ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Learning Sparse Feature Representations using Probabilistic Quadtrees and Deep Belief Nets

Saikat Basu[1], Manohar Karki[1], Sangram Ganguly[2],
Robert DiBiano[1], Supratik Mukhopadhyay[1] and Ramakrishna Nemani[3]

1- Louisiana State University - Department of Computer Science
Baton Rouge, Louisiana 70803 - USA

2- Bay Area Environmental Research Institute (BAERI)
/NASA Ames Research Center, Moffett Field, California 94035 - USA

3- NASA Advanced Supercomputing Division,
NASA Ames Research Center, Moffett Field, California 94035 - USA

Abstract. Learning sparse feature representations is a useful instrument for solving an unsupervised learning problem. In this paper, we present three labeled handwritten digit datasets, collectively called n-MNIST. Then, we propose a novel framework for the classification of handwritten digits that learns sparse representations using probabilistic quadtrees and Deep Belief Nets. On the MNIST and n-MNIST datasets, our framework shows promising results and significantly outperforms traditional Deep Belief Networks.

1 Introduction

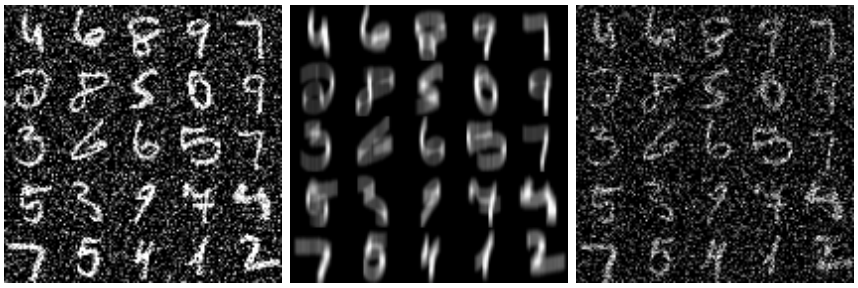
Deep Learning has gained popularity over the last decade due to its ability to learn data representations in an unsupervised manner and generalize to unseen data samples using hierarchical representations. The most recent and best-known Deep learning model is the *Deep Belief Network*[1]. In [2], Deep Belief Networks have been used for modeling acoustic signals and have been shown to outperform traditional approaches using Gaussian Mixture Models for Automatic Speech Recognition (ASR). Deep Belief Network is trained one layer at a time using Restricted Boltzmann Machines (RBM). A sparse feature learning algorithm for Deep Belief Networks was proposed in [3]. However, their work was focused on maximization of information content in the learned representations. Restricted Boltzmann Machines, on the other hand, are trained by minimizing a contrastive term in the loss function.

The main contributions of our work are twofold – (1) We first present three labeled handwritten digit datasets, collectively called n-MNIST, created by adding white gaussian noise, motion blur and reduced contrast to the original MNIST dataset[4]. (2) Then, we present a framework for the classification of handwritten digits that a) learns probabilistic quadtrees from the dataset, b) performs a Depth First Search on the quadtrees to create sparse representations in the form of linear vectors, and c) feeds the linear vectors into a Deep Belief Network for classification. On the MNIST and n-MNIST datasets, our framework shows promising results and significantly outperforms traditional Deep Belief Networks.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

2 Datasets[1]

We evaluate our framework on the MNIST dataset[4] of handwritten digits as well as three artificial datasets collectively called n-MNIST (noisy MNIST) created by adding – (1) additive white gaussian noise, (2) motion blur and (3) a combination of additive white gaussian noise and reduced contrast to the MNIST dataset. Some of the images from these datasets are shown in Figure 1.



(a) MNIST with Additive White Gaussian Noise (b) MNIST with Motion Blur (c) MNIST with AWGN and reduced contrast

Fig. 1: Example images from the n-MNIST dataset created as part of the experiments

3 Probabilistic Quadtrees for Learning Sparse Representations

We propose a novel technique based on probabilistic quadtrees that can reduce the dimensionality of a dataset in a probabilistically sound way. We learn the structure of the quadtree from the samples of a dataset. A quadtree splits each image into four equi-sized windows, and then performs a test of homogeneity on each image window. If a block meets the homogeneity criterion, it is not divided any further into sub-windows. If otherwise, it does not meet the criterion, it is again subdivided into four sub-windows, and the test criterion is in turn applied to those smaller windows. This process is repeated on all the sub-windows until each meets the homogeneity criterion. The resulting data structure can have windows of several different sizes. The homogeneity criterion can be defined as follows - Split a block if the maximum value of the block elements minus the minimum value is greater than a threshold τ. Threshold τ is specified as a value between 0 and 1 (chosen here as 0.27 by experiments). Denoting the homogeneity criterion for sample d as H_d, this can be formally presented as follows:

[1]The datasets are available at the web link [5] along with a detailed description of the methods and parameters used to create them

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

$$H_d = \begin{cases} true, & \text{if} \max_{i \in d}(i) - \min_{i \in d}(i) \leq \tau \mid \tau \in [0,1] \\ false, & \text{if} \max_{i \in d}(i) - \min_{i \in d}(i) > \tau \mid \tau \in [0,1] \end{cases} \quad (1)$$

Alternatively, the homogeneity criterion can be considered proportional to the standard deviation of the probability distribution of the dataset. So, higher the standard deviation, higher the average texture of a block and higher is the probability of the block being divided into sub-blocks.

In the learned quadtree structure for a given dataset, a node is divided into smaller windows if the homogeneity criterion is not met for any sample in the dataset. The node is not divided into smaller windows only if the homogeneity criterion is met by all samples in the dataset.

We can consider each node of the quadtree as a binary random variable X, which can take one of two values 1 or 0 based on whether it is divided into smaller windows or not. So, for a total of N samples in dataset D, the random variable X may take on one of $N+1$ possible split state values: one value for each of the samples not meeting the homogeneity criterion, and one value indicating that all samples meet the homogeneity criterion. This can be formally presented as follows:

$$X = \begin{cases} 1, & \text{if } \exists d \in D \mid D = \{d_0, d_1, d_2, ..., d_N\} \cap \{H_d = false\} \\ 0, & \text{if } \forall d \in D \mid D = \{d_0, d_1, d_2, ..., d_N\} \cap \{H_d = true\} \end{cases} \quad (2)$$

Once learned, the probabilistic quadtree helps in reducing the dimensionality of the data, which captures the statistics of the training samples in the dataset. A depth first search on the learned tree yields a linear vector that is then fed into an unsupervised learning framework.

4 Deep Belief Network for Feature Learning

Deep Belief Network (DBN) consists of multiple layers of stochastic, latent variables trained using an unsupervised learning algorithm followed by a supervised learning phase using Feedforward Backpropagation Neural Networks. In the unsupervised pre-training stage, each layer is trained using a *Restricted Boltzmann Machine* (RBM). Once trained, the weights of the DBN are used to initialize the corresponding weights of a Neural Network [6]. A Neural Network initialized in this manner converges much faster than an otherwise uninitialized one. A DBN is a graphical model [7] where neurons of the hidden layer are conditionally independent of each other given a particular configuration of the visible layer and vice versa. A DBN can be trained layer-wise by iteratively maximizing the conditional probability of the input vectors or visible vectors given the hidden vectors and a particular set of layer weights. As shown in [1], this layer-wise training can help in improving the variational lower bound on the probability of the input training data, which in turn leads to an improvement of the overall generative model. We first provide a formal introduction to the Restricted

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

Boltzmann Machine. The RBM can be denoted by the energy function:

$$E(v,h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j h_j w_{i,j} v_i \tag{3}$$

where, the RBM consists of a matrix of layer weights $W = (w_{i,j})$ between the hidden units h_j and the visible units v_i. The a_i and b_j are the bias weights for the visible units and the hidden units respectively. The RBM takes the structure of a bipartite graph and hence it only has inter-layer connections between the hidden or visible layer neurons but no intra-layer connections within the hidden or visible layers. So, the visible unit activations are mutually independent given a particular set of hidden unit activations and vice versa [8]. Hence, by setting either h or v constant, we can compute the conditional distribution of the other as follows:

$$P(h_j = 1|v) = \sigma(b_j + \sum_{i=1}^{m} w_{i,j} v_i) \tag{4}$$

$$P(v_i = 1|h) = \sigma(a_i + \sum_{j=1}^{n} w_{i,j} h_j) \tag{5}$$

where, σ denotes the log sigmoid function:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

The training algorithm maximizes the expected log probability assigned to the training dataset D. So if the training dataset D consists of the visible vectors v, then the objective function is as follows:

$$\underset{W}{\operatorname{argmax}}\, E\Big[\sum_{v \in V} \log P(v)\Big] \tag{7}$$

A Restricted Boltzmann Machine is trained using a *Contrastive Divergence* algorithm [8]. Once trained the DBN is used to initialize the weights of a feed-forward backpropagation neural network that is then used for classification.

5 Results and Comparative Studies

Various network architectures along with the test set error for the traditional DBN framework and the probabilistic quadtree based framework on the MNIST and the three n-MNIST datasets are listed in Tables 1 and 2. From the Tables, it is evident that our best performing network outperforms the best traditional Deep Belief Network on both the MNIST and n-MNIST datasets. On the MNIST dataset, our best network exhibits a relative improvement of ∼25% over the traditional DBN. For the n-MNIST dataset, it provides a relative improvement of ∼36% for Additive White Gaussian Noise (AWGN), ∼26% for Motion Blur and ∼12% for AWGN and Reduced contrast.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

	MNIST		**n-MNIST with AWGN**	
Architecture (Neurons)	**Test Error DBN(%)**	**Test Error Ours(%)**	**Test Error DBN(%)**	**Test Error Ours(%)**
50-50	4.64	2.93	89.95	13.41
100-100	3.01	2.45	91.43	12.01
150-150	2.34	2.21	89.95	13.49
200-200	2.08	1.96	88.49	10.56
250-250	1.93	1.83	88.49	13.00
300-300	2.02	1.80	68.18	11.24
350-350	1.96	1.74	90.31	13.15
400-400	1.95	1.67	49.27	10.96
450-450	1.93	**1.38**	**32.26**	12.62
500-500	**1.86**	1.43	69.68	**9.93**

Table 1: Test Error of a traditional DBN and our framework with various architectures on MNIST and n-MNIST with AWGN

	n-MNIST with Motion Blur		**n-MNIST with AWGN and Reduced Contrast**	
Architecture (Neurons)	**Test Error DBN(%)**	**Test Error Ours(%)**	**Test Error DBN(%)**	**Test Error Ours(%)**
50-50	5.64	4.17	10.21	9.29
100-100	4.68	3.31	**9.43**	9.21
150-150	3.99	3.29	16.40	9.00
200-200	3.74	3.03	15.57	8.79
250-250	3.74	**2.60**	52.31	8.94
300-300	**3.50**	3.04	32.29	8.28
350-350	3.82	2.91	86.31	8.90
400-400	3.74	3.01	68.78	8.31
450-450	3.91	2.75	51.32	8.36
500-500	3.66	2.83	68.19	**7.84**

Table 2: Test Error of a traditional DBN and our framework with various architectures on n-MNIST with Motion Blur; and with AWGN and Reduced Contrast

6 Discussion and Future Directions

Our learning framework based on probabilistic quadtrees significantly outperforms traditional Deep Belief Networks on both the MNIST and n-MNIST datasets. Probabilistic quadtrees help in generating sparse representations for the dataset and significantly improve the discriminative power of the framework.

We plan to investigate the use of various pooling techniques like SPM [9] as well as certain sparse representations like sparse coding [10] to handle n-MNIST. Hierarchical representations like Convolutional DBN [11] are other useful can-

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

didates for investigation. We believe that n-MNIST will help researchers better apply and extend the research on understanding representations for noisy object recognition datasets.

Acknowledgment

The project is supported by Army Research Office (ARO) under Grant #W911-NF1010495 and NASA Carbon Monitoring System through Grant #NNH14ZD-A001NCMS. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the ARO or the United States Government.

References

[1] Geoffrey E. Hinton and Simon Osindero. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:2006, 2006.

[2] Abdel-rahman Mohamed, George E. Dahl, and Geoffrey E. Hinton. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech & Language Processing*, 20(1):14–22, 2012.

[3] Marc'Aurelio Ranzato, Y-lan Boureau, and Yann Lecun. Sparse Feature Learning for Deep Belief Networks. In *Advances in Neural Information Processing Systems*, 2008.

[4] WWW. Mnist. `http://yann.lecun.com/exdb/mnist/`.

[5] WWW. Datasets. `https://tigerbytes2.lsu.edu/users/sbasu8/n-mnist/`.

[6] Yoshua Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, January 2009.

[7] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[8] Miguel A. Carreira-Perpinan and Geoffrey E. Hinton. On contrastive divergence learning. 2005.

[9] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. CVPR '06, pages 2169–2178, 2006.

[10] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng. Efficient sparse coding algorithms. In *NIPS*, pages 801–808. NIPS, 2007.

[11] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. ICML '09, pages 609–616, 2009.