Deep Neural Networks: A theory, application and new trends

Vladimir Golovko

Brest State Technical University and **National Research Nuclear University "MEPhI"**, Moskowskaja 267, Brest 224017, Belarus, gva@bstu.by

Abstract: *Over the last decade the deep neural networks are the revolutionary technique in the domain of artificial intelligence and machine learning. In the general case a deep neural network consists of multiple layers of neural units and can accomplish a deep hierarchical representation of their input data. The first layer extracts low-level features; the second layer detects higher level features, and as a result the deep neural network performs deep non-linear transformation of input data into more abstract level of representation. This paper provides an overview of deep neural networks and deep learning. Different deep learning techniques, including well-known and new approaches are discussed.*

***Keywords*:** Restricted Boltzmann machine, deep neural networks, deep learning.

1. INTRODUCTION

Over the last decade the deep neural networks are the powerful tool in the domain of machine learning and has been successfully applied to many problems in artificial intelligence, namely video, speech recognition, computer vision, natural language processing, data visualization, etc. [1-10]. This kind of neural network has been investigated in many studies [1-10]. Deep neural networks (DBN) [1-6] consist of many hidden layers and 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.

The important problem is training of deep neural network, because learning of such a network is much complicated compared to shallow neural networks. This is due to the vanishing gradient problem, poor local minima and unstable gradient problem. Therefore a lot of deep learning techniques were developed that permit us to overcome some limitations of conventional training approaches [1, 10]. Since 2006, the unsupervised pre-training of deep neural network was proposed [2]. In this case the training of deep neural networks consists of two stages: pre-training of the neural network using a greedy layer-wise approach and fine-tuning all the parameters of the neural network using back-propagation or the wake-sleep algorithm [2-6]. The pre-training of deep neural network is based on either the restricted Boltzmann machine (RBM) or auto-encoder approach [6, 12]. At present, the stochastic gradient descent (SGD) with rectified linear unit (ReLU) activation function is used for training of deep neural networks in supervised manner [1].

There exist different types of deep neural networks, namely deep belief neural networks, deep convolutional neural networks, deep recurrent neural networks, deep autoencoder and so on. This paper provides an overview of deep neural networks and deep learning. Different deep learning techniques, including well-known and new approaches are discussed. So, for instance, a new technique called "REBA" for the training of deep neural networks, based on the restricted Boltzmann machine is demonstrated. This approach in contrast to an energy-based model is based on the minimization of the reconstruction mean square error in the hidden and reconstructed layers of the RBM. We will show that classical equations for RBM training are a special case of the proposed technique. The experiments demonstrate a high potential of deep neural networks in real applications.

The rest of the paper is organized as follows. Section 2 introduces the contemporary techniques to deep learning. Section 3 describes the conventional approach for restricted Boltzmann machine training. Section 4 deals with SGD. In Section 5 we propose a novel approach for inference of RBM training rules. Section 5,6 and 7 present the current learning paradigm, conclusion and future of deep learning.

2. DEEP LEARNING: CONTEMPORARY TECHNIQUES

Let us consider the conventional approach to RBM machine learning [2-7]. It is based on energy-based model. As already mentioned the DBN consists of multiple layers and can perform a deep hierarchical representation of the input data as shown in Fig. 1.

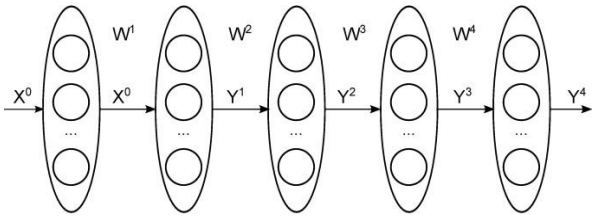

Fig.1 – Deep neural network.

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

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

where W is a 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 DBNN is often defined using softmax function:

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

There exist the following techniques for learning of deep neural networks: learning with pre-training and stochastic gradient descent approach (SGD) with rectified linear unit (ReLU) transfer function [1].

The learning with pre-training consists of two stages. The first stage is the pre-training of neural network using greedy layer-wise approach. This procedure is started from first layer and performed in unsupervised manner. The second one is fine-tuning all of parameters of neural network using back-propagation or wake-sleep algorithm [2-10].

The training with stochastic gradient descent approach is the online or mini-batch learning using conventional backpropagation algorithm. Use of rectified linear unit (ReLU) transfer function can help to avoid of vanishing gradient problem, poor local minima and unstable gradient problem due to due to the greater linearity of such kind of activation function.

The important stage of training deep neural networks is the pre-training of each layer of the DBN. DBN pre-training is based on either the restricted Boltzmann machine (RBM) or auto-encoder approach [1-10]. In accordance with the greedy layer-wise training procedure, in the beginning the first layer of the DBN is trained using RBM or auto-encoder training rules and its parameters are fixed. After this the next layer is trained, and so on. As a result a good initialization of the neural network is achieved and we can then use back-propagation or the wake-sleep algorithm for fine tuning the parameters of the whole neural network.

3. RESTRICTED BOLTZMANN MACHINE

The important stage of training deep neural networks is the pre-training of each layer of the DBN. DBN pre-training is based on either the restricted Boltzmann machine (RBM) or auto-encoder approach [1-10]. In accordance with the greedy layer-wise training procedure, in the beginning the first layer of the DBN is trained using RBM or auto-encoder training rules and its parameters are fixed. After this the next layer is trained, and so on. As a result a good initialization of the neural network is achieved and we can then use back-propagation or the wake-sleep algorithm for fine tuning the parameters of the whole neural network.

In this section we will consider the DBN pre-training technique based on the restricted Boltzmann machine. 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 traditional approach to RBM training is based on an energy model and training rules [2] which take into account only a linear nature of neural units. Let's examine the restricted Boltzmann machine, which consists of two layers of units: visible and hidden (Fig. 2). The restricted Boltzmann machine can represent any discrete distribution if enough hidden units are used [6]. As can be seen, the states of all the units are obtained through a probability distribution. The hidden units of the RBM are feature detectors which capture the regularities of the input data.

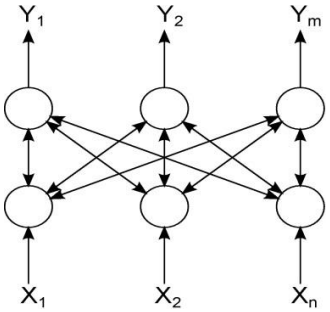

Fig.2 – Restricted Boltzmann Machine

The layers of neural units are connected by bidirectional weights ***W***. The binary units are used very often [2]. The RBM is a stochastic neural network and the states of visible and hidden units are defined using a probabilistic version of the sigmoid activation function:

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

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

The key idea of RBM training is to reproduce as closely as possible the distribution of the input data using the states of the hidden units. 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 [2]. Then the modification of synaptic weights is defined by

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \alpha \frac{\partial LnP(x)}{\partial \omega_{ij}(t)}$$

Using this approach Hinton proposed to use contrastive divergence (CD) technique for RBM learning [2]. It is based on Gibbs sampling. In this case the first term in the 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=n. As a result we can obtain the RBM training rules. In the case of CD-1 the training rule is defined as

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) + \alpha(x_i(0)y_j(0) - x_i(1)y_j(1)) \\ T_i(t+1) &= T_i(t) + \alpha(x_i(0) - x_i(1)) \\ T_j(t+1) &= T_j(t) + \alpha(y_j(0) - y_j(1)). \end{aligned} \qquad (8)$$

If we use CD-k

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) + \alpha(x_i(0)y_j(0) - x_i(k)y_j(k)) \\ T_i(t+1) &= T_i(t) + \alpha(x_i(0) - x_i(k)) \\ T_j(t+1) &= T_j(t) + \alpha(y_j(0) - y_j(k)). \end{aligned} \qquad (9)$$

Here α is the learning rate. As can be seen from these equations, the training rules for RBMs are 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.

Training an RBM is based on presenting a training sample to the visible units, then using the CD-n procedure to compute the binary states of the hidden units, sampling the visible units (reconstructed states), and so on. After performing these iterations the weights and biases of the restricted Boltzmann machine are updated. Then we stack on 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 this unsupervised pre-training we can obtain a good initialization of the neural network. Finally, supervised fine-tuning of the whole neural network is performed.

4. SGD WITH ReLU

As stated earlier the learning using the stochastic gradient descent approach represents the online or mini-batch learning using conventional backpropagation algorithm. In this case the rectified linear unit activation function is used (Fig. 3).

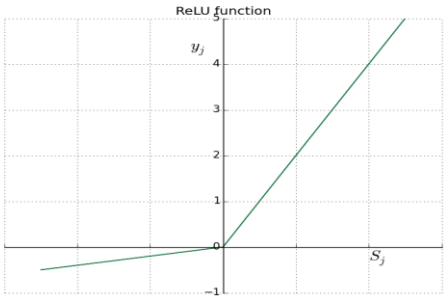

Fig.3 –ReLU Activation Function

In accordance with this function the output of neuron is given by

$$y_j = F(S_j) = \begin{cases} S_j, S_j > 0 \\ kS_j, S_j \le 0 \end{cases}$$

Here k=0 or other small value, for instance k= 0.01 or 0.001.

Use of rectified linear unit (ReLU) transfer function can overcome the restrictions of standard backpropagation approach [9, 11], namely vanishing gradient problem, poor local minima and unstable gradient problem.

5. NEW INSIGHT ON RBM

In this section we propose a novel approach in order to infer RBM training rules. It is based on minimization of reconstruction mean square error, which we can obtain using a simple iterations of Gibbs sampling. In comparison with traditional energy-based method, which is based on linear representation of neural units, the proposed approach permits to take into account nonlinear nature of neural units [8-10].

Let's examine the restricted Boltzmann machine. We will represent the RBM, using three layers (visible, hidden and visible) [8, 9] as shown in Fig. 4.

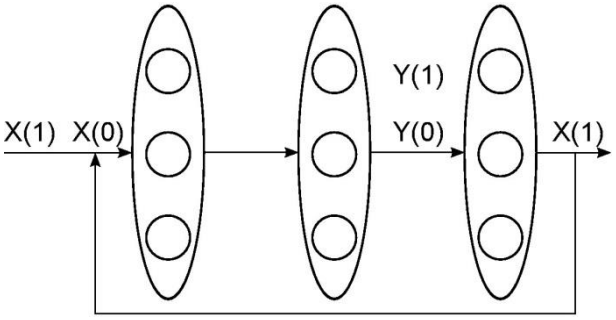

Fig.4 – Unfolded Restricted Boltzmann Machine

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 the training this neural network is to minimize the reconstruction mean squared error (MSE) in a hidden and visible layers. In case of CD-k the reconstruction mean squared error 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

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

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 restricted Boltzmann machine is equivalent to special case of minimizing the reconstruction mean squared error 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 in order to 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 linear RBM.

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

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

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

Thus as can be seen the classical equations for RBM training are particular case of proposed technique.

Theorem 2. Maximization of the log-likelihood input data distribution P(x) in the space of synaptic weights restricted Boltzmann machine is equivalent to minimizing the cross-entropy CE_s error function.

Theorem 3. Maximization of the log-likelihood input data distribution P(x) in the space of synaptic weights restricted Boltzmann machine is equivalent to minimizing the cross-entropy and to minimizing the reconstruction mean squared error using linear neurons in RBM.

$$\max(\ln P(x)) = \min(CE_s) = \min(E_s)$$

Thus the traditional energy-based method is based on maximization of the log-likelihood input data distribution and leads to the linear representation of neural units from the training point of view. The proposed approach is based on minimization of reconstruction mean square error, which we can obtain using simple iterations of Gibbs sampling and leads to the nonlinear representation of neurons.

6. CURRENT LEARNING PARADIGM

If training data set is large then SGD with ReLU is used for deep neural network learning.

Otherwise pre-training and fine-tuning is applied. As stated in [1] "For smaller data sets, unsupervised pre-training helps to prevent overfitting, leading to significantly better generalization when the number of labeled examples is small, or in a transfer setting where we have lots of examples for some 'source' tasks but very few for some 'target' tasks. Once deep learning had been rehabilitated, it turned out that the pre-training stage was only needed for small data sets".

7. CONCLUSION

The traditional energy-based method is based on maximization of the log-likelihood input data distribution and leads to the linear representation of neural units from the training point of view. It is proved that maximization of the log-likelihood input data distribution P(x) in the space of synaptic weights restricted Boltzmann machine is equivalent to minimizing the cross-entropy and to minimizing the reconstruction mean squared error using linear neurons in RBM. The proposed approach is based on minimization of reconstruction mean square error, which we can obtain using simple iterations of Gibbs sampling and leads to the nonlinear representation of neurons. The conventional equations for RBM training are special case of proposed technique when identity transfer function is used.

8. THE FUTURE OF DEEP LEARNING

Finally, I would like to quote an article (LeCun, Y., Bengio, Y. and Hinton, G. E. *Deep Learning Nature*, Vol. 521, pp 436-444. (2015)) about future of deep learning "Although at present the supervised training with ReLU is used mainly for deep neural networks learning, we expect unsupervised learning to become far more important in the longer term. Human and animal learning is largely unsupervised: we discover the structure of the world by observing it, not by being told the name of every object".

9. REFERENCES

[1] LeCun, Y., Bengio, Y. and Hinton, G. E. *Deep Learning Nature*, Vol. 521, pp 436-444. (2015)

[2] Hinton, G. E., Osindero, S., Teh, Y. *A fast learning algorithm for deep belief nets.* Neural Computation, 18, 1527-1554 (2006)

[3] Hinton, G. *Training products of experts by minimizing contrastive divergence.* Neural Computation, 14, 1771-1800 (2002)

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

[5] Hinton, G. E. *A practical guide to training restricted Boltzmann machines*. (Tech. Rep. 2010-000).

Toronto: Machine Learning Group, University of Toronto (2010)

[6] Bengio, Y. *Learning deep architectures for AI. Foundations and Trends in Machine Learning*, 2(1), 1-127 (2009)

[7] Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H. *Greedy layer-wise training of deep networks*. In B. Scholkopf, J. C. Platt, T. Hoffman (Eds.), Advances in neural information processing systems, 11, pp. 153-160. MA: MIT Press, Cambridge (2007)

[8] Golovko, V. A. *Learning Technique for Deep Belief Neural Networks* / V. Golovko, A. Kroshchanka, U. Rubanau, S. Jankowski // in book *Neural Networks and Artificial Intelligence*. – Springer, 2014. – Vol. 440.Communication in Computer and Information Science. – P. 136-146.

[9] Golovko, V. *From multilayer perceptron to deep belief neural networks: training paradigms and application / in book "Lectures on neuroinformatics"*, Moscow, 2015, P. 47-84.

[10] Golovko, Vladimir. *A New Technique for Restricted Boltzmann Machine Learning*/Vladimir Golovko, Aliaksandr Kroshchanka, Volodymyr Turchenko, Stanislaw Jankowski, Douglas Treadwell // Proceedings of the 8th IEEE International Conference IDAACS-2015, Warsaw 24-26 September 2015. – Warsaw, 2015 –P.182-186.

[11] Golovko, V. A. *Nejronie seti: obycenie, organizacija i ptimemenie* / Izdat. Predprijatie Red. Žurnala Radiotechnika**,** 2001, 256.