General Sparse Penalty in Deep Belief Networks: Towards Domain Adaptation*

Xanadu C. Halkias, Sébastien Paris and Hervé Glotin

Abstract

Deep Belief Networks (DBN) have been successfully applied on popular machine learning tasks. Specifically, when applied on hand-written digit recognition, DBNs have achieved approximate accuracy rates of 98.8%. In an effort to optimize the data representation achieved by the DBN and maximize their descriptive power, recent advances have focused on inducing sparse constraints at each layer of the DBN. In this paper we present a novel generalized theoretical approach for sparse constraints in the DBN using the sparse mixed norm. We explore how these constraints affect the classification accuracy for digit recognition in three different datasets and provide initial estimations for domain adaptation applications through cross-training and testing of the networks.

1 Introduction

Restricted Boltzmann Machines (RBMs) are Energy Based Models (EBMs) that have been extensively used for a diverse set of machine learning applications mainly due to their generative and unsupervised learning framework. These applications range from image scene recognition and generation Hinton et al. (2006), video-sequence recognition Hinton (2007) and dimensionality reduction Hinton & Salakhutdinov (2006).

An equally important aspect of RBMs is that they serve as the building blocks of DBNs Hinton & Salakhutdinov (2006)Bengio et al. (2007). Their use as such has been favored in the machine learning community due to the conditional independence between the hidden units in the RBM that allows for the efficient and computationally tractable implementation of deep architectures.

In recent years, sparsity has become an important requirement in both shallow and deep architectures. Although primarily used in statistics for optimization tasks in order to overcome the curse of dimensionality in many applications, it also serves as a way to emulate biological plausible models of the human visual

*LSIS.RR.2012.DY.01 - This work was accomplished under the support of the project Cognilego ANR-2010-CORD-013 financed by the Agence Nationale de la Reserche (FRANCE) program CONTINT Science and Technology Committee of Information and Communication.

cortex where it has been shown that sparsity is an integral process in the hierarchical processing of visual information Lee et al. (2008); Olshausen & Field (1997).

Moreover, an added benefit of using sparse constraints in the form of mixed norm regularizers in deep architectures is that they can alleviate their restrictive nature by allowing implicit interactions between the hidden units in the RBMs. Mixed norm regularizers such as $l_{1,2}$ have been extensively used in statistics and machine learning Sra et al. (2011). In this paper we provide a generalized approach for inducing sparse constraints by using the sparse mixed norm regularizer introduced by Friedman, Hastie and Tibshirani Friedman et al. (2010) on the activation probabilities of the RBMs. We also show that this regularizer can be used to train DBNs that offer an advantage in digit recognition in several datasets, but also provide insights for robust domain adaptation applications.

2 Restricted Boltzmann Machines

An RBM is a type of two layer neural network comprised of a visible layer that represents the observed data x and a hidden layer that represents the hidden variables h. The addition of these hidden units allows the model an increased capacity in expressing the underlying distribution of the observed data.

RBMs are energy based models and as such they define a probability distribution through an energy function as seen in Eq. 1

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

Where Z, provided in Eq. 2, is called the partition function and is a normalizing factor ensuring that Eq. 1 is a probability.

$$Z = \sum_{\mathbf{x}, \mathbf{h}} e^{-E(\mathbf{x}, \mathbf{h})} \tag{2}$$

In the case of an RBM the energy function $E(\mathbf{x}, \mathbf{h})$ is defined in Eq. 3:

$$E_\theta(\mathbf{x}, \mathbf{h}) = -\sum_{i=1}^{I} \sum_{j=1}^{J} x_i h_j w_{ij} - \sum_{i=1}^{I} b_i x_i - \sum_{j=1}^{J} a_j h_j, \tag{3}$$

where $\boldsymbol{\theta} \triangleq \{\mathbf{W}, \mathbf{b}, \boldsymbol{a}\}$ are the parameters of the network and $h_j \in \{0, 1\}$. $\mathbf{W}$ is the weight matrix, $\mathbf{b}$ are the visible unit biases and $\boldsymbol{a}$ are the hidden unit biases.

In the common case where we are using stochastic binary units for both visible and hidden units, then the conditional probabilities of activation are obtained by:

$$\begin{aligned} p(x_i = 1 | \mathbf{h}) &= \sigma(b_i + \textstyle\sum_j h_j w_{ij}) \\ p(h_j = 1 | \mathbf{x}) &= \sigma(a_j + \textstyle\sum_i x_i w_{ij}), \end{aligned} \tag{4}$$

where σ is the sigmoid function and

$$\sigma(f(x)) \triangleq \frac{1}{1+e^{-f(x)}}. \tag{5}$$

Since an RBM does not allow for connections amongst hidden units or amongst visible units we can easily obtain Eq. 6.

$$\begin{aligned} p(\mathbf{x}|\mathbf{h}) &= \prod_i p(x_i|\mathbf{h}) \\ p(\mathbf{h}|\mathbf{x}) &= \prod_j p(h_j|\mathbf{x}) \end{aligned} \tag{6}$$

Intuitively, the observed data, $\mathbf{x}$ will be modeled by those hidden units, $\mathbf{h}$ that are expressed with a high conditional probability $p(h_j|\mathbf{x})$. The goal of adding sparse constraints to the network is to allow for the salient activation of the hidden units based on the differences of the observed data. As a result, we can achieve an initial clustering of the observed data that will increase the discriminative power of the model.

2.1 Training an RBM

RBMs are energy based, generative models that are trained to model the marginal probability $p(\boldsymbol{x})$ of the observed data where:

$$p(\mathbf{x}) = \sum_{\mathbf{h}\in\{0,1\}^J} p(\mathbf{x},\mathbf{h}). \tag{7}$$

In general, energy based models can be learnt by performing gradient descent on the negative log-likelihood of the observed data. Specifically, to learn the parameters of the network we need to compute the gradient provided in Eq. 8 given the observed (training) data $\boldsymbol{x^l}$.

$$-\frac{\partial logp(\mathbf{x})}{\partial\boldsymbol{\theta}} = \langle\frac{\partial E_\theta(\boldsymbol{x^l},\mathbf{h})}{\partial\boldsymbol{\theta}}\rangle_{\boldsymbol{h}} - \langle\frac{\partial E_\theta(\mathbf{x},\mathbf{h})}{\partial\boldsymbol{\theta}}\rangle_{\boldsymbol{x},\boldsymbol{h}}, \tag{8}$$

where $\langle\cdot\rangle_n$ denotes the expectation with respect to n. As evident in Eq.8, the gradient has two phases. The positive phase which tries to lower the energy of the training data $\boldsymbol{x^l}$ and the negative phase which tries to increase the energy of all x in the model.
Assessing the energy on all the data can be an intractable task given the size of the network and the number of possible configurations. In order to obtain an approximation Hinton Hinton (2000); Hinton & Salakhutdinov (2006) successfully proposed the use of Contrastive Divergence (CD). This allows us to sample an approximation of the expectation over $(\mathbf{x},\mathbf{h})$ using Gibbs sampling at only k steps. Empirically, it has been shown that setting $k=1$ will provide an adequate approximation although it will not follow the theoretical gradient Bengio et al. (2007).

Applying CD on Eq. 8 we can obtain the following update equations for the parameters of the network.

$$\Delta \boldsymbol{w}_{\cdot j} = \tfrac{1}{L} \sum_{l=1}^{L} \boldsymbol{x}^{\boldsymbol{l}} p(h_j = 1|\boldsymbol{x}^{\boldsymbol{l}}) - \widetilde{\boldsymbol{x}}^{\boldsymbol{l}} p(h_j = 1|\widetilde{\boldsymbol{x}}^{\boldsymbol{l}}) \tag{9}$$

$$\Delta b_i = \tfrac{1}{L} \sum_{l=1}^{L} p(x_i^l = 1|\boldsymbol{h}) - p(\widetilde{x}_i^l = 1|\boldsymbol{h}) \tag{10}$$

$$\Delta a_j = \tfrac{1}{L} \sum_{l=1}^{L} p(h_j = 1|\boldsymbol{x}^{\boldsymbol{l}}) - p(h_j = 1|\widetilde{\boldsymbol{x}}^{\boldsymbol{l}}), \tag{11}$$

where the $\widetilde{(\cdot)}$ defines the generated distributions obtained by the CD.

In the next section, we introduce a general version of sparse constraints in the learning phase of the RBM through the use of the sparse mixed norm in an effort to control the activation probabilities of the hidden units.

3 Sparse Mixed Norm RBMs

Several attempts in inducing sparse constraints in the RBM by Lee et al. (2008); Ranzato et al. (2008) have been successful in increasing the discriminative power of the models. Examples of these sparse constraints range from weight decay Hinton (2010) to modified norm penalties Ranzato et al. (2007). In this paper we offer a generalized penalty based on the work by Friedman et al Friedman et al. (2010) that blends the sparse, l_1 norm with the mixed norm ($l_{1,2}$). We will refer to this generalized penalty applied to the expectations of the activation probabilities as the Sparse Mixed Norm RBM (SMNRBM).

As mentioned before, learning an RBM consists of performing gradient descent on the negative log-likelihood. We can thus define the cost function L to be minimized as $L = -logp(\boldsymbol{x})$. When applying the sparse mixed norm regularizer the cost function takes the general form of Eq. 12.

$$\begin{aligned} &L = -logp(\mathbf{x}) \\ &+\lambda_1(\|p(\mathbf{h} = \mathbb{1}|\mathbf{x})\|_{1,2}) + \lambda_2(\|p(\mathbf{h} = \mathbb{1}|\mathbf{x})\|_1) \end{aligned} \tag{12}$$

Where λ_1 and λ_2 are regularizer constants. The second term of Eq. 12 defines the mixed norm penalty on the expectations of the hidden unit activation probabilities. In order to apply the mixed norm we assume that the hidden units are equally divided into non-overlapping groups. As a result, we are able to penalize a whole group and not just individual hidden units.

Given an RBM with J hidden units we define a partition of the hidden units into groups P_m where $m = 1, 2, ...M$ defines the number of groups. The groups are non-overlapping and of equal size to alleviate computational issues. The mixed norm penalty for a data sample $\boldsymbol{x}^{\boldsymbol{l}}$ is defined in Eq. 13.

$$\begin{aligned} \|p(\mathbf{h} = \mathbb{1}|\boldsymbol{x}^{\boldsymbol{l}})\|_{1,2} &= \sum_{m=1}^{M} \|p(\boldsymbol{h}_{Pm}|\boldsymbol{x}^{\boldsymbol{l}})\|_2 \\ &= \sum_{m=1}^{M} \sqrt{\sum_{k \in P_m} p(h_k = 1|\boldsymbol{x}^{\boldsymbol{l}})^2} \end{aligned} \tag{13}$$

In practice, the desire behind the application of the mixed norm penalty is to set groups of the hidden units to zero when representing the observed data by forcing their activation probabilities to zero. As a result, given an observed data sample only a small number of groups of hidden units will be activated, leading to its sparse representation.

The third term of Eq. 12 defines the sparse l_1 penalty on the totality of the hidden units' activation probabilities. This can be further seen in Eq. 14.

$$\|p(\mathbf{h} = \mathbb{1}|\boldsymbol{x}^{\boldsymbol{l}})\|_1 = \sum_{j=1}^{J} |p(h_i = 1|\boldsymbol{x}^{\boldsymbol{l}})| \tag{14}$$

The addition of this penalty aims at inducing sparsity at the individual level of the hidden units by forcing single activation probabilities to zero.

Overall, blending the two penalties will result in sparse representations of the observed data both at a low resolution, group level, as well as in a high resolution, individual level, of the activation probabilities of the hidden units.

3.1 Training the Sparse Mixed Norm RBM

In order to train the SMNRBM and obtain the model parameters $\boldsymbol{\theta}$ we need to minimize the cost function presented in Eq. 12. This can be achieved by performing a coordinate descent once we have obtained the gradients of the regularizers.

The gradient of the mixed norm penalty for the weights, W is as follows:

$$\begin{aligned}
&\tfrac{\partial}{\partial \boldsymbol{w}_{\cdot j}}(\|p(\mathbf{h} = \mathbb{1}|\boldsymbol{x}^{\boldsymbol{l}})\|_{1,2}) = \\
&= \tfrac{1}{2} \cdot \tfrac{1}{\sqrt{\sum\limits_{k \in P_m} p(h_k=1|\boldsymbol{x}^{\boldsymbol{l}})^2}} \cdot 2 \cdot p(h_k = 1|\boldsymbol{x}^{\boldsymbol{l}}) \cdot \tfrac{\partial p(h_k=1|\boldsymbol{x}^{\boldsymbol{l}})}{\partial \boldsymbol{w}_{\cdot j}} \\
&= \tfrac{p(h_k=1|\boldsymbol{x}^{\boldsymbol{l}})}{\|p(\boldsymbol{h}_m|\boldsymbol{x}^{\boldsymbol{l}})\|_2} \cdot \tfrac{\partial p(\boldsymbol{h}_k=1|\boldsymbol{x}^{\boldsymbol{l}})}{\partial w_{\cdot j}} \\
&= \tfrac{p(h_k=1|\boldsymbol{x}^{\boldsymbol{l}})}{\|p(\boldsymbol{h}_m|\boldsymbol{x}^{\boldsymbol{l}})\|_2} \cdot p(h_k = 1|\boldsymbol{x}^{\boldsymbol{l}})[1 - p(h_k = 1|\boldsymbol{x}^{\boldsymbol{l}})] \cdot \boldsymbol{x}^{\boldsymbol{l}} \\
&= \tfrac{p(h_k=1|\boldsymbol{x}^{\boldsymbol{l}})^2}{\|p(\boldsymbol{h}_m|\boldsymbol{x}^{\boldsymbol{l}})\|_2} \cdot p(h_k = 0|\boldsymbol{x}^{\boldsymbol{l}}) \cdot \boldsymbol{x}^{\boldsymbol{l}}.
\end{aligned} \tag{15}$$

When applied on the expectations of the activation probabilities the mixed norm penalty will follow their trend while forcing the groups that include members with low activation probabilities towards zero. The l_2 norm in the denominator ensures that the groups with low activations will be pushed further closer to zero.

Similarly, the gradient for the sparse l_1 penalty for the weights is as follows:

$$\begin{aligned}
&\tfrac{\partial}{\partial \boldsymbol{w}_{\cdot j}}(\|p(\mathbf{h} = \mathbb{1}|x^l)\|_1) = \\
&= p(h_k = 1|\boldsymbol{x}^{\boldsymbol{l}})[1 - p(h_k = 1|\boldsymbol{x}^{\boldsymbol{l}})] \cdot \boldsymbol{x}^{\boldsymbol{l}} \\
&= p(h_k = 1|\boldsymbol{x}^{\boldsymbol{l}})p(h_k = 0|\boldsymbol{x}^{\boldsymbol{l}}) \cdot \boldsymbol{x}^{\boldsymbol{l}}
\end{aligned} \tag{16}$$

Algorithm 1 Sparse Mixed Norm RBM learning algorithm

```
1. Update the parameters $\boldsymbol{\theta}$ using CD and Eq. 9- 11
2. Update the parameters again using the gradient of the regularizations as in Eq. 17- 18
3. Repeat steps 1, 2 until convergence
```

In this case, the expectations of the activations of the hidden units are individually penalized and low activations are forced to zero. Given the gradients of the penalties the update equations for the SMNRBM are presented bellow:

$$\begin{aligned}&\Delta \boldsymbol{w}_{\cdot j} = \\ &\tfrac{1}{L}\sum_{l=1}^{L}[(p(h_j = 1|\boldsymbol{x^l}) + \lambda_1 \tfrac{p(h_j=1|\boldsymbol{x^l})p(h_j=0|\boldsymbol{x^l})}{\sqrt{\sum p(h_m=1|\boldsymbol{x^l})^2}} + \\ &\lambda_2 p(h_j = 1|\boldsymbol{x^l})p(h_j = 0|\boldsymbol{x^l})) \cdot \boldsymbol{x^l} - p(h_j = 1|\widetilde{\boldsymbol{x}}^{\boldsymbol{l}})\widetilde{\boldsymbol{x}}^{\boldsymbol{l}}]\end{aligned} \tag{17}$$

$$\begin{aligned}&\Delta a_j = \\ &\tfrac{1}{L}\sum_{l=1}^{L}[(p(h_j = 1|\boldsymbol{x^l}) + \lambda_1 \tfrac{p(h_j=1|\boldsymbol{x^l})p(h_j=0|\boldsymbol{x^l})}{\sqrt{\sum p(h_m=1|\boldsymbol{x^l})^2}} + \\ &\lambda_2 p(h_j = 1|\boldsymbol{x^l})p(h_j = 0|\boldsymbol{x^l}) - p(h_j = 1|\widetilde{\boldsymbol{x}}^{\boldsymbol{l}})]\end{aligned} \tag{18}$$

The detailed steps for training the SMNRBM are depicted in Algorithm 1.

Mixed Norm RBM: The general penalty of Eq. 12 allows us through the manipulation of the constant regularizers, λ_1 and λ_2 to obtain different types of architectures. In the case where $\lambda_2 = 0$ we have the mixed norm RBM as described in the work of Luo et al. (2011). In this case the sparsity is induced at the group level of the hidden units whereby the observed data is represented by a small number of groups of hidden units. The λ_1 constant is empirically set based on the task at hand.

Fig. 1 shows sample weights for the mixed norm RBM when using $\lambda_1 = 0.1$. Fig 2 provides the average probability activations for the hidden units given the training data. As seen in the figure, the activation probabilities of the hidden units appear to be more towards the left-hand side of the figure which is the desired effect. However, there appears to be a bimodality whereby a large proportion of the activation probabilities is set to a high value. This may be attributed to the choice and size of grooups when applying the mixed norm penalty. Given that the activation probabilities are pushed towards high values one can expect that such a process may have an adverse result for classification tasks since the hidden units will over-represent the observed data.

Sparse l_1 RBM: Setting $\lambda_1 = 0$ in Eq. 12 we obtain the sparse l_1 RBM. In this case sparsity is induced by setting individual hidden units to zero. The λ_2 constant is, as before, empirically chosen based on the task at hand. Fig. 3 shows sample weights for the mixed norm RBM when using $\lambda_2 = 0.01$. Fig 4 provides the average of the probability activations for the hidden units given a single batch (100 samples) of the training data. In this case, the induced sparsity is more obvious when using the sparse l_1 compared to that of the vanilla RBM.

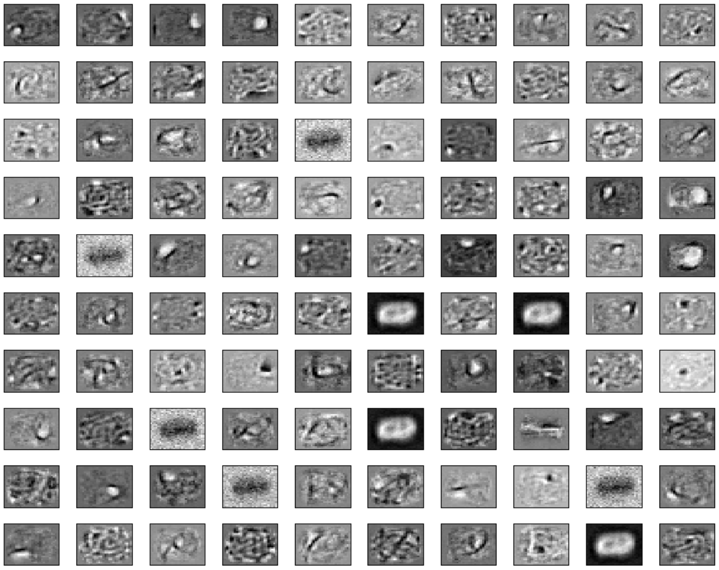

Figure 1: Sample learned weights W for the mixed norm RBM using the MNIST data set

Sparse Mixed Norm RBM: Setting both constant regularizers to non-zero values we obtain the generalized sparse mixed norm RBM. Empirically obtained values for $\lambda_1 = 0.1$ and $\lambda_2 = 0.01$ are used for the task at hand. Fig. 5 shows sample weights for the sparse mixed norm RBM. Fig 6 provides an example of the average activation probabilities for the hidden units given a batch of the MNIST training data. In this case we see the combination of the previous architectures whereby the mixed norm penalty tends to saturate the activation probabilities, while the sparse penalty forces activations to zero.

3.2 Pre-training DBNs with SMNBMs

RBMs became increasingly popular when Hinton and Salakhudinov Hinton & Salakhutdinov (2006) Hinton et al. (2006) used them as building blocks for creating and pre-training efficient DBNs. The proposed SMNRBMs can be utilized in the same manner to initialize DBNs and obtain a sparse and computationally efficient representation of the observed data.

In order to offer a comparative view between the different architectures we used Hinton's model for digit recognition, but we substituted the vanilla RBM

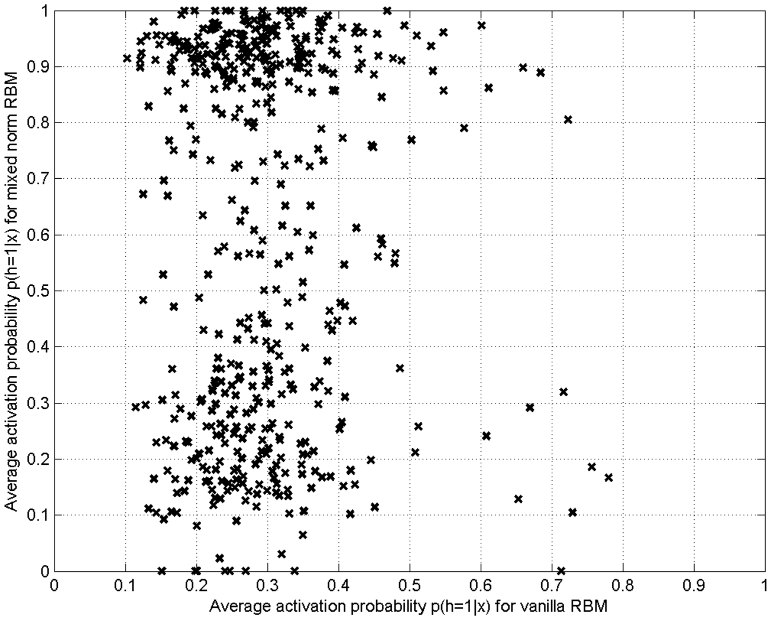

Figure 2: Average of hidden unit activation probabilities for the mixed norm RBM using a batch of the MNIST data set. Y-axis: Hidden unit activation probabilities for mixed norm RBM. X-axis: Hidden unit activations for vanilla RBM

with the proposed SMNRBM. We pre-trained a $500 - 500 - 2000$ DBN and tested it on three different data sets, MNIST, RIMES and UPS.

Continuing, to obtain classification error rates we added 10 softmax layers to get the posterior probabilities for the different classes. The network was fine-tuned using conjugate gradient as described in Hinton & Salakhutdinov (2006). The constant regularizers were empirically set to $\lambda_1 = 0.1$ and $\lambda_2 = 0.01$ and for the mixed norm architecture we used different group sizes for the hidden units, 5, 20 and 100 respectively. For the SMNDBN architecture we used $\lambda_1 = 0.1$ and $\lambda_2 = 0.01$ and a group size of 5. Results on the classification accuracy and the computational cost of the models can be seen in Table 1 and Table 2 respectively. All experiments were performed on a 24 core server (AMD Opteron processor 8435) with a core CPU of 2593.831MHz and a cache of 512KB.

From Table 1 we can infer that our proposed general penalty can offer the flexibility of creating architectures that will be able to either match or marginally increase the classification accuracy of the models depending on the underlying distributions. It appears that for the task of hand-written digit recognition the

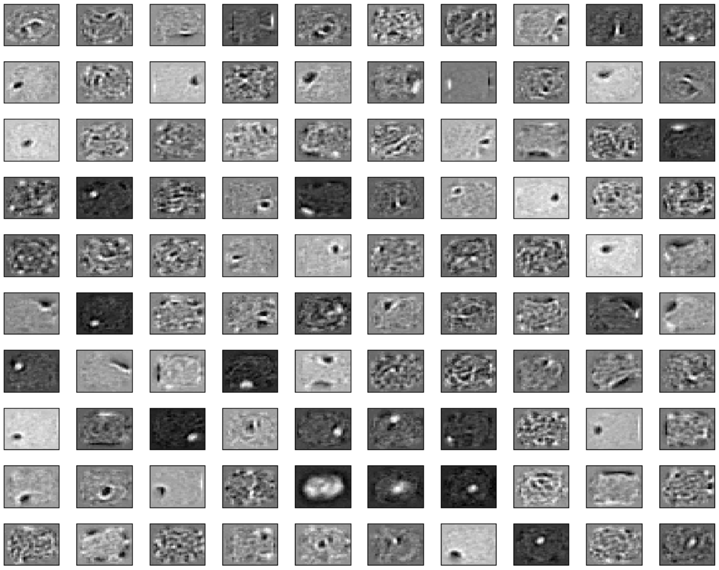

Figure 3: Sample learned weights W for the sparse l_1 norm RBM using the MNIST data set

distribution of the observed data favors the use of larger group sizes for the mixed norm architectures.

In order to get a better understanding of the impact of the different sparse constraints and architectures, Figure 7 depicts the average probability density functions of the expectations of the activation probabilities for the MNIST training data.

It is interesting to note that the proposed architectures that utilize the mixed norm penalty (MNDBN, SMNDBN) tend to aggressively push their activation probabilities to zero, which can be perceived as a direct result of the use of the l_1 norm on the groups of the hidden units. However, these architectures also tend to increasingly activate their hidden units with a high probability, which can be perceived as a reverse effect since we will have similar groups of hidden units representing different classes of the data. A possible way for alleviating this phenomenon may be to constrain the penalty of the expectations as seen in Ranzato et al. (2008).

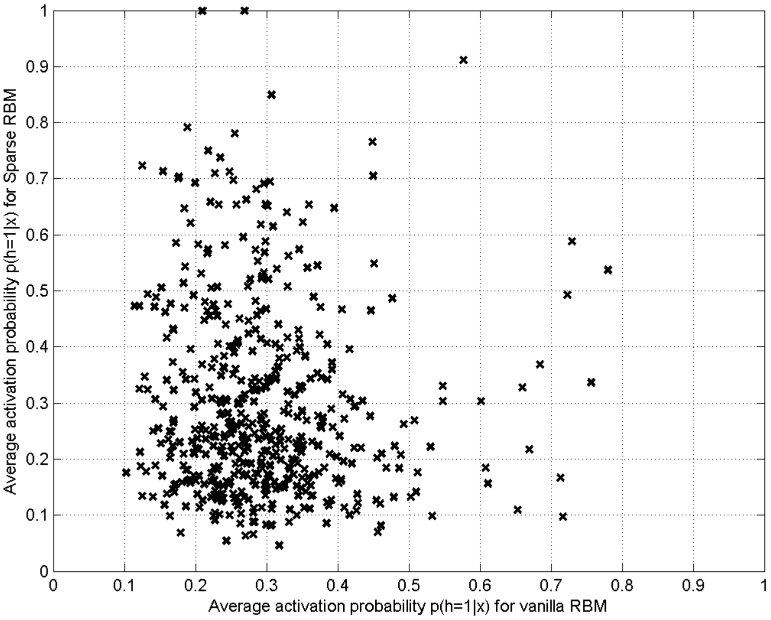

Figure 4: Average of the hidden unit activation probabilities for the sparse l_1 norm RBM using a batch of the MNIST data set. Y-axis: Hidden unit activation probabilities for mixed norm RBM. X-axis: Hidden unit activations for vanilla RBM

4 DBNs and domain adaptation

Domain adaptation refers to the ability of a model to generalize its performance when the data classes remain the same, but the observed data (domain) changes. Existing literature on hand-written digit recognition is limited in the use of specific data sets such as MNIST and does not offer enough insight on how deep architectures adapt to different domain settings. In order to explore this proposition we employ a $784-500-500-2000-10$ DBN, pre-trained using the proposed SMNRBMs on three different data sets for training and cross-testing.

4.1 Data

We have used three different data sets in order to train and test the network.

- MNIST is a popular data set in the community for hand-written digit recognition and is comprised of 70000, 28×28 images (60000 train - 10000 test). It is publicly available at `yann.lecun.com/exdb/mnist`.

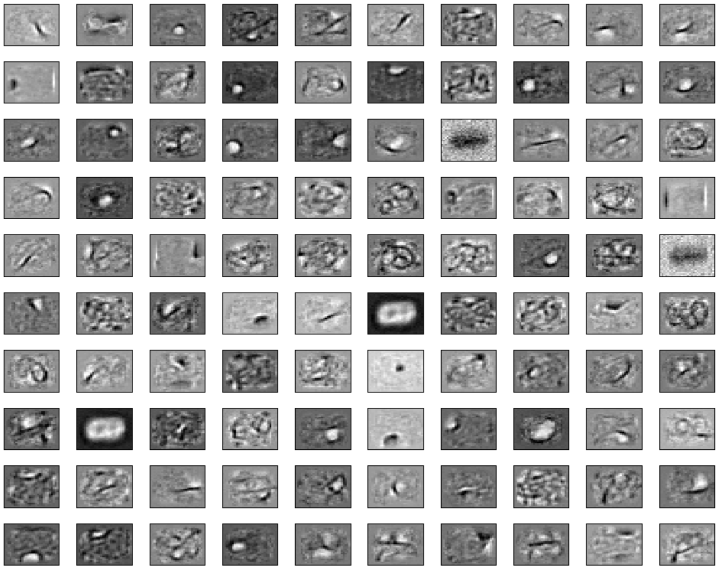

Figure 5: Sample learned weights W for the sparse mixed norm RBM using the MNIST data set

- The RIMES data set which was created by asking volunteers to write hand written letters for different scenarios. In this paper we used the digit set of the data base. In total the set we used was comprised of 37200 images of different sizes (29800 train - 7400 test). Further information can be obtained at `www.rimes-database.fr`.
- The USPS digit data set that we used is comprised of 9280 (7280 train - 2000 test), 16×16 images. The extracted images were scanned from mail in working U.S. Post Offices Hull (1994).

In order to achieve the cross-training and testing all images were resized to have the same size as the MNIST dataset (28×28) given its extensive use in this task. All images were also checked to ensure that orientations/translations were uniform across the data sets. No other pre-processing was employed. Example images from the three datasets can be seen in figure 8

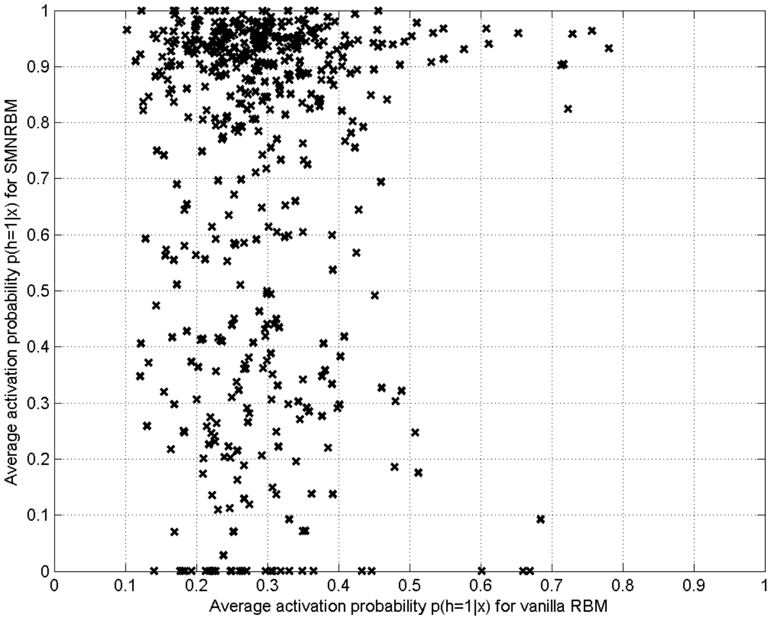

Figure 6: Average of hidden unit activation probabilities for the sparse mixed norm RBM using a batch of the MNIST data set. Y-axis: Hidden unit activation probabilities for SMNRBM. X-axis: Hidden unit activations for vanilla RBM

4.2 Experimental Results

In this section we offer the classification accuracies for digit recognition under the domain adaptation framework using the three different datasets (MNIST, RIMES and UPS) as well as three different architectures that are based on the use of the general sparse penalty RBM when pre-training the DBNs.

In order to provide a comparative baseline, Table 3 depicts the classification accuracies using vanilla DBNs with no added sparse penalty i.e. $\lambda_1 = 0$ and $\lambda_2 = 0$. As shown, the efficacy of the models is highly coupled with the type of training data that is utilized.

Table 4 provides results for the mixed norm DBN with $\lambda_1 = 0.1$ and $\lambda_2 = 0$ with a group size of 5. the results indicate that when we are testing on the UPS dataset there is a significant decrease in the classification accuracy $\sim 7\% - 10\%$. However, when training on the UPS dataset we see an increase in the classification accuracy for both the MNIST and RIMES datasets. This may be attributed to the fact that the UPS dataset has the smallest bounding box as it relates to the placement of the digit and thus the induced sparsity appears

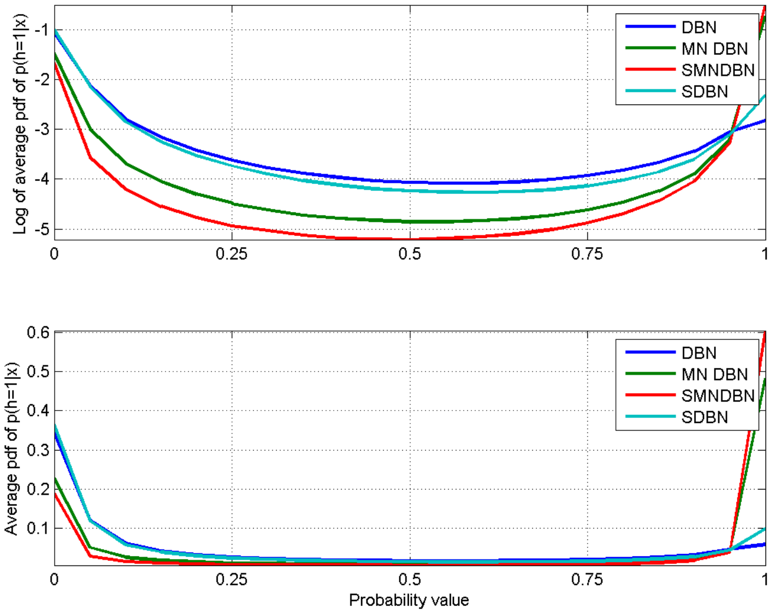

Figure 7: Average Probability density function for activation probabilities, $p(\boldsymbol{h} = \mathbf{1}|\boldsymbol{x})$. Top: Log of the average pdf Bottom: Average pdf

to be sensitive to the relative distances of the digits within the bounding box.

On the contrary, we see that there is a significant increase in the accuracy when we train with the MNIST dataset and test with the RIMES dataset. Both of these datasets are similar regarding their bounding boxes and relative distances of the digits within the image.

Continuing, in Table 5 we notice that for the SDBN we are able to achieve marginally better classification accuracies for almost all of the different pairs. This can be attributed to the fact that the l_1 penalty forces the hidden units to zero which may result in a better representation of the different classes in the underlying data.

Finally, Table 6 provides the cross-training/testing results for the SMNRBM. As discussed previously, the SMNRBM appears to minimize the sparse effect by setting the activation probabilities to high values. This explains the moderate behavior of this architecture in terms of its classification accuracy.

Table 1: Classification accuracies for the different architectures based on the general sparse penalty.

Architecture	MNIST	RIMES	UPS
DBN	**98.83**%	99.30%	94.85%
MN DBN (5)	98.63%	99.24%	94.25%
MN DBN (20)	**98.83**%	99.36%	95.30%
MN DBN (100)	98.77%	**99.38**%	94.55%
SMN DBN (5)	97.03%	99.15%	92.30%
SDBN	98.78%	99.27%	**94.90**%

Table 2: CPU times for the different architectures based on the general sparse penalty.

Architecture	MNIST	RIMES	UPS
DBN	167.9h	132.3h	21.43h
MN DBN (5)	49.9h	25.2h	5.5h
MN DBN (20)	95.2h	53.8h	8.9h
MN DBN (100)	x	67.8h	11.2h
SMN DBN (5)	46.5h	23.02h	5.6h
SDBN	x	73.7h	11.9h

5 Conclusions

In this paper we provided a theoretical framework for a general sparse penalty in the RBMs leading to the Sparse Mixed Norm RBM (SMNRBM). This general penalty allows the creation of multiple architectures by selecting the regularizers, λ_1 and λ_2 in such a way that will adequately represent the underlying data distribution.

We also utilized the SMNRBMs to pre-train efficient DBNs for classifying digits using three different datasets. In an effort to explore the ability of the generalized sparse penalty across datasets in a domain adaptation framework we cross-trained and cross-tested using the different architectures.

Although we were able to achieve marginal increases in the classification accuracies for hand written digit recognition, it is evident that a more robust methodology is needed to capture the relationships between the different datasets when applying the SMNRBM for domain adaptation applications. This methodology can include the use of a structured sparse penalty that would capture the underlying structures of the data distribution.

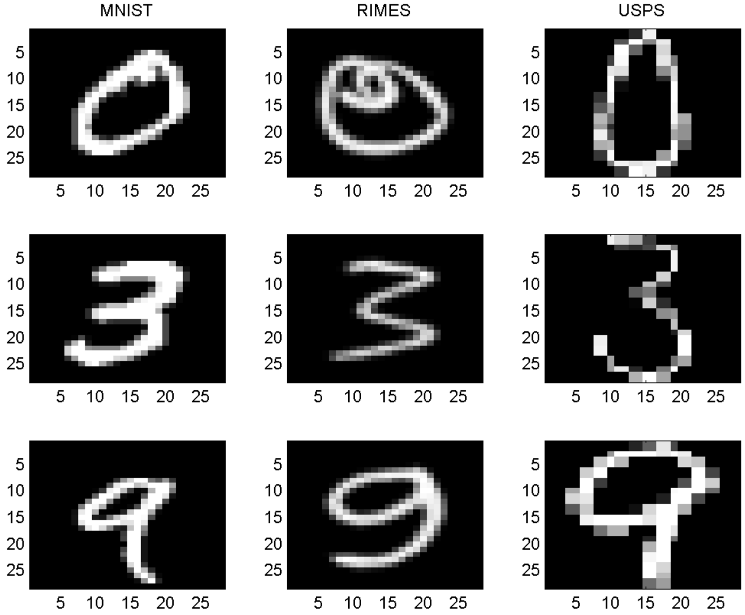

Figure 8: Examples of images from the three datasets MNIST (left), RIMES (center) and USPS (right)

References

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy layer-wise training of deep networks. In Schölkopf, B., Platt, J., and Hoffman, T. (eds.), *Advances in Neural Information Processing Systems 19*, pp. 153–160. MIT Press, Cambridge, MA, 2007.

Friedman, J., Hastie, T., and Tibshirani, R. A note on the group lasso and a sparse group lasso, 2010.

Hinton, G. E. Learning multiple layers of representation. *Trends in Cognitive Sciences*, 11:428–434, 2007.

Hinton, G. E. A practical guide for training restricted boltzmann machines, 2010.

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

Hinton, G. E., Osindero, S., and Teh, Y. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(5786):1527–1554, 2006.

Table 3: Classification accuracies for domain adaptation using vanilla DBNs.

Training/Testing	MNIST	RIMES	UPS
MNIST	–	82.75%	**53.80**%
RIMES	**69.27**%	–	58.55%
UPS	30.35%	41.89%	–

Table 4: Classification accuracies for domain adaptation using Mixed Norm DBNs with a group size of 5 and $\lambda_1 = 0.1$.

Training/Testing	MNIST	RIMES	UPS
MNIST	–	64.90%	47.90%
RIMES	67.22%	–	47.40%
UPS	30.83%	**45.19**%	–

Hinton, Geoffrey. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:2002, 2000.

Hull, J. J. A database for handwritten text recognition research. *IEEE Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

Lee, H., Ekanadham, C., and Ng, A. Sparse deep belief net model for visual area v2. In Platt, J.C., Koller, D., Singer, Y., and Roweis, S. (eds.), *Advances in Neural Information Processing Systems 20*, pp. 873–880. MIT Press, Cambridge, MA, 2008.

Luo, H., Shen, R., Niu, C., and Ullrich, C. Sparse group restricted boltzmann machines. In *Proceedings of the 25th Conference on Artificial Intelligence (AAAI 2011)*, pp. 1207–1216, San Francisco, CA, 2011. AAAI Press.

Olshausen, B. A. and Field, D. J. Sparse coding with an overcomplete basis set: a strategy employed by v1? *Vision Research*, 37(5786):3311–3325, 1997.

Ranzato, M., Poultney, C., Chopra, S., and LeCun, Y. Efficient learning of sparse representations with an energy-based model. In Schölkopf, B., Platt, J., and Hoffman, T. (eds.), *Advances in Neural Information Processing Systems 19*, pp. 1137–1144. MIT Press, Cambridge, MA, 2007.

Ranzato, M., Boureau, Y., and LeCun, Y. Sparse feature learning for deep belief networks. In Platt, J.C., Koller, D., Singer, Y., and Roweis, S. (eds.), *Advances in Neural Information Processing Systems 20*, pp. 1185–1192. MIT Press, Cambridge, MA, 2008.

Sra, S., Nowozin, S., and Wright, S. J. (eds.). *Optimization for Machine Learning*. MIT Press, 2011.

Table 5: Classification accuracies for domain adaptation using Sparse l_1 DBNs for $\lambda_2 = 0.01$.

Training/Testing	MNIST	RIMES	UPS
MNIST	–	**82.85**%	49.75%
RIMES	66.79%	–	**58.95**%
UPS	33.71%	39.86%	–

Table 6: Classification accuracies for domain adaptation using Sparse Mixed Norm DBNs for a group size of 5 with $\lambda_1 = 0.1$ and $\lambda_2 = 0.01$.

Training/Testing	MNIST	RIMES	UPS
MNIST	–	60.4%	46.15%
RIMES	67.85%	–	45.55%
UPS	**33.97**%	36.20%	–