MSR to different layers of the network. Error rate (%) on the MNIST test set.

Layers			Clean	Random	Border	Block
1st	2nd	3rd				
No Reg	No Reg	No Reg	1.11	1.39	87.61	16.46
MSR	No Reg	No Reg	1.14	1.25	4.37	13.08
MSR	MSR	No Reg	1.09	1.19	5.34	12.73
MSR	MSR	MSR	**1.05**	**1.13**	**2.89**	**12.44**

We have also conducted an extra experiment on this dataset to show the advantage of using MSR in the upper layers of the network. These results are summarized in Table 2, where the error rate is shown for the same network of the previous experiment depending whether or not the MSR algorihm has been applied to each layer. According to these results the best performance is achieved when MSR is used in all layers. Note that the first and the last rows of the Table 2 correspond to the already results presented in Table 1. These results also demonstrate that MSR is useful when the topology of the data is not known as in the upper layers of the network.

5.2. USPS

The USPS dataset is composed by a training set of 7290 examples, and a test set of 2007 examples. The size of each image is 16×16 pixels and the pixel values were normalized to the range [0,1]. For these experiments we have used a $784 - 300 - 300 - 1200 - 10$ deep network as suggested in [25]. The results are given in Table 3. The training and testing procedure is the same as in the MNIST, except that in this case we have used $\lambda_1 = 0.001$ and $\alpha = 0.6$.

Table 3: Error rate (%) on the USPS test set for different noise sources.

Model	Clean	Random	Border	Block
No reg	5.17	33.45	71.08	28.36
L_2	5.10	28.26	62.78	28.55
L_1	5.24	27.18	62.63	26.75
EN	5.75	**20.82**	59.00	24.55
MSR	**5.03**	23.16	**55.64**	**24.25**

Again, some improvements are obtained on the clean version of the test dataset. Our 5% error rate obtained by MSR on clean images is comparable with other results obtained with convolutional networks [26]. Also, there is a clear advantage of the proposed technique when dealing with noise in most cases. It should be mentioned that since USPS images are smaller than MNIST images and their digits are scaled to fit the available area, the Border and Block noises affect the error rate more severely in the USPS case compared to the MNIST dataset (Table 1).

5.3. 20-Newsgroups

The 20-Newsgroups corpus is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. This corpus is processed using the *Rainbow* toolkit and selecting the 5,000 most informative words to define the vocabulary. Moreover, we have added noise to the text data (word counts) to evaluate also the behaviour of the different regularizations w.r.t noise. The network is formed by just one hidden layer with 1000 hidden units and a discriminative layer. The training and testing procedure is the same as in the MNIST with $\lambda_1 = \lambda_2 = 0.0001$ and $\alpha = 0.6$. Table 4 shows the results.

Table 4: Error rate (%) on the 20-Newsgroups test set for different noise sources.

Model	Clean	10%	20%	30%
No reg	**22.11**	24.14	25.56	28.09
L_2	22.23	23.85	25.17	**27.45**
L_1	22.35	24.32	26.38	28.94
EN	22.73	24.38	26.57	28.74
MSR	22.13	**23.73**	**25.14**	27.48

The results using our MSR are comparable with other results using similar models [27]. According to the results, MSR performs well on clean and noisy data in a non-image database as well, although the differences are not as significant as in the other tasks. Finally, it is worth to mention that in some other omitted results in this task, we found the inability of the EN to deal with values of λ_1 and λ_2 of different magnitude orders which worsens the results extremely. This effect does not happen using MSR, which gives reasonable results even in this extreme case.

5.4. CIFAR-10

The CIFAR-10 dataset consists of 60000 32×32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. They are normalized to be zero-mean and unit-variance. The network is formed by just one hidden layer with 4000 hidden units. To deal with real-valued data such as the pixel intensities in natural images we have replaced the visible binary units by gaussian units obtaining a Gaussian-RBM [2]. The GRBM model can be trained with the CD-1 algorithm as well. Weights were initialized with small random values sampled from a normal distribution with zero mean and standard deviation of 0.05. The learning rate value was set to 0.001 for both weights and biases. The regularization terms λ_1 and λ_2 were set to 0.001 for all the models, whereas α is fixed to 0.3 for our model. Unlike the other datasets, the discriminative step, once the GRBM is trained, has been done feeding the hidden output activations of each sample into a linear SVM. This methodology has been previously used in the literature [10, 28]. Table 5 shows the results for the different regularizations and the proposed MSR method. Note that we have compared the results only on clean images, since the robustness of MSR have been already demonstrated in the previous datasets.

Table 5: Accuracy (%) on the CIFAR-10 test set.

Model	Accuracy
L_2	51.14
L_1	52.98
EN	52.44
MSR	**58.64**

Despite these results are far from the state of the art [29, 30], it is important to mention that our algorithm does not make any assumption on the geometry of the data. The best results on CIFAR-10 are obtained using convolutional models that exploit the strong spatially local correlation present in natural images. In fact, in [31] the authors show the difficulty of learn interesting-looking filters on natural images without a convolutional approach. The best result in that work is 64.84% of accuracy on the test set, but using a hidden layer with 10000 units and pre-training the RBM with a much bigger subset from the Tiny Images dataset. However, in the present work, our model uses a conventional RBM (non-convolutional) and is trained with the standard training set. Under this restrictions, our method outperforms other results obtained with similar non-convolutional models on CIFAR-10. For instance, in [32] a three layer stacked autoencoder is trained on CIFAR-10 obtaining 53.2% of accuracy far from our 58.64% using MSR.

6. Conclusions

In this paper, we have described a new algorithm called Mask Selective Regularization (MSR) to improve the RBM's learning capabilities based on two important issues. On the one hand, we propose a new method to take advantages of the topology of the input space. On the other hand, we presented a new scheme that uses a selective L_2–L_1 regularization criterion to ensure sparseness of the representation and generalization capabilities. In order to assess the new proposed method we have compared the performance of MSR with other regularization schemes using both noisy and original versions of image and non-image datasets. MSR outperforms the other models in all cases in presence of noise, being still a good discriminative classifier comparable with the state of the art on the original sets using similar methods. Also, a key aspect of our approach is that it can be applied without any knowledge about the geometry of the data, and may be applied in upper layers of in a deep network.

For future work, we would like to investigate the applicability of the MSR algorithm to other types of deep networks, like Deep Boltzmann Machines, Deep Autoencoders, etc. Also, although the applicability of MSR in upper layers of the network has been done, it would be interesting to evaluate the performance of our approach with other non-image datasets.

Acknowledgment

This work was financially supported by the Ministerio de Ciencia e Innovación (Spain), Plan Nacional de I+D+i, TEC2009-09146, and the FPI grant BES-2010-032945.

References

[1] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, Pattern Analysis and Machine Intelligence, IEEE Transactions on 35 (8) (2013) 1798–1828.

[2] G. E. Hinton, R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[3] A. Mller, H. Schulz, S. Behnke, Topological features in locally connected rbms, in: IJCNN, IEEE, 2010, pp. 1–6.

[4] Y. Tang, C. Eliasmith, Deep networks for robust visual recognition, in: J. Frnkranz, T. Joachims (Eds.), Proceedings of the 27th International Conference on Machine Learning, June 21-24, 2010, Haifa, Israel, Omnipress, 2010, pp. 1055–1062.

[5] W. Li, Mutual information functions versus correlation functions, Journal of Statistical Physics 60 (1990) 823–837.

[6]

[7] M. Thom, P. Gnther, Sparse activity and sparse connectivity in supervised learning, J. Mach. Learn. Res. 14 (1) (2013) 1091–1143.

[8] H. Lee, C. Ekanadham, A. Y. Ng, Sparse deep belief net model for visual area V2, in: J. C. Platt, D. Koller, Y. Singer, S. Roweis (Eds.), Advances in Neural Information Processing Systems 20, MIT Press, Cambridge, MA, 2008, pp. 873–880.

[9] P. Vincent, H. Larochelle, Y. Bengio, P. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th international conference on Machine learning, ACM, New York, NY, USA, 2008, pp. 1096–1103.

[10] Y. Tang, R. Salakhutdinov, G. E. Hinton, Robust boltzmann machines for recognition and denoising, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, Washington, DC, USA, 2012, pp. 2264–2271.

[11] A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, A. Y. Ng, Recurrent neural networks for noise reduction in robust asr, in: Proceedings of INTERSPEECH, ISCA, 2012.

[12] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408.

[13] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, CoRR abs/1207.0580.

[14] V. Nair, G. E. Hinton, 3d object recognition with deep belief nets, in: Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, A. Culotta (Eds.), NIPS, Curran Associates, Inc., 2009, pp. 1339–1347.

[15] K. Gregor, A. Szlam, Y. LeCun, Structured sparse coding via lateral inhibition, in: J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems (NIPS 2011), 2011, pp. 1116–1124.

[16] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: JMLR W&CP: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011), 2011, pp. 315–323.

[17] Y. Bengio, Deep learning of representations: Looking forward, in: Statistical Language and Speech Processing, Vol. 7978 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 1–37.

[18] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[19] H. Schulz, A. Mller, S. Behnke, Exploiting local structure in boltzmann machines, Neurocomputing 74 (9) (2011) 1411–1417.

[20] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems 25, 2012, pp. 1106–1114.

[21] G. E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. 14 (8) (2002) 1771–1800.

[22] G. E. Hinton, A practical guide to training restricted boltzmann machines, Tech. rep. (2010).

[23] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, Journal of the Royal Statistical Society, Series B 67 (2005) 301–320.

[24] R. Salakhutdinov, G. E. Hinton, Deep boltzmann machines, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2009, pp. 448–455.

[25] G. E. Hinton, S. Osindero, Y. W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.

[26] Y. LeCun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, D. Henderson, in: D. S. Touretzky (Ed.), Advances in neural information processing systems 2, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990, Ch. Handwritten digit recognition with a back-propagation network, pp. 396–404.

[27] H. Larochelle, Y. Bengio, Classification using discriminative restricted boltzmann machines, in: In ICML 08: Proceedings of the 25th international conference on Machine learning. ACM, 2008.

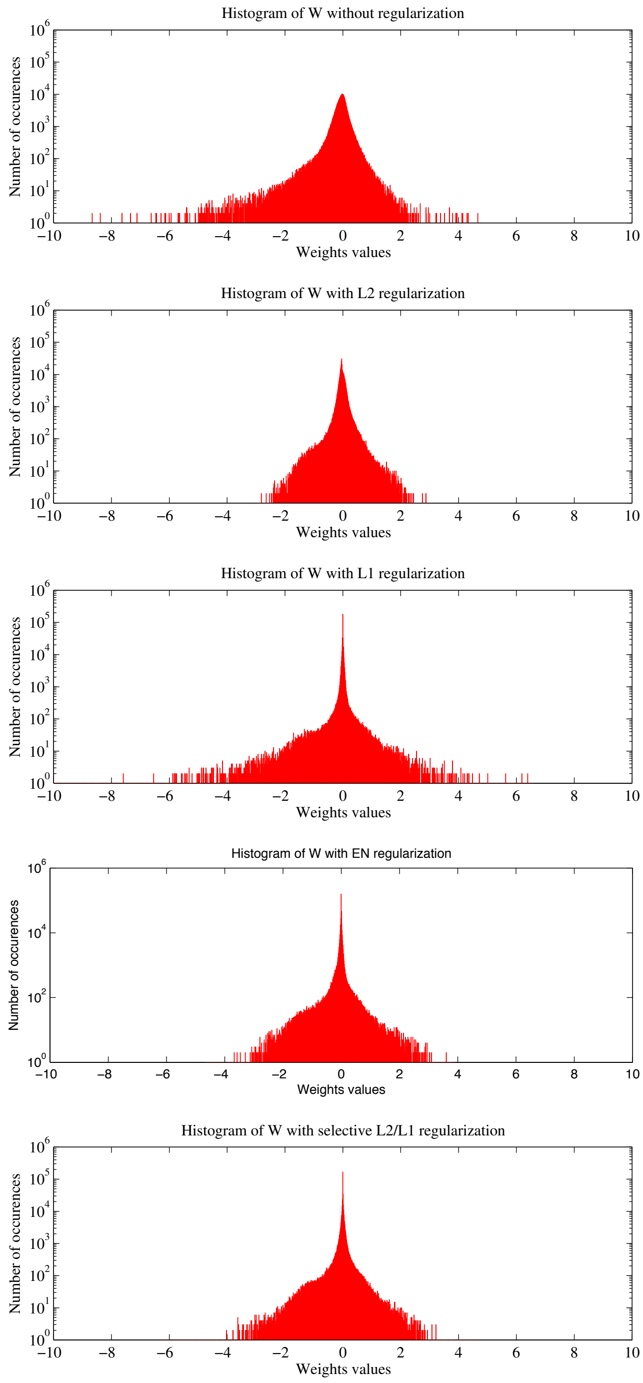
[28] A. Coates, H. Lee, A. Y. Ng, An analysis of single-layer networks in unsupervised feature learning, in: AISTATS, 2011.

[29] L. Wan, M. D. Zeiler, S. Zhang, Y. LeCun, R. Fergus, Regularization of neural networks using dropconnect, Vol. 28 of JMLR Proceedings, JMLR.org, 2013.

[30] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, Y. Bengio, Maxout networks, in: ICML (3), 2013, pp. 1319–1327.

[31] A. Krizhevsky, Learning multiple layers of features from tiny images, Tech. rep. (2009).

[32] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks., JMLR.org, 2011.


Histogram of W without regularization

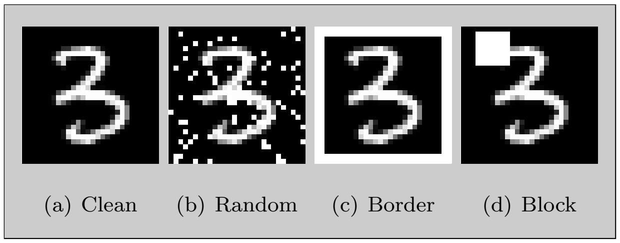
Histogram of W with L2 regularization

Histogram of W with L1 regularization

Histogram of W with EN regularization

Histogram of W with selective L2/L1 regularization

Figure 4: Histogram of weights for different regularization schemes.



(a) Clean (b) Random (c) Border (d) Block

Figure 5: Clean and noise images