



HAL
open science

Towards an Spiking Deep Belief Network for Face Recognition Application

Mazdak Fatahi, Mahmood Ahmadi, Arash Ahmadi, Mahyar Shahsavari,
Philippe Devienne

► **To cite this version:**

Mazdak Fatahi, Mahmood Ahmadi, Arash Ahmadi, Mahyar Shahsavari, Philippe Devienne. Towards an Spiking Deep Belief Network for Face Recognition Application. 6th International Conference on Computer and Knowledge Engineering (ICCKE 2016), Oct 2016, Mashhad Iran. hal-01382624

HAL Id: hal-01382624

<https://hal.science/hal-01382624>

Submitted on 17 Oct 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards an Spiking Deep Belief Network for Face Recognition Application

Mazdak Fatahi*, Mahmood Ahmadi*, Arash Ahmadi†, Mahyar Shahsavari§ and Philippe Devienne§

*Computer Engineering Department

Razi University, Kermanshah, Iran

Email: mfatahi@pgs.razi.ac.ir, m.ahmadi@razi.ac.ir

†Computer Engineering Department

Razi University, Kermanshah, Iran

Email: a.ahmadi@razi.ac.ir

‡Electrical Engineering Department

Razi University, Kermanshah, Iran

§CRIStAL laboratory

University of Lille, F-59000 Lille, France

Abstract—Understanding brain mechanisms and its problem solving techniques is the motivation of many emerging brain inspired computation methods. In this paper, respecting deep architecture of the brain and spiking model of biological neural networks, we propose a spiking deep belief network to evaluate ability of the deep spiking neural networks in face recognition application on ORL dataset. To overcome the change of using spiking neural networks in a deep learning algorithm, Siegert model is utilized as an abstract neuron model. Although there are state of the art classic machine learning algorithms for face detection, this work is mainly focused on demonstrating capabilities of brain inspired models in this era, which can be serious candidate for future hardware oriented deep learning implementations. Accordingly, the proposed model, because of using leaky integrate-and-fire neuron model, is compatible to be used in efficient neuromorphic platforms for accelerators and hardware implementation.

I. INTRODUCTION

The brain inspired techniques provide a great opportunity to imitate the brain computational capabilities to address different types of machine learning problems. Face recognition as a challenge in machine learning has been investigated perfectly using artificial intelligence tools and specifically using Artificial Neural Networks (ANN) [1], [2]. Using deep learning algorithms for training deep neural networks has improved the ANN results in many aspects of machine learning. Using deep learning and specifically Deep Belief Networks (DBN) in object recognition tasks [3], [4], speech and voice recognition [5]–[10] and image and medical images processing [11], [12] have demonstrated promising achievements.

As a fact of fact, the brain not only uses deep and hierarchical structures to represent abstract models of its sensory inputs from environment to infer subjects, but also uses spikes in coding and data interactions between pre- and post-synaptic neurons in a deep architecture. Therefore, to have better understanding of the efficiency of brain inspired computational methods in face recognition applications, one needed to:

- Utilize a deep neural network,

- Use biological models of neurons as the building blocks of the network,
- Use spike trains between neurons for sending and receiving data.

Biological neural models have been previously investigated in literature [13]–[15] in terms of their suitability to be implemented on different VLSI platforms. Considering this aspect of bio-inspired neural networks and the advancements in neuroscience on one hand and recent achievements in nanotechnology on the other hand, has introduced a new field of electronic and computer engineering called Neuromorphic. Neuromorphic engineering is related to a variety of subjects ranging from sciences such as biology, physics, mathematics to nano devices and computer engineering. The Neuromorphic term was first proposed by Carver Mead [16], which explores using VLSI for implementing neural systems. Therefore, proposing a deep spiking neural network not only shows the capability of brain like models in face recognition application, also introduces a suitable framework for deep neural networks to be implemented on Neuromorphic platforms to have a specific *Hardware Accelerator* for face recognition application.

Considering the the efficiency of Deep Belief Networks (DBN) [17] in various type of machine learning tasks such as image processing [3], [4] and video [18], voice and speech recognition [9], [10], [19], [20]. In this paper, we use a DBN which is consisted of two stacked Restricted Boltzmann Machines (RBM) [17]. The first RBM is trained in a completely unsupervised manner and the second one is used to classify the input images. To evaluate the proposed method, the Olivetti Research Laboratory (ORL) dataset [21] is used. Although there are variety of advanced works using state of the art techniques of image processing with high level of accuracy success rate of face recognition [22]–[26]; as was mentioned before, this work is not aiming to improve the result of the current face recognition algorithms, but looking for a way to evaluate deep belief networks capability in a spiking

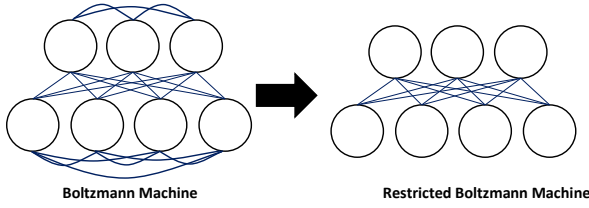


Fig. 1. Restricted Boltzmann Machine is a Restricted version of Boltzmann Machine

framework, which can be implemented in hardware platform using Neuromorphic and nanotechnology approaches.

Rest of the paper is organized as follows. According to the goal of this research, section II introduces the RBM as the main building block of the deep architectures, where we briefly talk about Contrastive Divergence (CD) as the most efficient training algorithm for RBMs. In section III the existing gap between machine learning domain and spike coding is explained and we introduce the Siegert neuron model to develop a spiking DBN. In the next, section IV, the proposed DBN with Siegert units is trained and the simulations results are presented. Using brain as a simulator for spiking neural networks, in section V we developed a DBN structure with Leaky Integrated-and-Fire (LIF) units and transferred the trained network in section IV on it to evaluate the proposed spiking deep belief network. Finally we conclude the paper and talk about our ongoing and future projects.

II. RESTRICTED BOLTZMANN MACHINE AND CONTRASTIVE DIVERGENCE

Restricted Boltzmann machines, as a model of artificial neural networks, have been driven from Boltzmann machines. RBM consists of binary stochastic units connected to each other using bidirectional edges. It can represent a probabilistic distribution to learn the basic features of an unknown distribution using observed data, which is considered as the training data. Generally, training in Boltzmann machine, as fully Recurrent Neural Network (RNN), is involved with a large number complex computations. Therefore, applying some restrictions to the Boltzmann machine topology leads to a less complex structure called Restricted Boltzmann Machine (Figure 1).

From a structural point of view, RBM has one visible and one hidden layer, where all the units in the visible and hidden layers are symmetrically connected, but there is no visible-visible or hidden-hidden connection. The structure of the Boltzmann machine is related to the proposed structure in 1982 by John Hopfield [27]. The Hopfield model is driven from thermodynamic systems and can be quantified through equilibrium energy. Each state of the Boltzmann machine can be expressed as a value called energy of the state. Equation 1 presents the energy function of a given RBM as a restricted type of Boltzmann machine.

$$E(V, H) = - \sum_i \sum_j v_i h_j w_{ij} - \sum_i a_i v_i - \sum_j b_j h_j, \quad (1)$$

where E is the total energy of the network, v_i is the state of visible i^{th} unit, h_j is the state of j^{th} hidden unit, w_{ij} is the weight between v_i and h_j , a_i and b_j are the biases. The assigned probability to each configuration of the network states are:

$$p(V, H) = \frac{1}{Z} e^{-E(V, H)}, \quad (2)$$

where $Z = \sum_{V, H} e^{E(V, H)}$ is the partition function. RBM, as a generative model, tries to generate an internal representation of its environment. Increasing the log-probability of the generating input data vector using equation 2 leads to contrastive divergence [28] updating weight rules for RBM:

$$\Delta w_{ij} = \eta (\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}). \quad (3)$$

In equation 3, $\langle v_i h_j \rangle_{data}$ represent the expectation under the distribution specified by input data vector and $\langle v_i h_j \rangle_{model}$ is the expectation under the distribution specified by internal representation of the RBM model [?]. Also the probability of h_j using a given V as the input data vector, for each j in hidden layer is 1 with probability $p(h_j = 1|V)$:

$$p(h_j = 1|V) = \sigma(b_j + \sum_i v_i w_{ij}), \quad (4)$$

when $\sigma(x)$ is the logistic sigmoid function and can be defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

In contrastive divergence using Gibbs updating chain (initiated with training data) and iteration sampling for a limited time, the approximated value for $\langle v_i h_j \rangle_{model}$ can be computed perfectly [28], [29]. A single step of Gibbs sampling using H as a given hidden vector can be expressed as

$$p(v_i = 1|H) = \sigma(a_i + \sum_j h_j w_{ij}). \quad (6)$$

According to [29] only one step of contrastive divergence (CD_1), can provide an acceptable approximation of gradient of the log probability of the training data and iterating the process for k times provides a more precise value:

$$(\Delta w_{ij})^k = \eta (\langle v_i h_j \rangle^0 - \langle v_i h_j \rangle^k), \quad (7)$$

when $\langle v_i h_j \rangle^0$ is equal to $\langle v_i h_j \rangle_{data}$ and $\langle v_i h_j \rangle^k$ means iterating Gibbs sampling for k times. Also the biased will be updated using these equations:

$$(\Delta a_i)^k = \eta (v_i^0 - v_i^k), \quad (8)$$

and

$$(\Delta b_j)^k = \eta (h_j^0 - h_j^k). \quad (9)$$

III. ARTIFICIAL NEURAL NETWORKS VERSUS SPIKING NEURAL NETWORKS

Artificial Neural Networks as a tools of artificial intelligence can demonstrate high level of accuracy in machine learning problems and specifically in face recognition applications [30], [31]. On the other side the brain inspired models and specifically Spiking Neural Networks are very suitable to be implemented in VLSI. Therefore, to have a biological model of neural networks in hardware level SNN is applied. The SNN

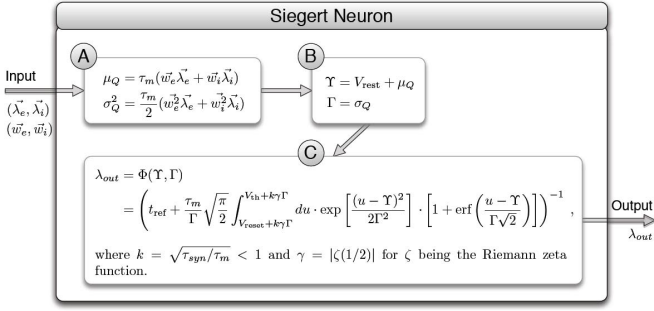


Fig. 2. Siegart abstract neuron model [34]

can be trained using an unsupervised Hebbian-based learning rules such as Spike-Timing Dependent Plasticity (STDP). However respecting to the deep architecture of the brain to have a deep SNN, we have to use efficient algorithms for deep architectures.

Spiking model of neuron uses short pulses (spikes) to coding data [32]. Since the shape of spikes are generally the same, then the rate and the time of spikes are the determinant parameters in data transferring in biological models. Comparing the biological inspired model to machine learning algorithms, it is obvious the time parameter has no role in machine learning domain. Consequently there is a gap between ANN and SNN that we have to overcome it if we want to use machine learning algorithms (ANN) in SNN platforms.

In this paper, we use the Siegart neuron model that can approximate the mean firing rate of Leaky Integrate-and-Fire neurons with Poisson-process inputs. Siegart abstract model using mathematical equations can estimate the input-output rate transfer function of Leaky Integrate-and-Fire neurons. Figure 2 shows the equation of the Siegart neuron [33], [34]. SNNs use spike trains with randomly distribution of spike times. To be able to use Siegart model in this paper, we assume the spike trains are Poisson-process with specific firing rates (λ_{in}). As it is depicted in Figure 2 the Siegart model receives excitatory and inhibitory inputs where (λ_e) is the excitatory and (λ_i) is the inhibitory rates of incoming spike trains from pre-synaptic neurons. w_e and w_i are the corresponding synaptic weights. Due to more simplicity, we did not categorize the neurons to excitatory and inhibitory neurons. We assume λ_{in} as total input spike rates from pre-synaptic neurons as well as w which indicating the both of w_e and w_i . By normalizing the value of pixels of the ORL images, we can interfere density of each pixel as the spike rate of corresponding pixel such that brighter pixel has higher spike rate and darker one has less firing rate. LIF neuron parameters can be set by adjusting the corresponding variables in Siegart equation. Table I shows the given values for the membrane time constant (τ_m), the resting potential (V_{rest}), the reset potential (V_{reset}), the threshold potential (V_{th}) and the absolute refractory time (t_{ref}).

In following sections stacking the proposed RBM with Siegart units, we developed a Deep Belief Network and after

TABLE I
LIF PARAMETERS

Parameter	Description	value
τ_m	Membrane time constant	5sec
V_{rest}	Resting potential	0
V_{reset}	Reset potential	0
V_{th}	Threshold potential	5mv
t_{ref}	Absolute refractory time	2ms

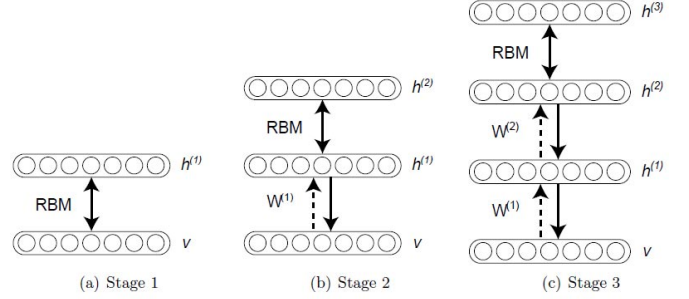


Fig. 3. Stacking RBMs as the main building blocks of DBN [35]

training phase we transferred the weight matrix to a DBN with LIF units in Brain simulator to evaluating the proposed model. According to the equality between Siegart transfer function and LIF transfer function, we can use the LIF neurons with same parameters (Table I).

IV. DEVELOPING AND TRAINING DEEP BELIEF NETWORK WITH SIEGERT UNITS

As was mentioned in the previous section, the RBMs with Siegart units can be trained in machine learning domain. In fact in this model the mean firing rate of input spike trains will be used instead of spike trains.

The Deep Belief Networks (DBN) has been introduced in [17] by professor Geoffrey Hinton as a stack of RBMs. In [17] professor Hinton has proposed a greedy layer wise algorithm to training the DBN by training the each RBM sequentially. Figure 3 illustrate the stages of stacking RBMs. In this paper, we used the ORL face dataset that is taken at the Olivetti Research Laboratory in Cambridge, UK [21]. This dataset originally contains 400 grey scale face images of 40 distinct volunteers. Figure 4 shows some images of ORL dataset. In this paper, we used a resized version of ORL that has been proposed in [36]. The input images with 32×32 pixels are used as input vectors with 1024 elements. According to [37] in each step of RBM training process a mini-batch of training data consisted of a small subset of all training data is used. The proposed DBN architecture respecting to the size of training vectors is depicted in Figure 5. The first RBM has 1024 (32×32) units in visible layer and 500 units in hidden layer. This RBM is trained without any label (unsupervised learning) to provide abstract features for the second one. After training the first RBM, the second RBM using the extracted features generated in previous step, is trained. In the recent RBM as a classifier, we used the joint of 500 extracted values and 80



Fig. 4. Some sample images from ORL dataset

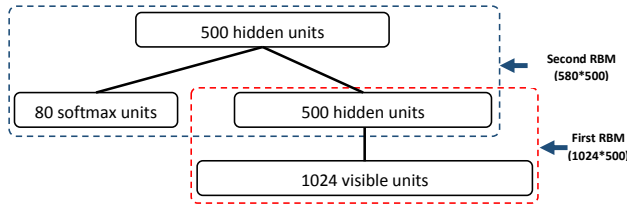


Fig. 5. The proposed DBN with Siebert neurons for learning ORL

softmax units. The labels of ORL images have been converted to softmax vectors such that for the first class of images the two first bits are one and others are zero, for the next one the two first bits are zero the next two bits are one and the others are zero and so on.

We divided the ORL images into two subsets. The first one is the training set and consisted of 8 images from 10 of each class. These images are used to train the model and didn't use for testing. The second one is the test images containing 2 of 10 from each class. Because of the full connections between each visible unit and each hidden unit, the dimensions of the arriving connections at each hidden unit are the same as the dimension of input images. Figure 6 shows the corresponding weights of connections between all visible units and 100 randomly selected hidden units. The weight vectors has been reshaped as 32*32 images. During the learning process the hidden units have learned some specific features and learned to be triggered only with some specific features. Visualizing these weight vectors (Figure 6), displays the learned features by each hidden unit and it is a perfect monitoring method to studying the network learning process [37]. As we can see in Figure 7, the training process needed too many iteration to reach a proper result. For this model which is implemented in Matlab, after about 800 iterations the results are close to 90%. The maximum value, 93.2%, is corresponded to iteration 1910th. In this paper, we used the Free Energy function (equation 10). To find the predicted label, using Free Energy function, each possible label has been tested to find the configuration

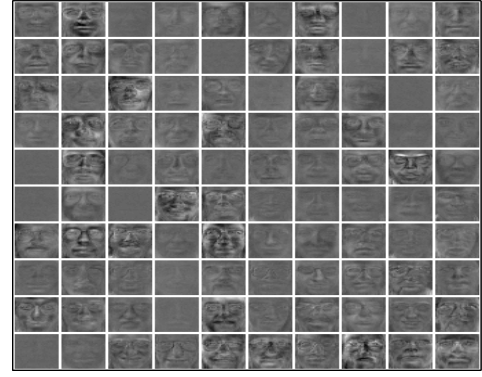


Fig. 6. Visualizing the learned features by hidden units

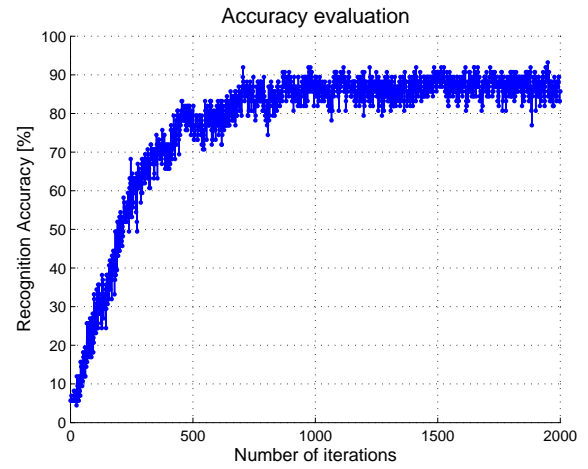


Fig. 7. Accuracy of the proposed DBN with Siebert neurons in face recognition on ORL dataset

with the lowest energy. The corresponding label to the recent configuration is assumed as the predicted label [37].

$$F(V) = - \sum_i v_i a_i - \sum_j \log(1 + e^{x_j}) \quad (10)$$

where

$$x_j = b_j + \sum_i v_i w_{ij}$$

The accuracy of the model is depended on the learning parameters [37], [38]. For example Figure 7 shows the results when the mini-batch size is 4. The effect of changing the model parameters using less iterations and using various mini-batch sizes is depicted in Figure 8. Obviously less iteration and larger mini-batch size leads to less accuracy. Eventually if one is interested in more precise accuracy, it can be possible through adjusting the learning parameters and also the Siebert neuron parameters (Table I).

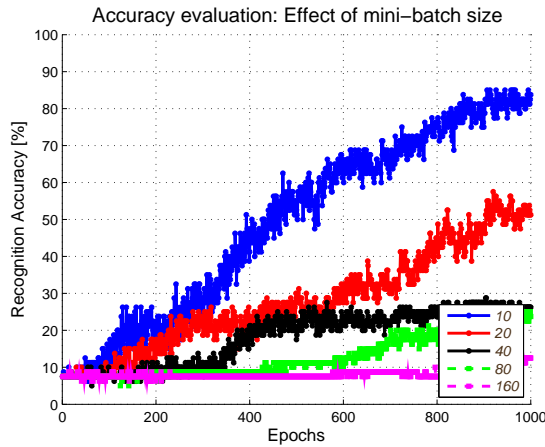


Fig. 8. Decreasing the number of epochs and increasing the mini-batch size reduced the model accuracy

V. EVALUATING THE MODEL

Thanks to the equality between the Siegert neuron's transfer function and the LIF transfer function, without any adjustments the trained weight matrix in the previous section can be copied to a network with same topology consisted of LIF neurons with same parameters (Tabel I). To develop such a network with LIF neuron we used the Brian simulator. Brian is a simulator for Spiking Neural Networks. This simulator is written in the Python programming language. Because of using development tools such as SciPy module, Python provides very fast routines for mathematical operations and specifically matrix operations [39]. We have developed a Deep Belief Network in Brian simulator with same topology as the one that has been implemented in Matlab. In this model since we want to test the accuracy of the model in a more realistic situation, the equation of LIF neuron (Equation 11) is applied besides the described parameters in Table I.

$$\tau_m \frac{dv}{dt} = -(v(t) - v_{rest}) + RI(t) \quad (11)$$

We know the spiking model uses spike trains instead of the real numbers. In Section IV we have talked about the basis of the assumption to use the normalized value of the pixels as the related firing rate. Consequently in this section to test the proposed spiking model, we have to convert the firing rate to spike trains. In [40] we have discussed in details about this approach for MNIST handwritten digits. In Brian simulator, using *PoissonGroup* function, we can generate spike trains with specified firing rates. Therefore the network can be tested with spike trains corresponding to the density of pixels of the test images. Having the trained weights matrix, despite of the Matlab model, the labels are not used in the second RBM for training but they are used as the outputs of the model for classifying the input images [41]. Respecting to the generative characteristic of DBNs, the model not only can reconstruct input images as it is internal representation of the given image (Figure 9), but also can generate the corresponding learned

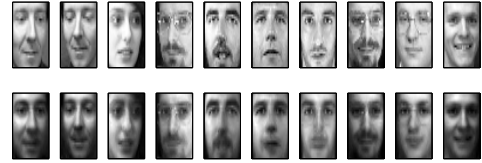


Fig. 9. The upper row shows 10 of the training images and the lower one illustrate the corresponding reconstructed images

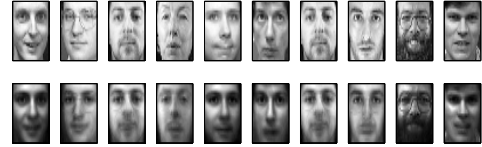


Fig. 10. The upper row shows 10 of the test images and the lower one illustrate the predicted images

labels as the predictions of the model. To evaluate the model in a spiking framework after transferring the weights matrix, the generated spike trains of test images are passed through the entire of the network and comparing the predicted labels with the original labels, we computed the model accuracy. Figure 10 shows predicted images respecting to the input test images. In addition, as it was predictable, the results of the model with LIF neurons have not changed considerably. However because of the difference between floating number precision in Matlab and Brian simulator reloading the weights matrix in Python can cause a small decreasing in accuracy. The accuracy of the model in Brian simulator reduced to 92.4%.

VI. CONCLUSION AND FUTURE WORKS

The ability of brain like computing motivated us to evaluate a model with biological structure in face recognition application. Regarding the brain as a deep neural network, we proposed a deep neural network with spiking neurons to understand if the brain like models are suitable for face recognition applications. The proposed model is different from a traditional neural network. Indeed this model is a Spike-Based deep model with biological inspired neurons (Leaky Integrate-and-Fire neurons). Considering the results of Deep Belief Networks in various tasks in Machine learning domain, a DBN with two RBM has been developed in Matlab with Siegert units. Then the trained weights matrix transferred to a Spiking DBN with LIF neurons. The recent model is simulated in the Brian simulator and the results shows the capability of the Spiking Deep Belief Networks in a simple face recognition application.

The accuracy of the model may improve using some parameters adjustment. Also utilizing some pre-processing such as using Gabor and Wavelet filters for extracting features can be useful. Therefore we are going to enhance the accuracy of model using the mentioned methods. For future works, we will take into account using other types of deep models such as Deep Autoencoders or Convolutional Neural Networks.

To use the proposed model, we have to train the model in one platform and using the trained weights in another one. Indeed our model uses an offline training model. Using the same platform for training and utilizing the model leads us to an online model of training. Additionally for next future work, we are looking for a way to perform the same work in a single platform next to other different model of neurons such as Izhikevich model.

ACKNOWLEDGMENT

This research has been supported by Center of International Scientific Studies and Collaboration (CISSC) of Iran and french embassy in Iran as Gundishapour project

REFERENCES

- [1] F. Smach, M. Atri, J. Miteran, and M. Abid, "Design of a neural networks classifier for face detection," *Matrix*, vol. 15, p. 15, 2006.
- [2] C. Tsai, W. Cheng, J.-S. Taur, and C.-W. Tao, "Face detection using eigenface and neural network," in *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on*, vol. 5. IEEE, 2006, pp. 4343–4347.
- [3] V. Nair and G. E. Hinton, "3d object recognition with deep belief nets," in *Advances in Neural Information Processing Systems*, 2009, pp. 1339–1347.
- [4] W. Diao, X. Sun, X. Zheng, F. Dou, H. Wang, and K. Fu, "Efficient saliency-based object detection in remote sensing images using deep belief networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 2, pp. 137–141, Feb 2016.
- [5] A.-r. Mohamed and G. E. Hinton, "Phone recognition using restricted boltzmann machines," in *ICASSP*, 2010, pp. 4354–4357.
- [6] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.
- [7] A.-r. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 14–22, 2012.
- [8] X. Xiao, S. Zhao, D. H. H. Nguyen, X. Zhong, D. L. Jones, E. S. Chng, and H. Li, "Speech dereverberation for enhancement and recognition using dynamic features constrained deep neural networks and feature adaptation," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, pp. 1–18, 2016.
- [9] G. Dahl, A.-r. Mohamed, G. E. Hinton *et al.*, "Phone recognition with the mean-covariance restricted boltzmann machine," in *Advances in neural information processing systems*, 2010, pp. 469–477.
- [10] M. Jagadeesh, M. A. Kumar, and K. Soman, "Deep belief network based part-of-speech tagger for telugu language," in *Proceedings of the Second International Conference on Computer and Communication Technologies*. Springer, 2016, pp. 75–84.
- [11] T. Schmah, G. E. Hinton, S. L. Small, S. Strother, and R. S. Zemel, "Generative versus discriminative training of rbms for classification of fmri images," in *Advances in neural information processing systems*, 2008, pp. 1409–1416.
- [12] J. J. Kivinen and C. Williams, "Multiple texture boltzmann machines," in *International Conference on Artificial Intelligence and Statistics*, 2012, pp. 638–646.
- [13] H.-Y. Hsieh and K.-T. Tang, "Vlsi implementation of a bio-inspired olfactory spiking neural network," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 7, pp. 1065–1073, 2012.
- [14] G. Indiveri, E. Chicca, and R. Douglas, "A vlsi array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 211–221, Jan 2006.
- [15] G. Indiveri, B. Linares-Barranco, and T. Serrano-Gotarredona, "Neuromorphic silicon neuron circuits," *Frontiers Neurosci.*, vol. 5, no. 73, pp. 1–23, May 2011.
- [16] C. Mead and M. Ismail, *Analog VLSI implementation of neural systems*. Springer Science & Business Media, 2012, vol. 80.
- [17] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [18] L. Bazzani, H. Larochelle, V. Murino, J.-a. Ting, and N. D. Freitas, "Learning attentional policies for tracking and recognition in video with deep networks," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 937–944.
- [19] A.-r. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," in *Nips workshop on deep learning for speech recognition and related applications*, vol. 1, no. 9, 2009, p. 39.
- [20] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.
- [21] (2016) The database of faces. [Online]. Available: <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>
- [22] M. Sharif, M. A. Ali, M. Raza, and S. Mohsin, "Face recognition using edge information and dct," *Sindh University Research Journal-SURJ (Science Series)*, vol. 43, no. 2, 2015.
- [23] C. Zhou, L. Wang, Q. Zhang, and X. Wei, "Face recognition based on pca image reconstruction and lda," *Optik-International Journal for Light and Electron Optics*, vol. 124, no. 22, pp. 5599–5603, 2013.
- [24] G.-F. Lu, Z. Jin, and J. Zou, "Face recognition using discriminant sparsity neighborhood preserving embedding," *Knowledge-Based Systems*, vol. 31, pp. 119–127, 2012.
- [25] H. Li and C. Y. Suen, "Robust face recognition based on dynamic rank representation," *Pattern Recognition*, vol. 60, pp. 13–24, 2016.
- [26] Z.-H. Huang, W.-J. Li, J. Wang, and T. Zhang, "Face recognition based on pixel-level and feature-level fusion of the top-levels wavelet sub-bands," *Information Fusion*, vol. 22, pp. 95–104, 2015.
- [27] G. E. Hinton and T. J. Sejnowski, "Learning and relearning in boltzmann machines," *Parallel distributed processing: Explorations in the microstructure of cognition*, vol. 1, pp. 282–317, 1986.
- [28] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.
- [29] M. Welling and G. E. Hinton, "A new learning algorithm for mean field boltzmann machines," in *Artificial Neural Networks ICANN 2002*. Springer, 2002, pp. 351–357.
- [30] W. K. Mleczo, T. Kapuściński, and R. K. Nowicki, "Rough deep belief network-application to incomplete handwritten digits pattern classification," in *Information and Software Technologies*. Springer, 2015, pp. 400–411.
- [31] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Modeling human motion using binary latent variables," in *Advances in neural information processing systems*, 2006, pp. 1345–1352.
- [32] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.
- [33] A. J. Siebert, "On the first passage time probability problem," *Physical Review*, vol. 81, no. 4, p. 617, 1951.
- [34] F. Jug, J. Lengler, C. Krautz, and A. Steger, "Spiking networks and their rate-based equivalents: does it make sense to use siebert neurons?" *Swiss Society for Neuroscience*, 2012.
- [35] (2016) Deep learning tutorials. [Online]. Available: <http://deeplearning.net/tutorial/>
- [36] D. Cai, X. He, Y. Hu, J. Han, and T. Huang, "Learning a spatially smooth subspace for face recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition Machine Learning (CVPR'07)*, 2007.
- [37] G. E. Hinton, "A practical guide to training restricted boltzmann machines," in *Neural Networks: Tricks of the Trade*. Springer, 2012, vol. 7700, pp. 599–619.
- [38] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 437–478.
- [39] D. F. Goodman and R. Brette, "The brian simulator," *Frontiers in neuroscience*, vol. 3, p. 26, 2008.
- [40] M. Fatahi, M. Ahmadi, M. Shahsavari, A. Ahmadi, and P. Devienne, "evt_mnist: A spike based version of traditional mnist," *arXiv preprint arXiv:1604.06751*, 2016.
- [41] H. Larochelle and Y. Bengio, "Classification using discriminative restricted boltzmann machines," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 536–543.