The 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
24-26 September 2015, Warsaw, Poland

A New Technique for Restricted Boltzmann Machine Learning

Vladimir Golovko [1], Aliaksandr Kroshchanka [2], Volodymyr Turchenko [3],
Stanislaw Jankowski [4], Douglas Treadwell [5]
[1,2] Brest State Technical University, Moskowskaja 267, Brest, 224017, Belarus, gva@bstu.by
[3] University of Lethbridge, University Drive 4401, Lethbridge, AB, T1K 3M4, Canada
[4] Warsaw University of Technology, Nowowiejska 15/19, Warsaw, 00-665, Poland
[5] Beckon, 107 South B Street, Suite 300, San Mateo, CA, 94401, USA

***Abstract*—Over the last decade, deep belief neural networks have been a hot topic in machine learning. Such networks can perform a deep hierarchical representation of input data. The first layer can extract low-level features, the second layer can extract high-level features and so on. In general, deep belief neural network represents many-layered perceptron and permits to overcome some limitations of conventional multilayer perceptron due to deep architecture. In this work we propose a new training technique called Reconstruction Error-Based Approach (REBA) for deep belief neural network based on restricted Boltzmann machine. In contrast to classical Hinton's training approach, which is based on a linear training rule, the proposed technique is based on a nonlinear learning rule. We demonstrate the performance of REBA technique for the MNIST dataset visualization. The main contribution of this paper is a novel view on the training of a restricted Boltzmann machine.**

Keywords—Restricted Boltzmann machine, deep learning, data visualization, machine learning.

I. INTRODUCTION

Deep learning is a revolutionary technique in machine learning and has been successfully applied to many problems in artificial intelligence, namely speech recognition, computer vision, natural language processing, and data visualization [1-9]. Deep belief neural networks (DBNs) [1-4] consist of many hidden layers, can perform a deep hierarchical transformation of the input data and as a result have been found to have better performance and more representational power than traditional neural networks. This paper deals with a learning technique for restricted Boltzmann machine (RBM) and correspondingly for deep belief neural networks training. The conventional approach to training the RBM uses an energy-based model. We propose a new technique called Reconstruction Error-Based Approach (REBA) for RBM training. This approach in contrast of energy-based models is based on the minimization of reconstruction mean square error (MSE) in hidden and reconstructed layers of RBM. We have shown that the classical Hinton's equations of RBM training are special case of the proposed technique.

The rest of the paper is organized as follows. Section 2 introduces the conventional approach for restricted Boltzmann machine training based on an energy model. In Section 3 we propose novel approach for inference of RBM training rules. Section 4 demonstrates the results of experiments and finally Section 5 concludes the paper.

II. RELATED WORKS

Deep belief neural networks were investigated in many studies [1-7]. As mentioned before, the first layer of DBN can extract low-level features, the second layer can extract high-level features, i.e. it can perform a deep hierarchical representation of the input data as shown in Fig. 1.

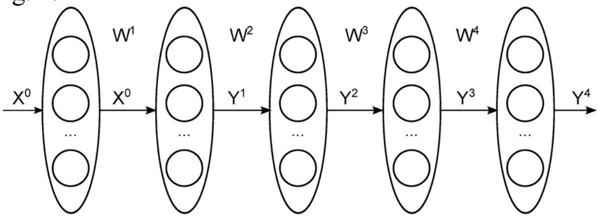

Figure 1. Deep belief neural network.

The *j*-th output unit for *k*-th layer is given by

$$y_j^k = F(S_j^k), \tag{1}$$

$$S_j^k = \sum_{i=1} w_{ij}^k y_i^{k-1} + T_j^k \ , \tag{2}$$

where F is the activation function, S_j^k is the weighted sum of the *j*-th unit, w_{ij}^k is the weight from the *i*-th unit of the (*k*-1)-th layer to the *j*-th unit of the *k*-th layer, T_j^k is the threshold of the *j*-th unit.

For the first layer

$$y_i^0 = x_i \, . \tag{3}$$

In common case we can write that

$$Y^k = F(S^k) = F(W^k Y^{k-1} + T^k), \quad (4)$$

where W is the weight matrix, Y^{k-1} is the output vector for (k-1)-th layer, T^k is the threshold vector.

It should be also noted that the output of the DBN is often defined using softmax function:

$$y_j^F = soft\max(S_j) = \frac{e^{S_j}}{\sum_l e^{S_l}}. \quad (5)$$

In general the training of deep belief neural networks consists of two stages:

1. Pre-training of neural network using the greedy layer-wise approach. This procedure is started from first layer and performed in an unsupervised manner.
2. Fine-tuning all of parameters of neural network using back-propagation or wake-sleep algorithm.

The DBN pre-training approach is based on the restricted Boltzmann machine or auto-encoder approaches [1-9]. In accordance with the greedy layer-wise training procedure, in the beginning the first layer of DBN is trained, using RBM or auto-encoder training rules and its parameters are fixed, after this the next layer is trained until all layers are processed. As a result, the good initialization of neural network is performed and we could use back-propagation or wake-sleep algorithm for fine tuning parameters of whole neural network. In this work, we will consider the DBN pre-training technique based on restricted Boltzmann machine. The deep belief neural networks can be represented as a set of restricted Boltzmann machines. Therefore in this case the RBM is the main building block for deep belief neural networks. The classical Hinton's training of RBM is based on an energy model and training rules take into account only linear nature of neural units [1-2].

Let's examine the restricted Boltzmann machine, which consists of two layers of units: visible and hidden (Fig. 2).

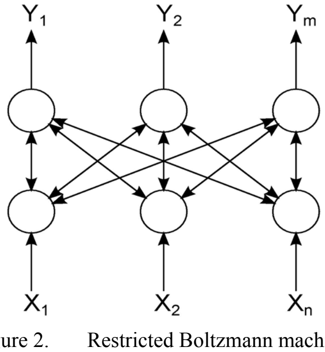

Figure 2. Restricted Boltzmann machine.

The hidden units of RBM are feature detectors which capture the regularities of the input data. Each unit has a bidirectional connection to each other unit in the system. The restricted Boltzmann machine can represent any discrete distribution, if enough hidden units are used [5]. The RBM is a stochastic neural network and the states of visible and hidden units are defined using probabilistic version of sigmoid activation function:

$$p(y_j \mid x) = \frac{1}{1+e^{-S_j}}, S_j = \sum_i^n w_{ij} x_i + T_j, \quad (6)$$

$$p(x_i \mid y) = \frac{1}{1+e^{-S_i}}, S_i = \sum_j^m w_{ij} y_j + T_i. \quad (7)$$

It should be noted that the visible and hidden units are conditionally independent:

$$P(x \mid y) = \prod_{i=1}^n P(x_i \mid y), \quad (8)$$

$$P(y \mid x) = \prod_{j=1}^m P(y_j \mid x). \quad (9)$$

As can be seen, the states of all the units are obtained through probability distribution. The key idea of RBM training is to reproduce the distribution of the input data using the states of the hidden units as closely as possible. This is equivalent to maximizing the likelihood of the data distribution P (x) by the modification of synaptic weights using the gradient of the log probability of the input data. Using this approach Hinton [1-4] proposed to use a contrastive divergence (CD) technique for RBM learning. It is based on Gibbs sampling. In case of CD-1 the training rule is defined as

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(x_i(0)y_j(0) - x_i(1)y_j(1)),$$
$$T_i(t+1) = T_i(t) + \alpha(x_i(0) - x_i(1)),$$
$$T_j(t+1) = T_j(t) + \alpha(y_j(0) - y_j(1)).$$

If we use CD-k

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(x_i(0)y_j(0) - x_i(k)y_j(k)),$$
$$T_i(t+1) = T_i(t) + \alpha(x_i(0) - x_i(k)),$$
$$T_j(t+1) = T_j(t) + \alpha(y_j(0) - y_j(k)).$$

In this case, the first term in training rule denotes the data distribution at the time t=0 and the second term is the model distribution of reconstructed states at the step t=k. Here α is the learning rate. As can be seen from these equations that the training rules for RBM is essentially minimizing the difference between the original data and the synthesized samples from model. The synthesized data can be obtained using a Gibbs sampling algorithm. It is easy to show, that RBM training rules are based on linear representation of neural units [8, 9].

Training an RBM is based on presenting a training sample to the visible units, then using the CD-n procedure we could compute the binary states of the hidden units $p(y \mid x)$, perform sampling the visible units (reconstructed states) $p(x \mid y)$ and so on. After performing these iterations the weights and biases of restricted Boltzmann machine are updated. Then we should stack another hidden layer to train a new RBM. This approach is applied to all layers of the deep belief neural network (greedy layer-wise training). As a result of such an unsupervised pre-training, we can obtain a good initialization of the neural network. Finally a fine-tuning algorithm of the whole neural network is performed. The main shortcoming of the traditional training rule for RBM is the linear nature of neurons in terms of learning as will be shown in the next section.

III. A New View on an RBM Learning Rule

In this section we propose a novel approach in order to infer RBM training rules. It is based on minimizing the reconstruction MSE, which we can obtain using a simple iterations of Gibbs sampling. In comparison with traditional energy-based method, which is based on a linear representation of neural units, the proposed approach permits to take into account nonlinear nature of neural units.

We represent the RBM, using three layers (visible, hidden and visible) [8, 9] as shown in Fig. 3.

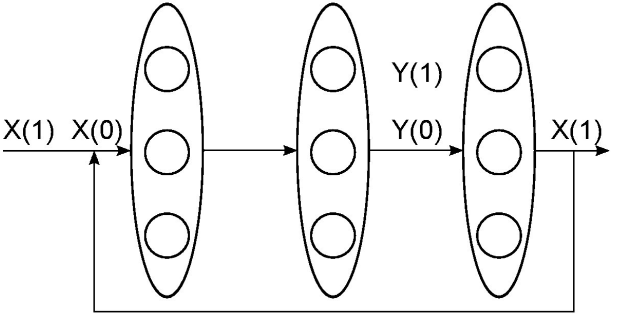

Figure 3. Expanded representation of RBM.

The Gibbs sampling will consist of the following procedure. Let $x(0)$ will be input data, which move to the visible layer at time 0. Then the output of hidden layer is defined as follows:

$$y_j(0) = F(S_j(0)), \tag{10}$$

$$S_j(0) = \sum_i w_{ij} x_i(0) + T_j . \tag{11}$$

The inverse layer reconstructs the data from hidden layer. As a result we can obtain $x(1)$ at time 1:

$$x_i(1) = F(S_i(1)), \tag{12}$$

$$S_i(1) = \sum_j w_{ij} y_j(0) + T_i . \tag{13}$$

After this the $x(1)$ enters to the visible layer and we can obtain the output of the hidden layer by the following way:

$$y_j(1) = F(S_j(1)), \tag{14}$$

$$S_j(1) = \sum_i w_{ij} x_i(1) + T_j . \tag{15}$$

Continuing the given process we can obtain on a step k, that

$$y_j(k) = F(S_j(k)),\ S_j(k) = \sum_i \omega_{ij} x_i(k) + T_j ,$$

$$x_i(k) = F(S_i(k)),\ S_i(k) = \sum_j w_{ij} y_j(k-1) + T_i .$$

The purpose of training this neural network is to minimize the reconstruction MSE in the hidden and visible layers. In case of CD-k, the reconstruction MSE is defined as

$$E_s = \frac{1}{2}\sum_{l=1}^{L}\sum_{j=1}^{m}\sum_{p=1}^{k}(y_j^l(p) - y_j^l(p-1))^2 + \\ + \frac{1}{2}\sum_{l=1}^{L}\sum_{i=1}^{n}\sum_{p=1}^{k}(x_i^l(p) - x_i^l(p-1))^2. \tag{16}$$

In case of CD-1 we can write

$$E_s = \frac{1}{2}\sum_{l=1}^{L}\sum_{j=1}^{m}(y_j^l(1) - y_j^l(0))^2 + \\ + \frac{1}{2}\sum_{l=1}^{L}\sum_{i=1}^{n}(x_i^l(1) - x_i^l(0))^2. \tag{17}$$

where L is the number of training patterns.

Theorem 1. Maximization of the log-likelihood input data distribution P(x) in the space of synaptic weights of restricted Boltzmann machine is equivalent to the minimizing the reconstruction MSE in the same space using linear neurons in RBM.

This theorem states that if we use identity activation function for RBM units, then the CD-k training rule for RBM for minimizing reconstruction mean squared error (16) will be the following:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(x_i(0)y_j(0) - x_i(k)y_j(k)),$$
$$T_j(t+1) = T_j(t) + \alpha(y_j(0) - y_j(k)),$$
$$T_i(t+1) = T_i(t) + \alpha(x_i(0) - x_i(k)).$$

As can be seen, the last equations are identical to the conventional RBM training rules. Thus the conventional RBM training rules are linear. Therefore we shall call such a machine as linear RBM.

Corollary 1. Linear restricted Boltzmann machine from the training point of view is equivalent to the linear PCA (auto associative) neural network if we use Gibbs sampling during training.

Corollary 2. The training rule for nonlinear restricted Boltzmann machine in case of CD-k is defined as

$$w_{ij}(t+1) = w_{ij}(t) -$$
$$\alpha(\sum_{p=1}^{k}\big(y_j(p) - y_j(p-1)\big)x_i(p)F'(S_j(p)) +$$
$$\big(x_i(p) - x_i(p-1)\big)y_j(p-1)F'(S_i(p))),$$
$$T_j(t+1) = T_j(t) -$$
$$-\alpha(\sum_{p=1}^{k}\big(y_j(p) - y_j(p-1)\big)F'(S_j(p))),$$
$$T_i(t+1) = T_i(t) -$$
$$-\alpha(\sum_{p=1}^{k}\big(x_i(p) - x_i(p-1)\big)F'(S_i(p))).$$

Corolarry 3. The training rule for nonlinear restricted Boltzmann machine in case of CD-1 is the following:

$$w_{ij}(t+1) = w_{ij}(t) -$$
$$\alpha((y_j(1) - y_j(0))x_i(1)F'(S_j(1)) +$$
$$+(x_i(1) - x_i(0))y_j(0)F'(S_i(1))),$$
$$T_j(t+1) = T_j(t) - \alpha(y_j(1) - y_j(0))F'(S_j(1)),$$
$$T_i(t+1) = T_i(t) - \alpha(x_i(1) - x_i(0))F'(S_i(1)).$$

Thus, as can be seen, the classical Hinton's equations for RBM training are special case of the proposed technique.

In this section we have obtained the new training rules for restricted Boltzmann machine. It is based on minimizing the reconstruction mean square error, which we can obtain using simple iterations of Gibbs sampling. The proposed approach permits to take into account the derivatives of nonlinear activation function for neural network units. We have called the proposed technique as Reconstruction Error-Based Approach (REBA). As it is mathematically proved above, the classical Hinton's equations for RBM training are special case of the proposed technique.

IV. Experiments

In order to illustrate the performance of REBA technique we present simulation results for visualization of handwritten digits using MNIST dataset. The MNIST dataset contains 28x28 handwritten digits in gray-scale and has a training set of 60000 samples, and a test set of 10000 samples. For mapping 784D digits data to a 2D feature space, the deep auto-encoder with topology 784-1000-500-250-2 is used.

The identity function for units is applied in bottleneck layer. In other layers, the sigmoid activation function is used. We have implemented pre-training of deep auto-encoder using greedy layer-wise approach with RBM and REBA techniques. This procedure is started from the first layer and performed in an unsupervised manner. Finally fine-tuning of all parameters of the neural network using back-propagation algorithm is fulfilled. We have compared two techniques: REBA and conventional RBM. We have used the following training parameters: the learning rate for pre-training is 0.2 for REBA and 0.05 for classic conventional RBM for all layers except the bottleneck layer. The learning rate for bottleneck layer is 0.001. The comparative analysis of two techniques is shown in the table 1.

TABLE I. Comparative analysis

Technique	MSE training	MSE test
Classic RBM	3.7801	4.0115
REBA	3.6490	3.8726

Fig. 4 depicts the evolution of mean square error depending on the number of the training epochs for the first layer of deep auto-encoder.

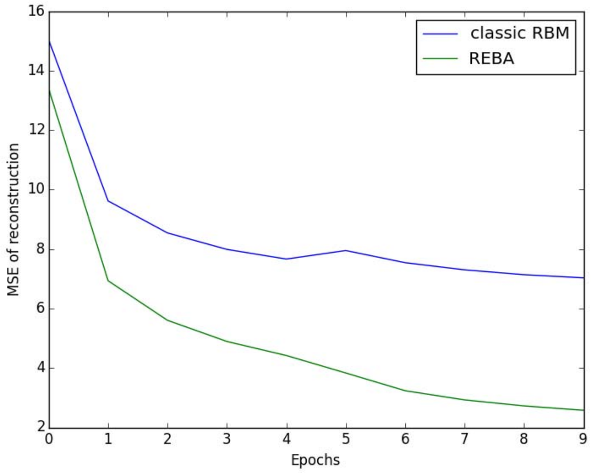

Figure 4. Evolution of MSE on the first layer RBM and REBA.

Visualization of the MNIST dataset on the basis of REBA for the first 500 test images for each class of digits is shown in Fig. 4.

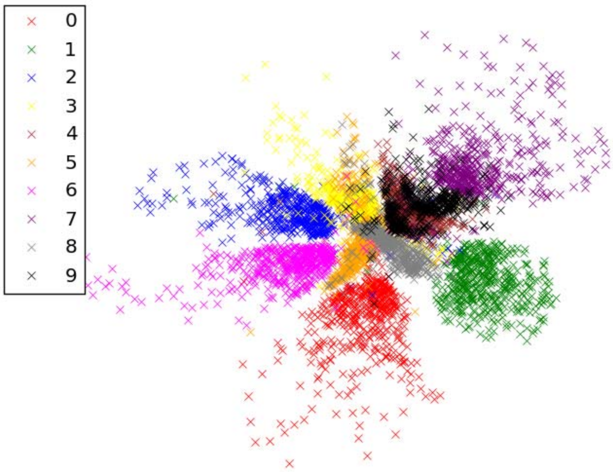

Figure 5. Visualization of handwritten digits.

V. Conclusion

We have presented a new approach to calculate a learning rule of a restricted Boltzmann machine. In comparison with the classical Hinton's energy-based method, based on maximization of the log-likelihood input data distribution, the proposed approach is based on minimization of reconstruction mean square error, which we can obtain using simple iterations of Gibbs sampling. The performance of the proposed approach is investigated on the task of visualization of the MNIST dataset. As a final note, we should say that the work presented in this paper is still exploratory in nature and we will continue our investigations of properties of the proposed Reconstruction Error-Based Approach in future.

References

[1] G. E. Hinton, S. Osindero, Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, issue 7, 2006, pp. 1527-1554.

[2] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, issue 8, 2002, pp. 1771-1800.

[3] G. Hinton, R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, 2006, pp. 504-507.

[4] G. E. Hinton, *A Practical Guide to Training Restricted Boltzmann Machines*, Tech. Rep. 2010-000), Toronto: Machine Learning Group, University of Toronto, 2010.

[5] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, issue 1, 2009, pp. 1-127.

[6] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, "Greedy layer-wise training of deep networks," In B. Scholkopf, J. C. Platt, T. Hoffman (Eds.), *Advances in Neural Information Processing Systems*, MA: MIT Press, Cambridge, vol. 11, 2007, pp. 153-160.

[7] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, S. Bengio, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, issue 3, 2010, pp. 625-660.

[8] V. Golovko, A. Kroshchanka, U. Rubanau, S. Jankowski, "Learning technique for deep belief neural networks," in book *Neural Networks and Artificial Intelligence*, vol. 440. *Communication in Computer and Information Science*, Springer, 2014, pp. 136-146.

[9] V. Golovko, *From Multilayer Perceptron to Deep Belief Neural Networks: Training Paradigms and Application, Lectures on Neuroinformatics*, Moscow, 2015, pp. 47-84.