

Original Research Article

**Two Hidden Layer Multiconnected Boltzmann Machine
for Controlling Moving Objects**

Abstract There are various areas of industry where the remote control of machines significantly increases productivity. In terms of household remote control is extremely important for elderly and disabled people. The most attractive remote control is the voice one. Moving objects can be successfully controlled by means of voice commands. The present paper deals with a remote control of moving objects. Voice control is used for managing robots, drones, wheelchairs etc. For this purpose a new artificial neural network is obtained. A successful classification by a new two hidden layer Boltzmann machine is realized. The process of deep machine learning is studied with respect to the mean field approximation problem. An algorithm for training the classifier is written explicitly in pseudocode. Real life tests are discussed.

Key words – voice control, signal processing, deep Boltzmann machine, mean field approximation.

Nomenclature

$\bar{v}$	2-valent cepstral coefficient tensor of all frames representing a fixed word
$\underline{v}$	cepstral coefficient vector of a fixed frame
v	cepstral coefficient
$\bar{h}^{(j)}$	2-valent binary tensor of hidden units
$\underline{h}_i^{(j)}$	hidden unit (binary vector)
$\underline{u}$	vector of binary labels
P	distribution
$\langle \cdot \rangle_P$	expectation over the distribution P
p	likelihood
$\mathbf{H}$	3-valent binary tensor containing both hidden layers
m	number of visible units (frames)

s_k	number of units in the k -th hidden layer
n	number of the words in the basis
t	number of training pairs
σ_i	standard deviation of the input unit $\underline{v}_i$
$w_{ij}, \tilde{w}_{ij}$	weight matrices
$W, \tilde{W}$	4-valent real-valued weight tensors or corresponding rectangular block weight matrices
$\underline{U}_{ij}^{(k)}$	weight vectors
$(\bar{v}, \underline{u})$	a training pair
$\mathbf{V}$	set of training pairs
$\bar{a}$	offsets
$\bar{b}, \bar{c}^{(j)}, \underline{d}$	biases
Θ	set of all biases, offsets and weights of the neural network
sigm	logistic function
Sigm	multivariate sigmoid
$\mathcal{N}$	multivariate Gaussian
λ	learning rate

1 Introduction

The remote control is one of the most contemporary approach to the impact of moving objects. It is successfully combined with voice commands in the last decade. There are a lot of applications of the voice control in the real life, for examples controlling of robots, drones, wheelchairs etc. The number of home systems with voice control is growing continuously. The voice banking will become a hit in the not too distant future. We should also mention navigating systems, voice identification systems and telecommunications. The major advantage of voice control systems is the hands-free communication with the managed object. The contemporary voice control is related to the probabilistic theory of deep machine learning. Every voice control system needs a speech recognition module. Generally speaking an automatic speech recognition tool consists of a voice activity detector well known as VAD filter, signal preprocessing techniques and classifier. There are various classification methods. The wavelet method, the dynamic time

warping and various neural networks are successfully used for classification in the last decades. The efficiency of the speech recognition systems strongly depends on the choice of classifier.

The first paper on Deep Boltzmann Machine (DBM) was published way back in 2009. There are just a few researchers who concentrated on DBM. Results for multilayer Boltzmann machines can be found in [1, 2, 3, 4, 5]. The term DBM was introduced for the very first time by R. Salakhutdinov, G. Hinton [2] in 2009. They obtained a DBM with two hidden and one visible layers of binary units. G. Montavon and Klaus-Robert Müller [3] consider a simplified DBM with binary visible and hidden units. The authors present the energy function of the DBM as a function of centered states. They conclude that the centering procedure essentially facilitates the training of DBM. The idea with centering trick is extended by N. Wang, D. Jancke, L. Wiskott [6] for Gaussian-binary deep Boltzmann machines. R. Salakhutdinov, H. Larochelle [7] develop a three layer DBM introducing very effective inference procedure. K. Cho, T. Raiko, A. Ilin [1] consider DBN with n hidden layers on the basis of the enhanced gradient and parallel tempering.

We use an improved version of the stochastic model of voice control system obtained by R. T. Ilarionov, C. Jayne, T. D. Todorov, G. S. Tsanev, S. Y. Yordanov [8] with a new original classifier. The new classifier is a two hidden layer multiconnected classification Boltzmann machine.

Further, the paper is organized as follows. A principle flowchart of voice control of a moving object is described in Section 2. A new artificial neural network is obtained in Section 3. Algorithmic aspects of the two hidden layer multiconnected Boltzmann machine is analyzed in the same section. Various numerical experiments can be seen in Section 4. Section 5 contains some concluding remarks.

2 Stochastic approach for controlling a moving object

A detail scheme of the voice control of a moving object is presented on Figure 1. The user says a voice command. The Signal Preprocessing includes Herz to mel conversion of the input signal and feature extraction. So many unpredictable noises can be found in a real environment. That is why we need a Wiener filter for a noise

reduction. The voice activity detection filter indicates the voice part of the signal.

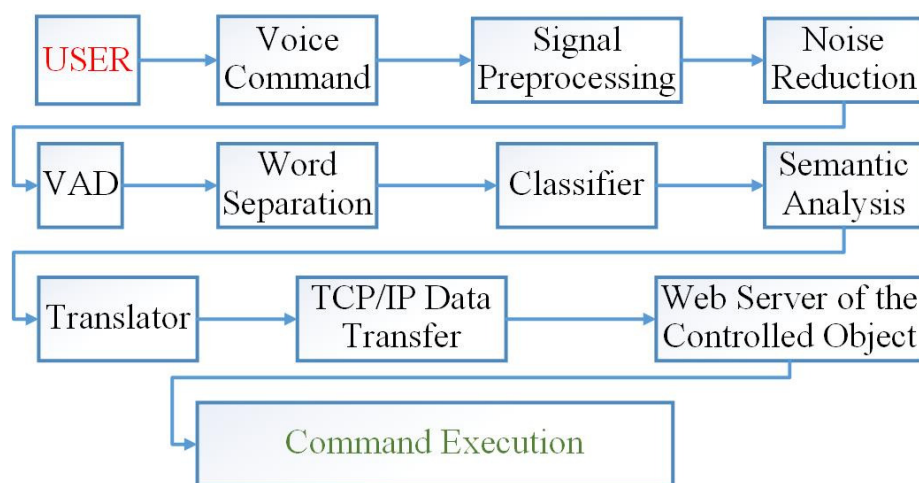


Figure 1 Principal scheme of voice control of a moving object

After that the input command is separated into its constituent words. The role of the classifier is to recognize the input command. The deep machine learning is related to this part of the signal processing. There are plenty of stochastic classifiers. We concentrate on two hidden layer multiconnected Boltzmann machine. A semantic analysis is used to improve the recognition, for more details see [8]. After specifying the command it is translated into the language of the moving object. Drones, robots and spy tanks are controlled by programming languages similar to BASIC. For instance EZ-Builder became very popular for managing robots. The translated command is sent to the web server of the controlled object. It follows an execution of the command.

3 Multiconnected deep Boltzmann machine

We assume that all hidden layers are connected excluding horizontal connections within a fixed layer. Boltzmann machines with horizontal connections within a fixed layer are said to be general Boltzmann machines [2]. The DBMs investigated in [2, 3, 6, 7, 1] are applied for image recognition. All considered models have a common feature. They have connections only between adjacent layers. We introduce completely different approach. The proposed multiconnected deep Boltzmann machine has four layers, Figure 2. The visible layer $\bar{v}$

consists of real-valued vectors of cepstral coefficients and it is connected with the top hidden layer $\bar{h}^{(1)}$. The hidden layers $\bar{h}^{(1)}$ and $\bar{h}^{(2)}$ consist of binary vectors and both layers are connected each other. Additionally we have a layer $\underline{u}$ with labels. The output layer $\underline{u}$ consists of binary units and it is connected with both hidden layers. Actually the proposed model is a natural generalization of the classification RBM obtained by R. T. Ilarionov, C. Jayne, T. D. Todorov, G. S. Tsanev, S. Y. Yordanov [8].

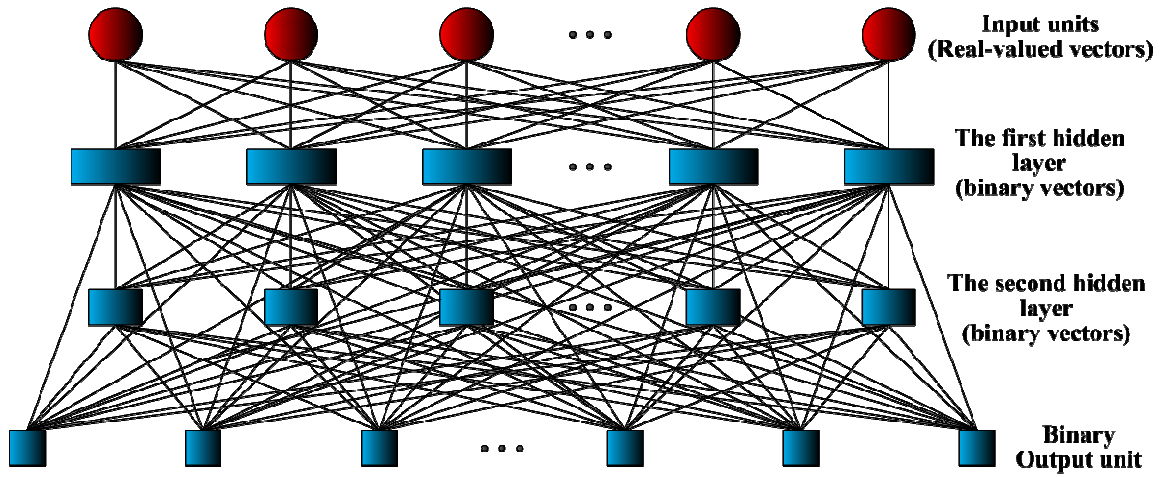


Figure 2 The multiconnected deep Boltzmann machine.

The energy of the model

$$\begin{aligned}
 E(\bar{v}, \mathbf{H}, \underline{u}) = & \frac{1}{2} \sum_{i=1}^m \frac{(\underline{v}_i - \underline{a}_i)^2}{\sigma_i^2} - \sum_{i=1}^m \underline{b}_i \underline{v}_i - \sum_{j=1}^2 \sum_{i=1}^{s_j} \underline{c}_i^{(j)} \underline{h}_i^{(j)} \\
 & - \sum_{i=1}^m \sum_{j=1}^{s_1} \frac{\underline{v}_i^T \underline{w}_{ij} \underline{h}_j^{(1)}}{\sigma_i} - \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} (\underline{h}_i^{(1)})^T \tilde{w}_{ij} \underline{h}_j^{(2)} - \sum_{i=1}^n d_i u_i \\
 & - \sum_{k=1}^2 \sum_{i=1}^{s_k} \sum_{j=1}^n (\underline{h}_i^{(k)})^T \underline{U}_{ij}^{(k)} u_j
 \end{aligned}$$

generates the following joint distributions:

$$P(\bar{v}, \mathbf{H}, \underline{u}) = \frac{e^{-E(\bar{v}, \mathbf{H}, \underline{u})}}{\sum_{\bar{v}, \mathbf{H}, \underline{u}} e^{-E(\bar{v}, \mathbf{H}, \underline{u})}}, \quad P(\bar{v}, \underline{u}) = \frac{\sum_{\mathbf{H}} e^{-E(\bar{v}, \mathbf{H}, \underline{u})}}{\sum_{\bar{v}, \mathbf{H}, \underline{u}} e^{-E(\bar{v}, \mathbf{H}, \underline{u})}}.$$

Let

$$\mathbf{V} = \{(\bar{v}_i, \underline{u}_i) | i = 1, 2, \dots, t\},$$

be a set of training pairs. A supervised learning is applied in the present investigation. Each training word $\bar{v}_i$ is related to a binary label $\underline{u}_i$. Our purpose is to maximize the logarithmic likelihood

$$L(\theta) = -\sum_{i=1}^t \log p(\bar{v}_i, \underline{u}_i), \theta \in \Theta.$$

The contrastive divergence method for training Boltzmann machines attains significant popularity after Hinton's paper [9] had been published way back in 2002. We apply this method with mean field approximation procedures [1] to obtain the probability of each hidden neuron to be active. Our algorithm for training the two hidden layer multiconnected Boltzmann machine is presented in pseudo code to obtain a clear exposition of our idea. Before starting we center the offsets with respect to the training data. All elements of the visible layer belong to the interval [0,1] as it is done in [8]. The output units satisfy

$$\sum_{i=1}^n u_i = 1.$$

Algorithm 1

```
% Contrastive divergence algorithm for training
the
% classification multiconnected Boltzmann
machine.
% -----
----
% Initialization
create offsets
center offsets
% The initial approximations for all of the
visible and % output units are set randomly.
create  $\bar{h}_i^{[0]}, i = 1, 2$  randomly
create  $\underline{u}^{[0]}, \bar{v}^{[0]}$  randomly
```

```

create all elements of biases and offsets
randomly in
    [0,1]
create normalized and positive definite weight
matrices
create normalized weight vectors
% Positive phase
% -----
----

forall  $(\bar{v}, \underline{u}) \in V$  do
% The index  $i$  is the current number of the
training pair.
    begin
        % The mean field approximation
        % The value  $K$  indicates the number of
iterations.
        for  $k = 0$  to  $K - 1$  do
            begin
                % The first hidden layer
                for  $j = 1$  to  $s_1$  do
                    % The index  $j$  shows the number of the
current
                    % hidden unit in the first hidden
layer.
                        begin

$$\underline{h}_{1j}^{[k+1]} := \text{Sigm} \left( \underline{c}_j^{(1)} + \sum_{i=1}^m \frac{v_i^T w_{ij}}{\sigma_i} + \sum_{i=1}^{s_2} \tilde{w}_{ij} \underline{h}_{2i}^{[k]} + \sum_{i=1}^n \underline{U}_{ji}^{(1)} u_i \right)$$

                        end  $j$ 
                    % The second hidden layer
                    for  $j = 1$  to  $s_2$  do
                        % The index  $j$  shows the number of the
current
                        % hidden unit in the second hidden
layer.
                            begin

```

```


$$h_{2j}^{[k+1]} := \text{Sigm} \left( \underline{c}_j^{(2)} + \sum_{i=1}^{s_1} \left( \underline{h}_{1i}^{[k+1]} \right)^T \tilde{w}_{ij} + \right.$$


$$\left. \sum_{i=1}^n \underline{U}_{ji}^{(2)} u_i \right)$$

    end j
  end k
end forall
% Approximate values for all units in each
hidden layer
% are obtained by  $K$  iterations.
 $\hat{\mathbb{H}} := \left( \bar{h}_1^{[k]}, \bar{h}_2^{[k]} \right).$ 
% Negative phase
% -----
----
% The positive phase results are used as
initial guesses
 $\underline{u}^{[0]} := \underline{u}; \quad \bar{v}^{[0]} := \bar{v}.$ 
  for  $k = 0$  to  $K - 1$  do
    begin
      for  $j = 1$  to  $s_1$  do
        % The index  $j$  shows the number of the
current
        % hidden unit in the first hidden
layer.
        begin
           $\underline{h}_{1j}^{[k+1]} \sim P \left( \underline{h}_{1j}^{[k]} \middle| \bar{v}^{[k]}, \bar{h}_2^{[k]}, \underline{u}^{[k]} \right)$ 
        end j
      for  $j = 1$  to  $s_2$  do
        % The index  $j$  indicates the number of
the current
        % hidden unit in the second hidden
layer.
        begin
           $\underline{h}_{2j}^{[k+1]} \sim P \left( \underline{h}_{2j}^{[k]} \middle| \bar{h}_1^{[k+1]}, \underline{u}^{[k]} \right)$ 
        end j
      end
    end
  end

```

```

for  $j = 1$  to  $m$  do
    % The index  $j$  shows the number of the
current
    % visible unit.
    begin
         $\underline{v}_j^{[k+1]} \sim P(\underline{v}_j^{[k]} | \bar{h}_1^{[k+1]})$ 
    end  $j$ 
    for  $j = 1$  to  $n$  do
        % The index  $j$  indicates the number of
the current
        % output unit.
        begin
             $u_j^{[k+1]} \sim P(u_j | \mathbf{H})$ 
        end  $j$ 
    end  $k$ 
end forall
% Approximate values for each unit of each
hidden layer % are obtained by means of  $K$ 
iterations.
 $\tilde{\mathbb{H}} := (\bar{h}_1^{[k]}, \bar{h}_2^{[k]}), \quad \tilde{v} := \bar{v}^{[k]}, \quad \tilde{u} := \underline{u}^{[k]}.$ 

% Update
% -----
for  $\theta \in \Theta$  do
    begin
         $\theta := \theta - \lambda \left( \frac{\partial}{\partial \theta} E(\bar{v}, \tilde{\mathbb{H}}, \underline{u}) - \frac{\partial}{\partial \theta} E(\tilde{v}, \tilde{\mathbb{H}}, \tilde{u}) \right)$ 
    end  $\theta$ 
end of Algorithm 1

```

For the negative phase we used the following conditional distributions:

$$P(\underline{v}_i | \bar{h}^{(1)}) = \mathcal{N} \left(\underline{a}_i + \underline{b}_i + \sigma_i \sum_{j=1}^{s_1} w_{ij} \underline{h}_j^{(1)}, \sigma_i^2 I \right),$$

$$\begin{aligned}
 P(u_j|\mathbf{H}) &= \text{sigm} \left(d_j + \sum_{k=1}^2 \sum_{i=1}^{s_k} \underline{h}_i^{(k)} \underline{U}_{ij}^{(k)} \right), \\
 P \left(\underline{h}_j^{(1)} \middle| \bar{\underline{v}}, \bar{\underline{h}}^{(2)}, \underline{u} \right) \\
 &= \text{Sigm} \left(\underline{c}_j^{(1)} + \sum_{i=1}^m \frac{\underline{v}_i^T \underline{w}_{ij}}{\sigma_i} + \sum_{i=1}^{s_2} \tilde{w}_{ij} \underline{h}_i^{(2)} + \sum_{i=1}^n \underline{U}_{ji}^{(1)} u_i \right), \\
 P \left(\underline{h}_j^{(2)} \middle| \bar{\underline{h}}^{(1)}, \underline{u} \right) &= \text{Sigm} \left(\underline{c}_j^{(2)} + \sum_{i=1}^{s_1} \left(\underline{h}_i^{(1)} \right)^T \tilde{w}_{ij} + \sum_{i=1}^n \underline{U}_{ji}^{(2)} u_i \right).
 \end{aligned}$$

To classify an unlabeled input $\underline{v}$ we solve the problem

$$k = \underset{i=1,2,\dots,n}{\text{argmax}} p(\bar{\underline{v}}, u_i)$$

Then the input $\bar{\underline{v}}$ is recognized as the word $\bar{\underline{v}}_k$ if $p(\bar{\underline{v}}, u_k) > \frac{1}{2}$.

If $p(\bar{\underline{v}}, u_k) \leq \frac{1}{2}$ we conclude that the input $\bar{\underline{v}}$ has not a corresponding word in the database. Successfully recognized data are stored for additional training of the DBM. Additionally we use a semantic analysis to improve the quality of recognition as it is done in [8].

4. Numerical tests

Designing a neural network we apply deep machine learning. This process is realized on the basis of the mean field approximation problem in [1, 10, 11, 4]. The role of the mean field approximation is to establish, which neurons in the hidden layers are active. Development of sufficient conditions for existence and uniqueness of the exact solution of the mean field approximation problem is a crucial point designing DBMs. Numerical tests indicate that there is a class of problems without solutions. A rigorous proof of this phenomenon could be made but it is beyond our consideration. The lack of solution leads to improper working of the corresponding neural network. Any divergent mean field approximation procedure generates confusions in the process of deep machine learning. The best way to avoid this difficulty is to work with normalized weight matrices. Note that there is a higher rate of convergence with a smaller norms of the weight matrices. Usually the initial guesses for

the hidden units are chosen randomly. This is a tradition designing neural networks. That is why it is very important to establish convergence in the case of random initial guesses. This is another difficulty training a DBM. Moreover a monotone decreasing of the error norm is strongly desirable. On the other hand it is very important to obtain fast convergence of the approximate solutions and the number of necessary iterations for satisfying the stop criterion should be independent of the number of unknowns. The training of the DBMs depends on the norms of the weight matrices. All authors in [1, 10, 11, 4] use the method of successive approximations for realize the fixed-point iteration. We establish low rate of convergence in the cases when the norms of the weight matrices is bigger than one. The number of necessary iterations for satisfying the stop criterion grows nonlinearly when the number of unknowns tends to infinity. Additionally we found a lot of cases when the method of successive approximations is divergent especially in the case of ill-conditioned weight matrices.

Let us denote the model for controlling moving objects obtained in the present paper by M_1 and the model obtained by R. T. Ilarionov, C. Jayne, T. D. Todorov, G. S. Tsanev and S. Y. Yordanov in [8] by M_2 . We compare both models on the same experimental data base. It was proved in [8] that the success of recognition strongly depends on the number of hidden units. Moreover it was obtained an optimal number of hidden units in regard to the classification success. For the model M_2 this number vary between 200 and 250. To obtain the same result with the model M_1 we need approximately 70 hidden units located in two layers. The latter means that there is a serious reduction of the used machine memory. We extend the 4-valent weight tensors W and $\tilde{W}$ to rectangular block matrices adding zero blocks where it is necessary. We write the obtained block weight matrices by the same letters. The time for training the DBM depends on the norm of the weight matrices W and $\tilde{W}$. We do not use random initial guesses for the weight matrices. All computational tests are made by positive definite and normalized initial guesses for W and $\tilde{W}$. In this case the model M_1 is much more reliable than M_2 . Additionally M_1 requires less random access memory than M_2 but M_1 needs more time for training its classifier.

6. Conclusion

A stochastic model with original classifier for controlling moving objects is investigated. A new artificial network for classifying speech commands is obtained. A detail algorithm for training the two hidden layer multiconnected Boltzmann machine is presented. The present stochastic model is compared with the corresponding single layer DBM [8] with respect to the voice command data base. The implementation indicates that important restrictions on the weight matrices are necessary. A random choice of the initial guesses for the weight matrices could adversely affect the process of convergence of the mean field approximations. The best way is to work with positive definite and normalized weight matrices. In this case the present model is more reliable than the corresponding model with shallow classifier and it needs less random access memory. Some computational tests with ill-conditioned weight matrices failed since the mean field approximation algorithm is divergent. Smaller norms of the weight matrix generates higher rate of convergence. All authors designing DBMs applied the method of successive approximations to establish the state of neurons in hidden layer to be active. As it is well-known this method generates low rate of convergence especially in the case when the norm of the Jacobian of the system is close to one. The necessary time for training a DBM can be essentially reduced if the mean field approximation problem is solved by methods with higher rate of convergence.

References

- [1] K. Cho, T. Raiko, A. Ilin, "Gaussian-Bernoulli deep Boltzmann machine," *IEEE International Joint Conference on Neural Networks, Dallas, Texas, USA*, pp. 1-7, 2013.
- [2] R. Salakhutdinov, G. Hinton, "Deep Boltzmann machines," in *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, vol. 5, p. 448-455, 2009.
- [3] G. Montavon, Klaus-Robert Müller, "Deep Boltzmann Machines and the Centering Trick," *Lecture Notes in Computer Science*, vol. 7700, pp. 621-637.
- [4] N. Srivastava, R. Salakhutdinov, "Multimodal learning with Deep Boltzmann Machines," *Journal of Machine Learning Research*, vol. 15, pp. 2949-2980, 2014.
- [5] G. Hinton, R. Salakhutdinov, "An Efficient Learning Procedure for Deep Boltzmann Machines," *Neural Computation*, vol. 24, no. 8, pp. 1967-2006, 2012.
- [6] N. Wang, D. Jancke, L. Wiskott, "Modeling correlations in spontaneous activity of

- visual cortex with centered Gaussian-binary deep Boltzmann machines," *arXiv:1312.6108v3 [cs.NE]* , pp. 1-9, 17 Feb 2014.
- [7] R. Salakhutdinov, H. Larochelle, "Efficient Learning of Deep Boltzmann Machines," *Journal of Machine Learning Research* , vol. 9, pp. 693-700, 2010.
- [8] R. T. Ilarionov, C. Jayne, T. D. Todorov, G. S. Tsanev, S. Y. Yordanov, "A Stochastic Model of Voice Control System," *International Journal of Science and Advanced Technology*, vol. 5, no. 9, pp. 7-16, 2015.
- [9] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput*, vol. 14, no. 8, pp. 1771-1800, 2002.
- [10] K. Cho, T. Raiko, A. Ilin, J. Karhunen, "A Two-Stage Pretraining Algorithm for Deep Boltzmann Machines," *Artificial Neural Networks and Machine Learning - ICANN 2013, Lecture Notes in*, vol. 8131 , pp. 106-113, 2013.
- [11] A. Dremeau, "Boltzmann Machine and Mean-Field Approximation for Structured Sparse Decompositions," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3425-3438, 2012.