UDC 004.85(045)
DOI:10.18372/1990-5548.60.13814

[1]V. M. Sineglazov,
[2]O. R. Tofaniuk

AUTOMATED ADJUSTMENT SYSTEM OF RESTRICTED BOLTZMANN MACHINE

[1,2]Aviation Computer-Integrated Complexes Department, National Aviation University, Kyiv, Ukraine
E-mails: [1]svm@nau.edu.ua ORCID 0000-0002-3297-9060, [2]alext98@ukr.net

Abstract—*In this paper the problem of learning the deep believe neural network with help of a restricted Boltzmann machine and the choose of an optimal algorithm for its training is considered. Different algorithms of restricted Boltzmann machine training, which are used for the pre-training of deep believe neural network, are considered, in order to increase the efficiency of this network and further solve the problem of structural-parametric synthesis of deep believe neural network. This task represents the task of justifying the necessity of optimal choice of the restricted Boltzmann machine adjustment algorithm for improving the quality of training of the neural network. To solve this problem, it is suggested to create an automated adjustment system of restricted Boltzmann machine, which choose the optimal training algorithm for this neural network.*

Index Terms—Deep Believe Network; Restricted Boltzmann Machine; Contrastive Divergence; Persistent Contrastive Divergence; Parallel Tempering.

I. INTRODUCTION

Today, artificial neural networks are used not only in the industry, but also in almost all spheres of human life. They come into practice wherever it is necessary to solve the tasks of forecasting, classification or control.

Such impressive success is determined by several reasons:

– rich opportunities: neural networks is an extremely powerful simulation method that allows to reproduce extremely complex dependencies;

– easy to use.

During the period of neural networks existence, (from the middle of the XX century), many types of neural networks were created. In this paper, a deep believe neural network (DBN) with previous training on the basis of a restricted Boltzmann machine (RBM) is considered..

However, despite the high popularity of artificial neural networks, there is an important problem with their use, namely, the choice of optimal training algorithm. In this article, it is considered different learning methods of RBM. It is necessary to admit that the choice of the method of training changes the efficiency of the neural network of deep learning. Therefore, finding an optimal algorithm for training artificial neural network is a very important task of its optimizing.

II. PROBLEM STATEMENT

A deep believe neural network is a multi-layer perceptron with two or more hidden layers of neurons in which neurons in the middle of each layer are not interconnected, but bound between neurons of neighboring layers. Thanks to the multilayered architecture, they allow you to process and analyze a large amount of data, as well as simulate cognitive processes in various areas.

Historically, the first appeared deep believe neural networks and deep perceptron, which in the general case are multilayered perceptron with more than two hidden layers [1]. The main difference between the deep neural network of deep perceptron is that the deep believe neural network in the general case is not a feed forward neural network. Until 2006, in the scientific environment, the paradigm was a priority, that the multilayer perceptron with one, maximum of two hidden layers is more effective for the nonlinear transformation of the input space of images into output compared with the perceptron with a large number of hidden layers. It was considered that it makes no sense to use perceptron with more than two hidden layers. This paradigm was based on the theorem that a perceptron with one hidden layer is a universal approximator [1].

The second aspect of this problem is that all attempts to apply a back propagation algorithm to study a perceptron with three or more hidden layers did not lead to an improvement in the solution of various problems. This is due to the fact that the back propagation algorithm is ineffective for learning perceptrons with three or more hidden layers when used the sigmoid activation function due to the problem of vanishing gradient problem [1].

In article [2] proposed proposed an “greedy layer-wise” algorithm, which became an effective means of learning deep neural networks. It has been shown that the deep neural network has a high efficiency of nonlinear transformation and representation of data in comparison with the traditional perceptron. As a result, the first hidden layer allocates a low-level space of signs of input data, the second layer is a detector of the space of signs of a higher level of abstraction, etc. [3], [1].

As already noted, the deep neural network contains a lot of hidden layers of neuronal elements and implements a deep hierarchical transformation of the input space of images.

For this neural network (Fig. 1), it is used the sample $J=\{(R_j,Y_j)\}$, $j=1,\dots,P$ pairs type of “attribute-value”, where R_j,Y_j is the input and output vectors neural network respectively.

It is necessary to choose an optimal algorithm of RBM training as a criterion it is proposed a generalized error criterion.

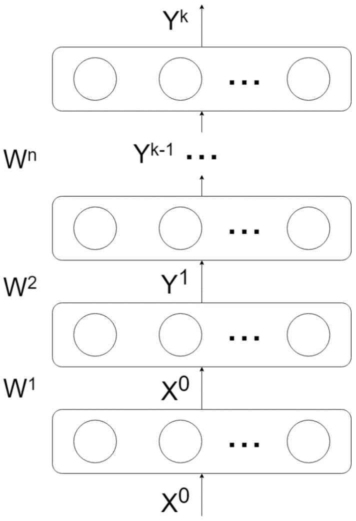

Fig. 1. Deep Believe Network

The output value of the *j*th neuron of the *k*th layer is determined as follows:

$$y_j^k = F\left(S_j^k\right),$$

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

where *F* is the activation function of the neural element; S_j^k is the weighted sum of the *j*th neuron of the *k*-layer; w_{ij}^k is the weighting factor between the *i*th neuron (*k*–1)th layer and the *j*th neuron of the *k*th layer; T_j^k is the threshold value of the *j*th neuron of the *k*th layer.

For the first layer $y_i^0 = x_i$. In the matrix form, the output vector of the *k*-th layer:

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

where *W* is the matrix of weight coefficients; Y^{k-1} is the output vector (*k*–1)th layer; T^k is the vector of the threshold values of the *k*-th layer neurons [1].

The activation function determines the output value of the neuron depending on the result of the weighted sum of the inputs and the threshold value.

The first type of the activation functions is step function. If the value of *Y* is greater than a certain threshold value, we consider the neuron activated. Otherwise, we say that the neuron is inactive. Such a scheme should work, but first let's formalize it:

1. Function *F* = activated if *Y* > bias, otherwise not.

2. Another way: *F* = 1, if *Y* > bias, otherwise *A* = 0.

Such a function is shown in the Fig. 2.

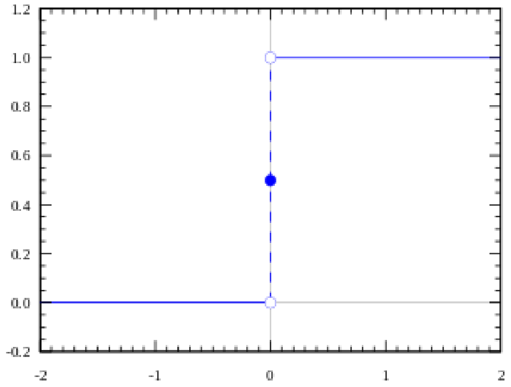

Fig. 2. Step activation function

The second type is linear activation function (Fig. 3).

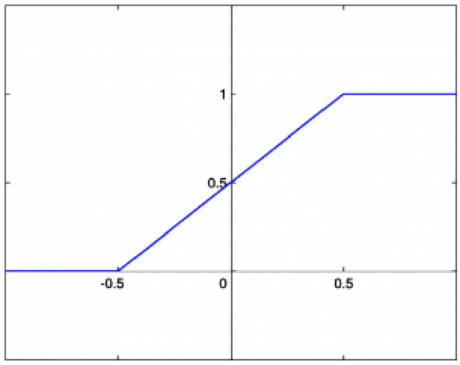

Fig. 3. Linear activation function

The linear function is a straight line and is proportional to the input (that is, the weighted sum of this neuron) $F = cx$.

Next on our list is the activation function ReLu $F(x) = \max(0, x)$.

Using the definition, it becomes clear that ReLu returns the value of *x* if *x* is positive, and 0 otherwise (Fig. 4).

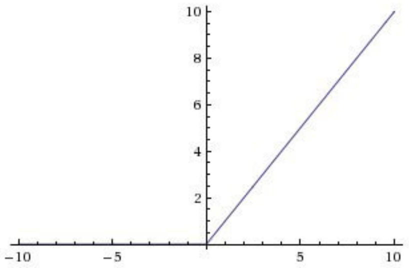

Fig. 4. ReLu activation function

In most cases, the ReLu works as a good approximator.

If a deep neural network is used to classify images, the output values of the network are often determined based on the softmax activation function, and also called sigmoid (Fig. 5).

$$y_j^F = \text{softmax}\left(S_j\right) = \frac{e^{S_j}}{\sum_l e^{S_j}}.$$

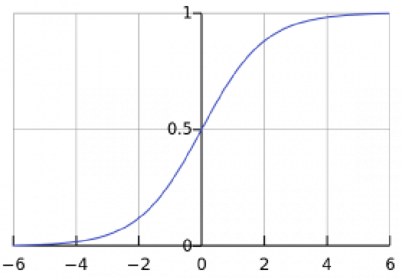

Fig. 5. Softmax activation function

III. Topology of RBM

The restricted Boltzmann machine consists of two layers of stochastic binary neural elements, which are interconnected by bidirectional symmetric connections (Fig. 6). The input layer of the neural elements is called visible (layer *i*), and the second layer is hidden (layer *j*). A deep neural network can be represented as a collection of restricted Boltzmann machines. A restricted Boltzmann machine can approximate (generate) any discrete distribution if sufficient neurons of the hidden layer are used [14].

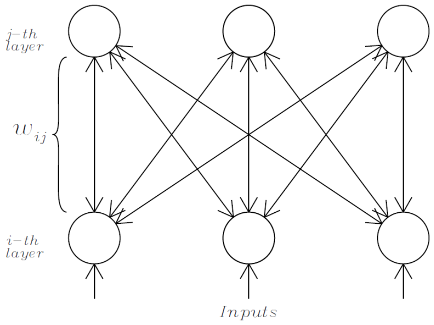

Fig. 6. Restricted Boltzmann Machine

This network is a stochastic neural network in which the states of visible and hidden neurons change in accordance with the probabilistic version of the sigmoid activation function:

$$p\left(y_i|x\right) = \frac{1}{1+e^{S_j}}, \quad S_j = \sum_i^n w_{ij}x_i + T_j. \qquad (1)$$

The states of visible and hidden neural elements are made independent:

$$p\left(x|y\right) = \prod_{i=1}^{n} P\left(x_i|y\right),$$

$$p\left(y|x\right) = \prod_{j=1}^{n} P\left(y_j|x\right).$$

Thus, the states of all the neural elements of a RBM are determined through the probability distribution. In restricted Boltzmann machine, neurons of the hidden layer are feature detectors that determine the patterns of input data. The main task of training is to reproduce the distribution of input data based on the states of the neurons of the hidden layer as accurately as possible. This is equivalent to maximizing the likelihood function by modifying the synaptic connections of the neural network.

Despite the architectural differences of deep neural networks, the principles of their learning are identical. Therefore, we consider the basic concepts of learning such networks on the example of a RBM. For each we will calculate generalized error

$$E = \frac{1}{2}\sum_{i=1}^{k}\left(y - y'\right)^2, \qquad (3)$$

where *k* is the number of elements of sample; *y* is the given output values; y' are expected output values.

IV. Review of Training Algorithms

In this section, algorithms for the training of a RBM will be considered. All common training algorithms for RBMs approximate the log-likelihood gradient given some data and perform gradient ascent on these approximations.

A. Contrastive Divergence

Obtaining unbiased estimates of log-likelihood gradient using Markov Chain Monte Carlo (MCMC) methods typically requires many sampling steps. However, recently it was shown that estimates obtained after running the chain for just a few steps can be sufficient for model training [4]. This leads to *contrastive divergence* (CD) learning, which has become a standard way to train RBMs [2], [4] – [7].

The idea of *k*-step contrastive divergence learning (CD-*k*) is quite simple: instead of approximating the second term in the log-likelihood gradient by a sample from the RBM-distribution (which would require running a Markov chain until the stationary distribution is reached), a Gibbs chain is run for only *k*-steps (and usually $k = 1$). The Gibbs chain is initialized with a training sample $v^{(0)}$ of the training set and yields the sample $v^{(k)}$ after *k* steps. Each step *t* consists of sampling $h^{(t)}$ from $p\left(h|v^{(0)}\right)$ and subsequently sampling $v^{(t+1)}$ from $p\left(v|h^{(t)}\right)$. The gradient, with respect to $\boldsymbol{\theta}$ of the log-likelihood for one training pattern $v^{(0)}$ is then approximated by

$$CD_k\left(\theta, v^{(0)}\right) = -\sum_h p\left(h|v^{(0)}\right)\frac{\partial E\left(v^{(0)},h\right)}{\partial\theta} + \sum_h p\left(h|v^{(k)}\right)\frac{\partial E\left(v^{(k)},h\right)}{\partial\theta}, \quad (4)$$

This algorithm was invented by Professor Hinton in 2002, and is distinguished by its simplicity. The main idea is that mathematical expectations are replaced by quite definite values. This approximation is based on Gibbs sampling. The CD-*k* process looks like this (Fig. 7):

– the state of visible neurons equates to the input image;

– the probabilities of states of the hidden layer are displayed;

– for each neuron of the latent layer, the condition "1" is brought in with the probability equal to its current state;

– the probabilities of the visible layer are derived based on the latent;

– if the current iteration is less than *k*, then return to step 2;

– the probabilities of states of the hidden layer are displayed.

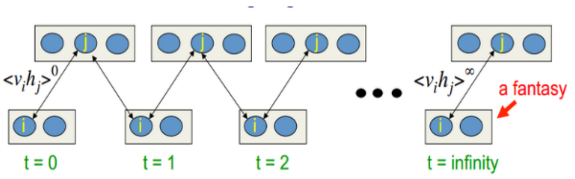

Fig. 7. Schematic application of CD-*k*

The longer do sampling, the more accurate our gradient will be. At the same time, the professor asserts that even for CD-1 (only one iteration of sampling) a quite good result is obtained. The first term is called the positive phase, and the second with the minus sign is called the negative phase.

In the Gibbs sampling, the first terms in the expressions for the gradient characterize the data distribution at time $t = 0$, and the second is the reconstructed or generated state of the state at time $t = k$. Proceeding from this, the CD-*k* procedure can be represented as follows:

$$x(0)\to y(0)\to x(1)\to y(1)\to\ldots\to x(k)\to y(k).$$

As a result, can got the following rules for training the RBM network. In the case of the use of CD-1 $k = 1$ and taking into account that in accordance with the method of gradient descent $w_{ij}(t+1) = w_{ij}(t) + \alpha\frac{\partial InP(x)}{\partial w_{ij}(t)}$, for consistent training have:

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

Similarly, for the CD-*k* algorithm:

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

In the case of group learning and CD-*k*

$$w_{ij}(t+1) = w_{ij}(t) + \alpha\sum_{l=1}^{L}\left(x_i^l(0)y_j^l(0) - x_i^l(k)y_j^l(k)\right),$$
$$T_i(t+1) = T_i(t) + \alpha\sum_{l=1}^{L}\left(y_j^l(0) - y_j^l(k)\right),$$
$$T_j(t+1) = T_j(t) + \alpha\sum_{l=1}^{L}\left(x_i^l(0) - x_i^l(k)\right).$$

From the last expressions it can be seen that the rules of training a restricted Boltzmann machine minimize the difference between the original data and the results generated by the model. The values generated by the model are obtained by Gibbs sampling.

B. Persistent Contrastive Divergence

The CD-1 is fast, has a low dispersion and is a reasonable approximation to the likelihood gradient, but it is still significantly different from the probability gradient when the mixing speed is low. Generally speaking, CD-*k* for greater n is better than CD-1 if there is enough time to work [8].

Although CD-1 is not a very good approximation to maximum likelihood learning, this does not seem to matter when an RBM is being learned in order to provide hidden features for training a higher-level RBM. CD-1 ensures that the hidden features retain most of the information in the data vector and it is not necessarily a good idea to use a form of CD that is a closer approximation to maximum likelihood but is worse at retaining the information in the data vector. If, however, the aim is to learn an RBM that is a good density or joint-density model, CD-1 is far from optimal.

At the beginning of learning, the weights are small and mixing is fast so CD-1 provides a good approximation to maximum likelihood. As the weights grow, the mixing gets worse and it makes sense to gradually increase the *n* in CD-*k*. When n is increased, the difference of pairwise statistics that is used for learning will increase so it may be necessary to reduce the learning rate.

Method, called Persistent Contrastive Divergence (PCD) solves the sampling with a related method, only that the negative particle is not sampled from the positive particle, but rather from the negative particle from the last data point [8].

The idea of persistent contrastive divergence (PCD) [8] is described in [9] for log-likelihood maximization of general MRFs (Markov Random Fields) and is applied to RBMs in [8]. The persistent contrastive divergence approximation is obtained from the CD approximation (4) by replacing the sample $v^{(k)}$ by a sample from a Gibbs chain that is independent of the sample $v^{(0)}$ of the training distribution. The algorithm corresponds to standard CD learning without reinitializing the visible units of the Markov chain with a training sample each time we want to draw a sample $v^{(k)}$ approximately from the RBM distribution. Instead one keeps "persistent" chains which are run for *k* Gibbs steps after each parameter update (i.e., the initial state of the current Gibbs chain is equal to $v^{(k)}$ from the previous update step). The fundamental idea underlying PCD is that one could assume that the chains stay close to the stationary distribution if the learning rate is sufficiently small and thus the model changes only slightly between parameter updates [8], [9]. The number of persistent chains used for sampling (or the number of samples used to approximate the second term of gradient) is a hyper parameter of the algorithm. In the canonical form, there exists one Markov chain per training example in a batch.

The persistent contrastive divergence algorithm was further refined in a variant called fast persistent contrastive divergence (FPCD) [10]. Fast PCD tries to reach a faster mixing of the Gibbs chain by introducing additional parameters w_{ij}^f, b_j^f, c_j^f (for $i = 1, \dots, n$ and $j = 1, \dots, m$) referred to as the fast parameters. This new set of parameters is only used for sampling and not in the model itself. When calculating the conditional distributions for Gibbs sampling, the regular parameters are replaced by the sum of the regular and the fast parameters, i.e., Gibbs sampling is based on the probabilities

$$\tilde{p}\left(H_i = 1 \middle| v\right) = \text{sig}\left(\sum_{j=1}^{m}\left(w_{ij} + w_{ij}^f\right)v_j + \left(c_i + c_i^f\right)\right), \quad (5)$$

and

$$\tilde{p}\left(V_j = 1 h\right) = \text{sig}\left(\sum_{i=1}^{m}\left(w_{ij} + w_{ij}^f\right)h_i + \left(b_j + b_j^f\right)\right), \quad (6)$$

instead of the conditional probabilities given by (1) and (2). The learning update rule for the fast parameters is the same as the one for the regular parameters, but with an independent, large learning rate leading to faster changes as well as a large weight decay parameter. Weight decay can also be used for the regular parameters, but it has been suggested that regularizing just the fast weights is sufficient [10]. Neither PCD nor FPCD seem to increase the mixing rate (or decrease the bias of the approximation) sufficiently to avoid the divergence problem, as can be seen in the empirical analysis in [11].

C. Parallel Tempering

However, the study of contrast divergence is considered an effective way to study RBMs, it has a drawback due to the biased approach in the learning curve. This chapter proposes using the advanced Monte Carlo method, which is called Parallel Tempering (PT), and experimentally shows that it works effectively [12].

The problem that was not solved by either the Gibbs sampling or the CD training is that the samples formed during the negative phase are not inclined to explain the whole state of the state. Thus, this section proposes to use another improved version of the Monte Carlo Markov Chain sampling method, called Parallel Tempering (PT).

The introduction of the PT sample occurs in the 1980s, when Swendsen and Wang introduced the Monte Carlo replica and applied it to the Ising model [15], which is equivalent to a Boltzmann machine with visible neurons. Simulation of the replica Monte Carlo suggested modeling several copies of particles (replicas) at different temperatures simultaneously, rather than simulating them consistently. Similarly, Geyer later introduced the use of a parallel MCMC sampling chain based on the mixing rate of samples through parallel chains to maximize the likelihood [16].

The basic idea of PT sampling is that samples are collected from multiple chains of Gibbs sampling with different temperatures. The term temperature in this context means the energy level of the overall system. The higher the temperature of the chain, the more likely it is that samples collected with Gibbs move freely.

For each pair of samples collected from two different chains, the probability of swapping is calculated, and the samples vary places according to the probability. The probability of a swap for a pair of samples is formulated in accordance with the Metropolis rule [13]

$$P_{\text{swap}}\left(x_{T_1}, x_{T_2}\right) = \min\left(1, \frac{P_{T_1}\left(x_{T_2}\right)P_{T_1}\left(x_{T_1}\right)}{P_{T_1}\left(x_{T_1}\right)P_{T_2}\left(x_{T_2}\right)}\right),$$

where T_1 and T_2 denote the temperatures of the two chains, and x_{T_1} and x_{T_2} denote samples collected from two chains.

After each round of sampling and swapping, the sample at the true temperature $T = 1$ is gathered as the sample for the iteration. The samples come from the true distribution, $p(v,h|\theta)$ in case of RBMs, assuming that enough iterations are run to diminish the effect of the initialization.

It must be noted that the Gibbs sampling chain with the highest temperature ($T = 0$) is never multi-modal such that all the neurons are mutually independent and likely to be active with probability 1/2. So, the samples from the chain are less prone to missing some modes. From the chain with the highest temperature to the lowest temperature, samples from each chain become more and more likely to follow the target model distribution. How PT sampling could being trapped into a single mode is illustrated in Fig. 8.

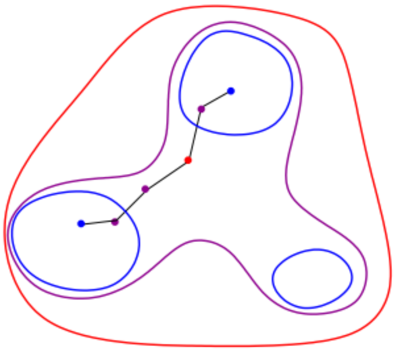

Fig. 8. Illustration of how PT sampling could avoid being trapped in a single mode. The red, purple, and blue curves and dots indicate distributions and the samples from the distributions with the high, medium, and cold temperatures, respectively. Each black line indicates a single sampling step

This nature of swapping samples between the different temperatures enables better mixing of samples from different modes with much less number of samples than that would have been required if Gibbs sampling was used.

Parallel tempering sampling in training RBMs can be simply uses as a replacement of Gibbs sampling in the negative phase. This method is, from now on, referred to as PT learning. Due to the previously mentioned characteristics, it is expected that the samples collected during the negative phase would explain the model distribution better, and that the learning process would be successful even with a smaller number of samples than those required if Gibbs sampling is used.

V. Problem Solution

As known, the main task of this work is finding an optimal algorithm for configuring a restricted Boltzmann machine. For a comparative example, three learning algorithms were used: Contrastive Divergence (CD), Persistent Contrastive Divergence (PCD), Parallel Tempering (PT).

This section is intended to describe a general algorithm that, based on criteria such as learning speed and accuracy of the neural network, determines the most productive algorithm of RBM training.

The block diagram of how to work an automated adjustment system of restricted Boltzmann machine is shown in Fig. 9.

On input of each deep believe neural network, which consists of a RBM stack and a multilayer perceptron, is served with a training sample. These three deep believe neural networks are pre-trained using RBMs with different learning algorithms, namely Contrastive Divergence (CD), Persistent Contrastive Divergence (PCD), Parallel Tempering (PT).

Then the results of the outputs from the each multilayer perceptron are recorded in the database and also for these outputs, the generalized error criterion is computed. The results of these calculations arrive at the block of determination minimum error, where among them the result is chosen with the minimum value of error.

After these calculations, a block of choice optimal algorithm is included into the work, where the results are from three deep believe neural networks. Based on results from the block of determination minimum error, the automatic adjustment system decides which algorithm is optimal and then outputs the result of the deep believe neural network with this algorithm.

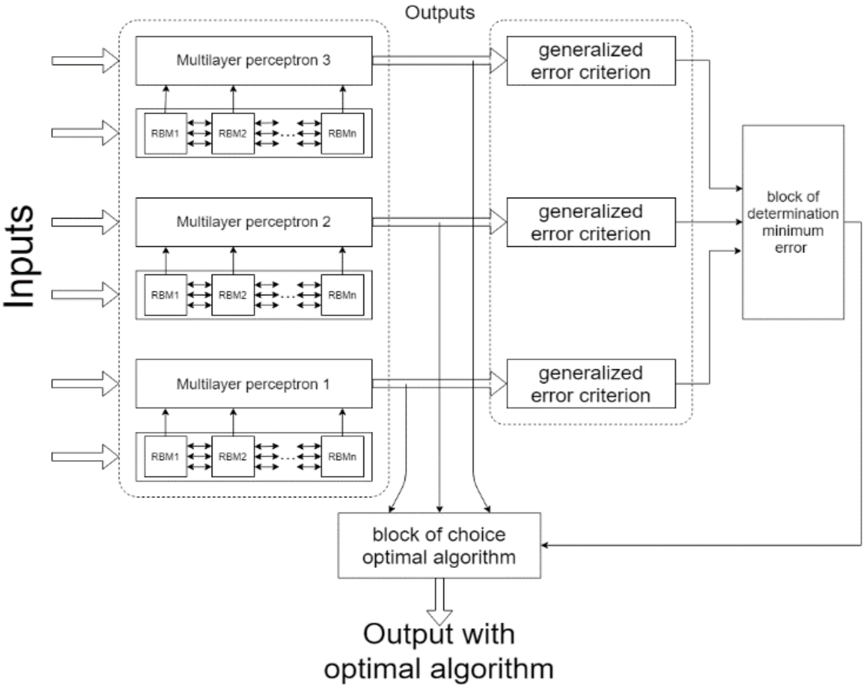
Outputs
Inputs
Multilayer perceptron 3
Multilayer perceptron 2
Multilayer perceptron 1
RBM1
RBM2
RBMn
generalized error criterion
block of determination minimum error
block of choice optimal algorithm
Output with optimal algorithm

Fig. 9. Structural scheme of automated adjustment system of restricted Boltzmann machine

Thus, the adjustment algorithm of deep believe neural network has the following form:

1. Restricted Boltzmann machine is adjusted according to the first, second and third algorithms one by one, for this training sample. As a result, receive the first, second and third sets of weight coefficients W_1, W_2, W_3.
2. By a result of this training, the main neural network – perceptron is initialized in sequence, which involves the determination of the initial values of RBM weights coefficients for the first, second and third cases.
3. As a result, using of the back propagation method, the values of weight coefficients are found according to the first, second and third algorithms of the training restricred Boltzmann machine.
4. The value of a generalized error criterion is found on the test sample for the first, second and third cases.
5. Based on the obtained values of the generalized error criterion, the optimal adjustment algorithm of restricted Boltzmann machine is chosen.

This is a simple by structure and logic structural scheme of an automated system for selecting an optimal algorithm for training a restricted Boltzmann machine, but also an extremely capacious one. Because we use up to three deep believe networks (RBM and MLP) and in addition, we create additional blocks for calculating the minimum errors of each algorithm and comparative blocks.

As an example, are considered the deep believe neural network (Fig. 10) with the three hidden layers of the perceptron, for which corresponds three Boltzmann machines connected sequentially.

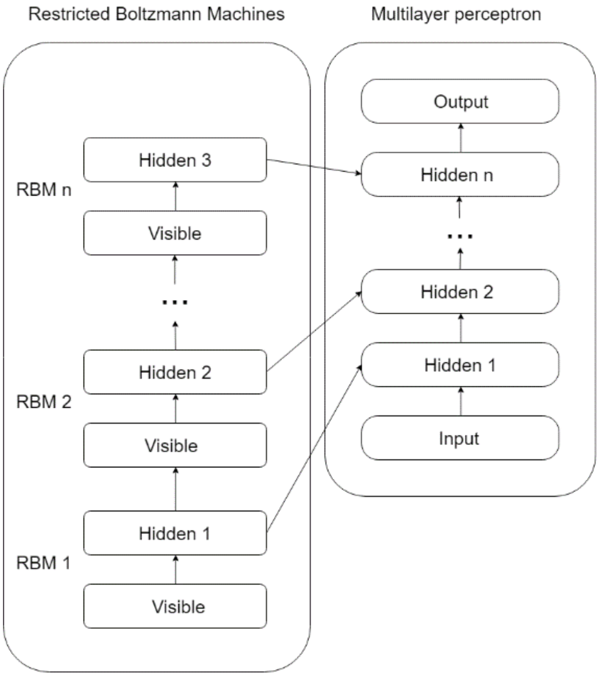

Fig. 10. Deep believe neural network

The training sample contained 50 examples, represented as vectors of 250 × 250 dimension and coordinate values on the segment (−1; 1). The test sample included an additional 10 examples of the same dimension. Results of modeling is represented in Tables I, II, and III.

TABLE I. RESULTS OF CD MODELING

Initialized weight coefficients of RBM	Initialized weight coefficients of perceptron	Generalized error criterion
[0.258; 0.650; 0.845; 0.254; 0.873]	[0.302; 0.570; 0.900; 0.258; 0.950]	0.0225

TABLE II. RESULTS OF PCD MODELING

Initialized weight coefficients of RBM	Initialized weight coefficients of perceptron	Generalized error criterion
[0.565; 0.841; 0.785; 0.255; 0.720]	[0.570; 0.850; 0.790; 0.258; 0.710]	0.0178

TABLE III. RESULTS OF PT MODELING

Initialized weight coefficients of RBM	Initialized weight coefficients of perceptron	Generalized error criterion
[0.356; 0.745; 0.552; 0.450; 0.854]	[0.300; 0.654; 0.720; 0.258; 0.957]	0.0365

According to results test sample of generalized error for three algorithms, the optimal algorithm for RBM training is a Persistent Contrastive Divergence. Because it has smaller value of generalized error than other two algorithms.

VI. CONCLUSION

Creating such an automated adjustment system of restricted Boltzmann machine allows to choose the optimal algorithm for its training, based on performance generalized error criterion of the neural network. Using this approach will increase the efficiency of solving tasks with the help of deep believe neural network and extend its use in the applied field.

REFERENCES

[1] V. Golovko, "From multilayer perceptron to deep belief neural networks: training paradigms and application," in book *Lectures on neuroinformatics,* Moscow, 2015, pp. 47–84. (in Russian)

[2] G. E. Hinton and S. Osindero, "A fast learning algorithm for deep belief nets," *Neural computation № 18*, 2006, pp. 1527–1554.

[3] G. E. Hinton, R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, no. 313 (5786), 2006, pp. 504–507.

[4] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," Neural Computation, pp. 1771–1800, 2002.

[5] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, and U. Montreal, "Greedy layerwise training of deep networks," in: B. SchËolkopf, J. Platt, T. Hoffman (eds.) *Advances in Neural Information Processing*, pp. 153–160, MIT Press, 2007.

[6] Y. Bengio and O. Delalleau, "Justifying and generalizing contrastive divergence", *Neural Computation* 21(6), pp. 1601–1621, 2009.

[7] G. E. Hinton, "Learning multiple layers of representation," *Trends in Cognitive Sciences*, 11(10), 428–434, 2007.

[8] T. Tieleman, "Training restricted Boltzmann machines using approximations to the likelihood gradient," in *International Conference on Machine learning (ICML)*, pp. 1064–1071, 2008.

[9] L. Younes, "Maximum likelihood estimation of Gibbs fields," in *Joint Conference on Spacial Statistics and Imaging, Lecture Notes Monograph Series*, Institute of Mathematical Statistics, Hayward, California, 1991.

[10] T. Tieleman and G. E. Hinton, "Using fast weights to improve persistent contrastive divergence," *International Conference on Machine Learning (ICML)*, pp. 1033–1040, 2009.

[11] A. Fischer and C. Igel, "Empirical analysis of the divergence of Gibbs sampling based learning algorithms for Restricted Boltzmann Machines," in *International Conference on Artificial Neural Networks (ICANN)*, vol. 6354 of LNCS, pp. 208–217, Springer-Verlag, 2010.

[12] K. Cho, T. Raiko and A. Ilin, "Parallel tempering is efficient for learning restricted Boltzmann machines," *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2010)*, pp. 3246–3253. IEEE Press, 2010.

[13] D. J. C. Mackay, *Information Theory, Inference & Learning Algorithms*, 1st ed. Cambridge University Press, June 2002.

[14] Bengio Y., "Learning deep architectures for AI," *Foundations and trends in machine learning*, no. 2(1), pp. 1–127, 2009.

[15] H. Robert, Swendsen and Jian-Sheng Wang, “Replica Monte Carlo Simulation of Spin-Glasses”, Lett. 57, 2607, 24 November 1986.

[16] Geyer and J. Charles, *Markov Chain Monte Carlo Maximum Likelihood*, 1991.

Received May 06, 2019

Victor Sineglazov. orcid.org/0000-0002-3297-9060
Doctor of Engineering Science. Professor. Head of the Department.
Aviation Computer-Integrated Complexes Department, Faculty of aeronavigation, electronics and telecomunication, National Aviation University, Kyiv, Ukraine.
Education: Kyiv Polytechnic Institute, Kyiv, Ukraine, (1973).
Research area: Air Navigation, Air Traffic Control, Identification of Complex Systems, Wind/Solar power plant.
Publications: more than 600 papers.
E-mail: svm@nau.edu.ua

Oleksii Tofaniuk. Bachelor.
Aviation Computer-Integrated Complexes Department, Faculty of aeronavigation, electronics and telecomunication, National Aviation University, Kyiv, Ukraine.
Research area: Artificial neural networks.
Publications: 1.
E-mail: alext98@ukr.net

В. М. Синєглазов, О. Р. Тофанюк. Автоматизована система налаштування обмеженої машини Больцмана

У даній роботі розглянуто задачу навчання нейронної мережі глибокої довіри за допомогою обмеженої машини Больцмана та вибором оптимального критерію для її навчання. Розглянуто різні алгоритми навчання обмеженої машини Больцмана, яка використовується для попереднього навчання нейронної мережі глибокої довіри, для підвищення ефективності роботи цієї мережі з подальшим вирішенням задачі структурно-параметричного синтезу нейронної мережі глибокої довіри. Це завдання являє собою задачу обгрунтування необхідності оптимального вибору алгоритму налаштування обмеженої машини Больцмана для підвищення якості навчання нейронної мережі глибокої довіри. Для вирішення цієї проблеми запропоновано створити автоматизовану систему налаштування нейронної мережі глибокої довіри, яка буде обирати оптимальні критерії навчання для даної нейронної мережі.

Ключові слова: нейронна мережа глибокої довіри; обмежена машина Больцмана; контрастна розбіжність; стійка контрастна розбіжність; паралельне загартування.

Синєглазов Віктор Михайлович. orcid.org/0000-0002-3297-9060
Доктор технічних наук. Професор. Завідувач кафедрою.
Кафедра авіаційних комп’ютерно-інтегрованих комплексів, факультет аеронавігації, електроніки та телекомунікацій, Національний авіаційний університет, Київ, Україна.
Освіта: Київський політехнічний інститут, Київ, Україна, (1973).
Напрям наукової діяльності: аеронавігація, управління повітряним рухом, ідентифікація складних систем, вітроенергетичні установки.
Кількість публікацій: більше 600 наукових робіт.
E-mail: svm@nau.edu.ua

Тофанюк Олексій Русланович. Бакалавр.
Кафедра авіаційних комп’ютерно-інтегрованих комплексів, факультет аеронавігації, електроніки та телекомунікацій, Національний авіаційний університет, Київ, Україна.
Напрям наукової діяльності: штучні нейронні мережі.
Кількість публікацій: 1.
E-mail: alext98@ukr.net

В. М. Синеглазов, А. Р. Тофанюк. Автоматизированная система настройки ограниченной машины Больцмана

В данной работе рассмотрена задача обучения нейронной сети глубокого доверия с помощью ограниченной машины Больцмана и выбором оптимального критерия для ее обучения. Рассмотрены различные алгоритмы обучения ограниченной машины Больцмана, которая используется для предварительного обучения нейронной сети глубокого доверия, для повышения эффективности работы сети с последующим решением задачи структурно-параметрического синтеза нейронной сети глубокого доверия. Эта задача представляет собой задачу обоснования необходимости оптимального выбора алгоритма настройки ограниченной машины Больцмана для повышения качества обучения нейронной сети глубокого доверия. Для решения этой проблемы предложено создать автоматизированную систему настройки нейронной сети глубокого доверия, которая будет выбирать оптимальные критерии обучения для данной нейронной сети.

Ключевые слова: нейронная сеть глубокого доверия; ограничена машина Больцмана; контрастное расхождение; устойчивое контрастное расхождение; параллельная закалка.

Синеглазов Виктор Михайлович. orcid.org/0000-0002-3297-9060
Доктор технических наук. Профессор. Заведующий кафедрой.
Кафедра авиационных компьютерно-интегрированных комплексов, факультет аэронавигации, электроники и телекомуникаций, Национальный авиационный университет, Киев, Украина.
Образование: Киевский политехнический институт, Киев, Украина, (1973).
Направление научной деятельности: аэронавигация, управление воздушным движением, идентификация сложных систем, ветроэнергетические установки.
Количество публикаций: более 600 научных работ.
E-mail: svm@nau.edu.ua

Тофанюк Алексей Русланович. Бакалавр.
Кафедра авиационных компьютерно-интегрированных комплексов, факультет аэронавигации, электроники и телекомуникаций, Национальный авиационный университет, Киев, Украина.
Направление научной деятельности: искусственные нейронные сети.
Количество публикаций: 1.
E-mail: alext98@ukr.net