2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)

Image classification with deep belief networks and improved gradient descent

Gang Liu, Liang Xiao, Caiquan Xiong
School of Computer Science
Hubei University of Technology
Wuhan, China
Email: lg0061408@126.com

***Abstract*—Image classification mainly uses the classifier to classify the extracted image features. In the traditional image feature extraction, it is difficult to set the appropriate feature patterns for the complex images. Simultaneously, the training algorithm of the classifier also affects the accuracy of image classification. In order to solve these problems, the combination of deep belief networks and the classifier is used for image classification. In the new image classification method, an improved gradient descent method is proposed to train the classifier. Restricted boltzmann machines can be stacked and trained in a greedy manner to form deep belief networks. The deep belief networks are used to extract the features of images and the classifier is used to classify these feature vectors. Compared with other depth learning methods to extract the image features, the deep belief networks can recover the original image using the feature vectors and can guarantee the correctness of the extracted features. Experimental verifications are conducted on MNIST dataset. Experimental results indicate that compared with other classifiers, our approach can get the higher classification accuracy in less time and have superior anti-overfitting ability.**

I. Introduction

Image classification is to classify a set of images into the specified classes and can be applied in many fields, such as face recognition [1] and hyperspectral image classification [2]. In classification methods, the commonly used method is feature-based method. Feature-based method extracts the features from the images and then those features are classified by the classifier. In the method, the extracted features and the training algorithm of the classifier have direct influence on the classification results. At present, some feature-based classification methods [3], [4] are proposed. However, for many image classification problems, it is still difficult to extract the appropriate features. In addition, the appropriate training algorithm of the classifier can get higher classification accuracy.

Deep learning technology (DL) [5], which can extract features from data automatically through its unique hierarchical structure, has been proposed to solve the problems of feature extraction. However, there is no corresponding physical or mathematical meaning for the features extracted by deep learning. Hence, it cannot guarantee the correctness of the feature extraction. In order to solve the above problems, deep belief networks (DBNs) [6], which is a kind of deep learning technology (DL), is proposed. The concept of DBNs is to model high-level abstractions in the data by using multiple processing layers with complex structures. There are many different ways to implement DBNs, such as restricted boltzmann machines (RBMs) [7] and auto-encoder [8]. Currently, DBNs have been applied to image classification and made some achievements [9], [10].

For feature-based method, the training algorithm of the classifier is also important. In image classification, artificial neural network (ANN) [11] and support vector machine (SVM) [12] are widely used. The training algorithms used by these classifiers are mainly the traditional gradient descent methods. However, the traditional gradient descent methods are easy to fall into the local optimum. This will make the the algorithm being stagnation and there will also be premature convergence in optimization. The situation shows that it is inefficient to use the traditional gradient descent methods and the hand-tuning approaches to find the suitable values for these parameters. Evolutionary algorithms (EAs) [13] have a long history of successfully solving global optimization problems. Compared with the traditional gradient descent methods, EAs have better adaptability and flexibility. But as a training algorithm of the classifier, the randomness of EAs causes the algorithm to converge slowly. In fact, image classification can be divided into two phases: feature extraction and classification. Two stages can be performed in the order. It can avoid to adjust the parameters of the entire network simultaneously and reduce the difficulty of adjusting parameters.

Based on the above considerations, this paper proposed an improved gradient descent method, called evolutionary gradient descent algorithm (EGD). EGD combines the advantages of evolutionary strategy (ES) [14] and the gradient descent methods and it is used as a training algorithm of the classifier. In this paper, DBN is composed of the stacked RBMs, which is used to extract the features from the images and then those extracted features will be classified by the softmax classifier. EGD is proposed as the training algorithm of the softmax classifier and is used to optimize the parameters of the softmax classifier. In EGD, the gradient descent method and ES take turns to optimize the softmax classifier and improves the classification accuracy. MNIST test data set is used to verify the proposed method. The experimental results show that the proposed classification method has better accuracy and anti-over-fitting than other image classification methods.

The remainder of this paper is organized as follows. The

978-1-5386-3221-5/17 $31.00 © 2017 IEEE
DOI 10.1109/CSE-EUC.2017.74

principle of the architecture of the stacked RBMs are described in Section II. The image classification method based on the stacked RBMs and the softmax classifier with EGD is presented in detail in Section III. Experimental results and discussions are reported in Section IV. Finally, some conclusions are given in Section V.

II. Restricted Boltzmann Machines

Deep belief networks is used to automatically extract data features. Unlike other depth learning techniques, DBNs can use the features to restore the original data. Hence, DBNs can verify the correctness of the extracted features. Restricted Boltzmann Machines are stochastic, energy-based neural network models and a stack of restricted Boltzmann machines is considered to be a DBN. Restricted Boltzmann machine is a bipartite graph consisting of the visible layer and the hidden layer. The two layers are connected by the weights. The visible layer acts as the training data input layer and the hidden layer acts as the feature detectors. The structure of RBMs is shown in Fig.1, where v is the visible layer, h is the hidden layer and W is a sets of weight. The neurons in RBMs are activated to generate the outputs according to the given probability. The neurons of the hidden layer tend to model features and patterns occurring in the data and they can be trained to model the joint distribution of the data.

The stacked RBMs are trained by the contrast divergence algorithm [7], which is an unsupervised learning method and ensure the robustness of the extracted features. The contrast divergence algorithm is used to adjust the connection weights of the visible layer and hidden layer. After adjusting the weights, the new hidden layer is stacked on the current hidden layer. And then the contrast divergence algorithm trains the connection weights of the new hidden layer and the current hidden layer. Namely, RBMs are can be trained layer by layer. It can stack the different number of the layers according to the problem. For RBMs, the classifier determines the effect of classification and is very important. In terms of image recognition, the RBMs are widely used for its efftive [15] or as a method of preprocessing [7].

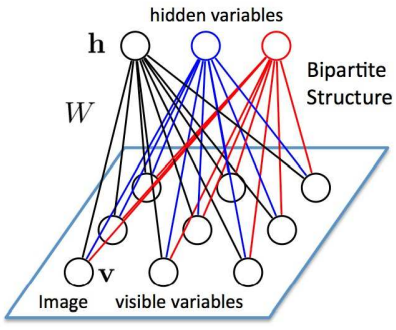

Fig. 1. The structure of restricted Boltzmann machines.

III. The Improved Gradient Descent Algorithm

In our method, the evolutionary gradient descent algorithm is used to train the softmax classifier to establish the classifier model. The feature vectors of the training set is as the input of the classifier. The role of of EGD is mainly used to adjust the weights and biases of the classifier. EGD is composed of the gradient descent algorithm and ES. In EGD, two algorithms take turns to optimize the parameters of the softmax classifier. It is worth noting that only the parameters of the output layer of the softmax classifier are optimized by ES in EGD and the parameters of other layers are optimized by the gradient descent algorithm in EGD. The parameters of other layers are fixed and remain unchanged during ES optimization process. This method is similar to the extreme learning machine [16]. This can save the optimization time of EGD.

For EGD, the gradient descent method is first used to optimize the parameters of the classifier. When the gradient descent method falls into a stagnation, the gradient descent method stops running. And then ES is used to optimize the parameters of the output layer of the softmax classifier. In this paper, the $1 + \lambda$ strategy is used. The employed ES is only based on selection and mutation operators. No crossover is applied because the literature[17] have indicated that the crossover operators do not significantly improve the quality of the search. According to the literature [17], $\lambda = 4$. The mutation scheme used is as follows:

$$w_{i,j} = w_{i,j} * 0.5 + (w_{i,r1} + w_{i,r2} - w_{i,r3}) * 0.5 \qquad (1)$$

where $w_{i,j}$ denotes the jth parameter (weight or bisas) of the ith neuron in the output layer, and $r1, r2, r3 \in \{1, 2, \ldots, D\}$ are randomly chosen integers, D is the dimension of the solution vector, $r1 \neq r2 \neq r3 \neq i$. Eq. 1 is inspired by differential evolution (DE) [18]. In addition, the fitness function is described as:

$$f(T, V) = \frac{\sum |output_i - class_i|}{M} \qquad (2)$$

where T is the training set, V is the encoding vector of the parameters, M is the number of examples in the training set, $output_i$ is the class label for the ith image outputted by the softmax classifier, $class_i$ is the class label of the ith image.

The framework of our method is shown in Algorithm 1.

In EGD, the gradient descent algorithm is used as a global optimization algorithm to optimize all the parameters of the classifier and ES is used as a local optimization algorithm to optimize the parameters of the output layer. ES is essentially a random search method, compared to the gradient descent algorithm, its optimization speed is relatively slow. But the exploitation ability of ES is stronger and are not affected by the gradient of the problems. Hence, the combination of the gradient descent algorithm and ES can achieve complementary advantages. Two search schemes based on the different principles can search the solution space effectively and improve the optimization ability of the training algorithm. ES in EGD only optimizes the parameters of the output layer in order to save

Algorithm 1 The improved gradient descent algorithm

Require: The training set T, mutation rate mr, maximum number of iterations, $MaxGen$

Ensure: The parameters of the classifier

1: Use contrastive divergence algorithm to train the stacked restricted Boltzmann machines.
2: Get the feature vectors of the training set.
3: Set counter $G = 1$.
4: **while** $G < MaxGen$ **do**
5: (The gradient descent algorithm) Use the gradient descent algorithm to update the parameters of the softmax classifier.
6: If $|f(k) - f(k+1)| < eps$, eps is a prefixed small tolerance, $f(k)$ is the fitness of the kth iteration of the gradient descent algorithm, then the gradient descent method is terminated.
7: (Evolutionary strategy) The parameters of the output layer of the softmax classifier optimized by the gradient descent algorithm are used as the best individual in ES.
8: Using Eq. 1 to create λ mutants of the best individual.
9: Create a new $1 + \lambda$ population using λ mutants and the best individual.
10: Evaluates all individuals by the fitness function.
11: Find the best individual.
12: G=G+1.
13: **end while**
14: **return** best individual.

the algorithm running time. The method balances the search ability and the optimization speed.

IV. Experiments

A. Benchmark dataset

In this section, a comprehensive sets of experiments is performed to verify the performance of the proposed approach. Experiments have been performed using a publicly available image dataset MNIST created by Yann LeCun[19]. The dataset is originally generated to demonstrate the capacity of deep networks to learn problems. In MNIST, the images are formatted as $28 * 28$ grayscale images and they are handwritten digits $0 - 9$, as shown in Fig. 2. MNIST has 60000 training images and 10000 test images.

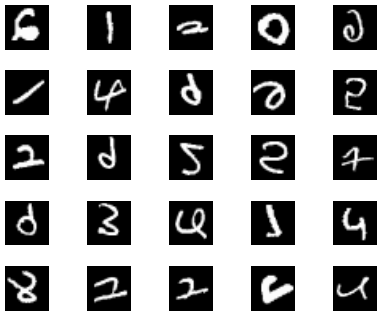

Fig. 2. Samples from the mnist dataset

B. Parameter Settings

For EGD, the following parameters are used: the population size $\lambda = 4$, the mutation rate $MR = 0.03$, the maximum number of iterations $MaxGen = 10000$, $eps = 0.01$. The structure of the stacked RBMs used for extracting the feature vectors is $784 - 300 - 100$ and the maximum number of iterations is 20000. The training set consists of 10000 examples selected from MNIST randomly and the test set consists of 2000 examples from MNIST. The input data of the softmax classifier is the output data the stacked RBMs. Because the images in MNIST are classified into 10 different classes, the structure of the softmax classifier is $100 - 10$.

Experiments were conducted using a machine with a Intel Core i3 (2.2Ghz) processor and 2GB of RAM. EFACV and the stacked RBMs were trained and implemented in Visual Studio, without GPU acceleration and multithreading.

C. Results and analysis

In this section, the performances of our approach are analyzed and our approach is compared with the combination of the stacked RBMs and softmax classifier, the combination of the stacked RBMs and support vector machine and the combination of the stacked RBMs and ES-based softmax classifier. In all experiments, the images in MNIST are classified into 10 different classes. The weights of the first layer of the stacked RBMs are shown in Fig. 3. The Fig. 3 shows several fuzzy digit-like shapes, which indicates that stacked RBMs have

learned the features of the images in the training set and the extracted features are reliable. The results of the combination

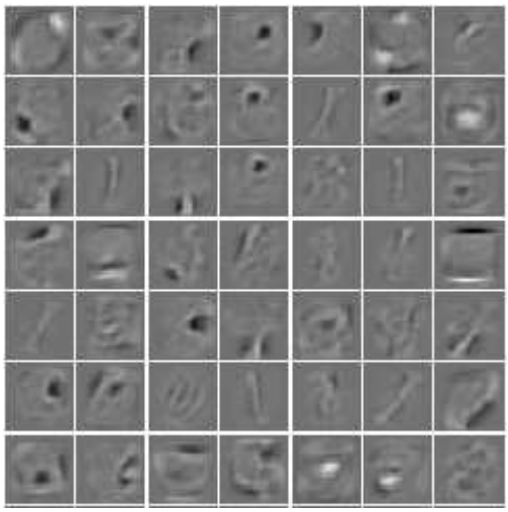

Fig. 3. Samples from the mnist dataset

of the stacked RBMs and the softmax classifier on the training set are shown in I. From Table I, on the training set, the highest accuracy is 99.88% obtained by the $1st, 2nd, 3rd, 9th, 10th$ classification label and the lowest accuracy is 91.50% obtained by the $5th$ classification label; on the test set, the highest accuracy is 100% obtained by the $1st, 2nd, 9th, 10th$ classification label and the lowest accuracy is 87.56% obtained by the $5th$ classification label. The accuracy on the training set of the softmax classifier is above 91.00% and it shows that the softmax classifier achieve good performance on the classification. It is worth noting that the error value between the accuracy on the training set and the accuracy on the test set are small. The maximum error value is 3.94% obtained by the $5th$ classification label and the minimum error value is 0.12% obtained by the $1st, 2nd, 9th, 10th$ classification label. All of the error values are below 4% and it shows the softmax classifier optimized by EGD has strong anti-overfitting ability. In general, the experimental results show that the proposed training algorithm has a strong ability to find the optimal solution. Fig. 4 shows the performance graphs of the softmax classifier optimized by EGD for both the training set and the test set. In Fig. 4, it is worth noting that the accuracy on the training set is the fitness value in EGD and the accuracy on the test set is calculated using the best individual in each generation. It can be observed that the algorithm has converged around 10000 generations and the convergence rate is faster. With increasing the generations of the training algorithm, the accuracy on the training set and the test set are gradually increased and tends to be stable. It also reflects the softmax classifier optimized by EGD has strong anti-overfitting ability.

TABLE I
RESULT OF THE COMBINATION OF THE STACKED RBMS AND THE SOFTMAX CLASSIFIER ON THE TRAINING SET

Classification label	Training set accuracy	Test set accuracy	Error
1	99.88%	100%	0.12%
2	99.88%	100%	0.12%
3	99.68%	99.08%	0.6%
4	96.72%	94.73%	1.99%
5	91.50%	87.56%	3.94%
6	95.46%	93.25%	2.21%
7	96.28%	94.11%	2.17%
8	99.76%	99.51%	0.25%
9	99.88%	100%	0.12%
10	99.88%	100%	0.12%

In order to verify the performance of our proposed approach, the softmax classifier optimized by EGD is compared with other popular classified methods. These methods are the combination of the stacked RBMs and the softmax classifier optimized by the traditional gradient descent method, the combination of the stacked RBMs and SVM and the combination of the stacked RBMs and the ES-based softmax classifier. The feature vectors used by all the classifiers in the experiments are extracted by the same stacked RBMs. The following parameters are used: 1) for the softmax classifier optimized by the traditional gradient descent method, the learning rate is 0.1, the momentum is 0.5 and the activation function is the sigmoid function; 2) for SVM, radial basis function (RBF) is used as the kernel function and the scailing factor is 1; 3) for ES-based softmax classifier, the population size $\lambda = 4$, the mutation rate $MR = 0.03$, the activation function is the sigmoid function. It is worth noting that the train algorithm of RBMs is not ES and only the train algorithm of softmax classifier is ES. The target to check is whether the softmax classifier optimized by EGD is better or worse than some state-of-the-art classified methods. All methods need to classify the images in MNIST into 10 different classes.

The cost time and error rate of the softmax classifier optimized by EGD, the softmax classifier optimized by the traditional gradient descent method, SVM and the ES-based softmax classifier are shown in Table II. In table II, the softmax classifier optimized by the traditional gradient descent method is called the traditional softmax classifier and the softmax classifier optimized by EGD is called the EGD-based softmax classifier. Among the 4 methods, the best results are shown in **boldface**. The results show that the EGD-based softmax classifier gets the lowest error in the training set and the test set (4.05% and 6.11%), and SVM gets minimal time cost for the image classification. The traditional softmax classifier requires most time because it uses the back propagation algorithm (BP) to adjust the weights of whole network include the stacked RBMs. EGD only adjust the parameters of the softmax classifier, hence, the time required is reduced. Compared with the traditional softmax classifier, the EGD-based softmax classifier is better in both the accuracy and the time cost. It means that EGD is superior to the traditional gradient descent method. ES is used in both ES-based softmax classifier and the EGD-based softmax classifier, but the EGD-based softmax classifier is still better in both the accuracy and the time cost. It indicates that EGD is an effective training method for the softmax classifier. Although the cost time of the EGD-based

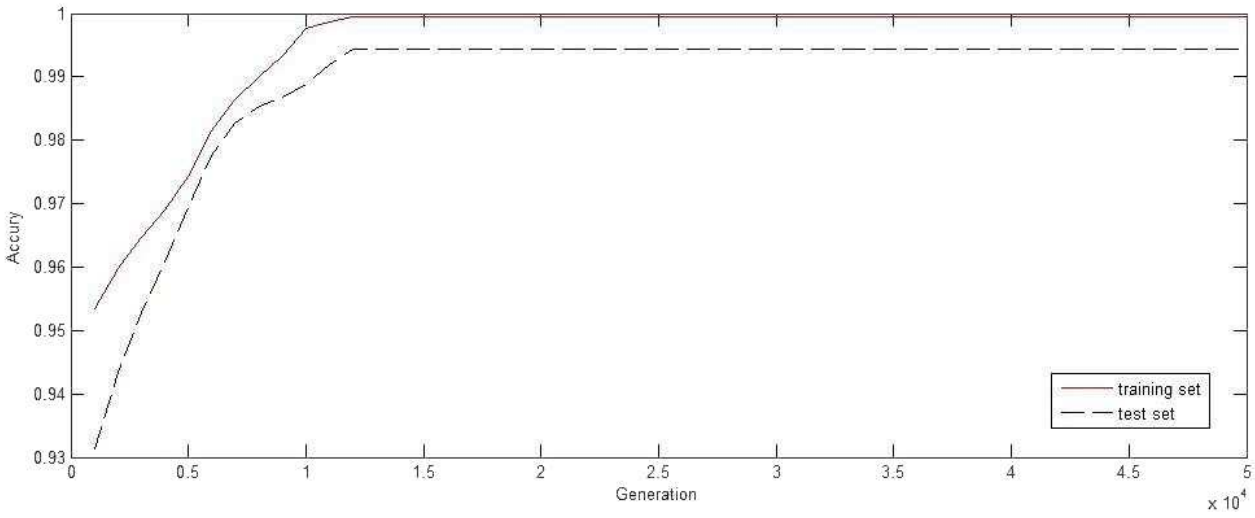

Fig. 4. The classification accuracy of the softmax classifier optimized by EGD on the training set and the test set

softmax classifier is higher than that of SVM, the error rate of the EGD-based softmax classifier is lower than that of SVM. Furthermore, the difference between the error of the training set and the error of the test set for the EGD-based softmax classifier is far below the the difference of SVM. These results demonstrate the EGD-based softmax classifier has better anti-overfitting ability and classification accuracy than the softmax classifier, SVM and the ES-based softmax classifier. The experimental results also show that the EGD-based softmax classifier is able to get the high accuracy in relatively short time and can be trained very fast.

TABLE II
COMPARISON OF THE ERROR VALUES BETWEEN FOUR CLASSIFIED METHODS FOR MNIST

Method	Training set error	Test set error	Cost time
Traditional softmax classifier	17.07%	18.71%	108000s
EGD-based softmax classifier	**4.05%**	**6.11%**	1550s
SVM	23.78%	40.73%	**110s**
ES-based softmax classifier	8.87%	11.26%	108000s

To further investigate our approach, the wine dataset from the UCI machine learning repository [20] is used in the paper. The total dataset contains 178 instances and 150 instances are randomly selected as training set, 28 instances are as test set. Each instance is 13 dimensions and all instances are divided into 3 different classes. Table III shows the result of each classifier on the wine dataset from the UCI machine learning repository.The training set error, test set error and cost time are listed respectively. The best results are shown in **boldface**. The EGD-based softmax classifier obtains the best performance both in the training set 0.42% and test set 0.73%. The cost time of SVM is the least and it has relatively high accuracy 4.66% for the train set. However, the difference between the error of the training set and the error of the test set for SVM is higher than other methods and it shows that SVM is over-fitting. The results of ES-based softmax do not differ significantly from the softmax classifier. The reason might be that the dataset is easy to solve for both classifiers.

TABLE III
COMPARISON OF THE ERROR VALUES BETWEEN FOUR CLASSIFIED METHODS FOR THE UCI DATASET

Method	Training set error	Test set error	Cost time
Traditional softmax classifier	12.00 %	14.28%	120s
EGD-based softmax classifier	**0.42%**	**0.73%**	12s
SVM	4.66%	10.71%	**3s**
ES-based softmax	13.33%	17.85%	120s

V. DISCUSSIONS

For the MNIST and the wine data set, the EGD-based softmax classifier is the fastest and the highest classification accuracy in 4 different classifiers. The key features of our approach are as follows: 1) Because the stacked RBMs are only used to extract image features, EGD only adjusts the parameters of the softmax classifier and the time required to train the classifier is reduced; 2) EGD combines the advantages of two algorithms and improves the ability to find the optimal solution of the gradient descent algorithm; 3) ES in EGD only optimizes the parameters of the output layer. This method avoids the problem that ES is inefficient when dealing with the high dimensional data and balances the search ability and the optimization speed. All experiments results indicate EGD remarkably accelerate the convergence rate and improves exploitation ability of the traditional gradient descent method. It shows that EGD has a better solving ability and our proposed EGD performs better than some state-of-the-art training algorithms.

VI. Conclusion

For feature-based image classification, feature extraction and the training algorithms of the classifiers are very important. However, it is difficult to extract the appropriate features and the traditional training algorithms are is easy to fall into the local optimum. In order to solve these problems, a novel training algorithm is presented, named evolutionary gradient descent algorithm (EGD). The EGD-based softmax classifier combines with the stacked RBMs to constitute a novel image classification method. The stacked RBMs is used to extract the features from the images and then those extracted features be classified by the EGD-based softmax classifier. EGD is composed of the gradient descent algorithm and ES. In EGD, two algorithms take turns to optimize the parameters of the softmax classifier. The experimental results show that the proposed method can obtain higher accuracy in a relatively short time, and has strong anti-overfitting ability. Comparisons with the traditional the gradient descent algorithm and ES, it demonstrates that EGD is more effective and efficient in terms of the classification accuracy and the cost time. It is an effective training algorithm for image classification.

Future works should includes investigating the performance of our approach and applying our approach to the practical applications.

Acknowledgment

The work described in this paper was support by National Natural Science Foundation of China, Foundation No.61300127. Any conclusions or recommendations stated here are those of the authors and do not necessarily reflect official positions of NSFC.

References

[1] H. Kamitomo and C. Lu, "3-d face recognition method based on optimum 3-d image measurement technology," *Artificial Life and Robotics*, vol. 16, no. 4, pp. 551–554, 2012.

[2] H. Su and P. Du, "Multiple classifier ensembles with band clustering for hyperspectral image classification," *European Journal of Remote Sensing*, vol. 47, no. 1, pp. 217–227, 2014.

[3] N. Nechikkat, V. Sowmya, and K.P.Soman, "Variational mode feature-based hyperspectral image classification," *Advances in Intelligent Systems and Computing*, vol. 380, no. 2016, pp. 365–373, 2016.

[4] L. Shen, Z. Zhu, S. Jia, J. Zhu, and Y. Sun, "Discriminative gabor feature selection for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters.*, vol. 10, no. 1, pp. 29–33, 2013.

[5] L. Deng and D. Yu, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3-4, pp. 197–387, 2013.

[6] 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.

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

[8] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends in Machine Learning*, vol. 2, no. 1, pp. 1–27, 2009.

[9] H. Lee, C. Ekanadham, and A. Y. Ng, "Sparse deep belief net model for visual area v2," in *Advances in Neural Information Processing Systems 20 - Proceedings of the 2007 Conference*. Google, December 2007, pp. 873–880.

[10] S. Kang, X. Qian, and H. Meng, "Multi-distribution deep belief network for speech synthesis," in *Proceedings of 2013 IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, May 2013, pp. 8012–8016.

[11] J. A. Hertz, *Introduction to the theory of neural computation*. Boulder, USA: Westview Press, 1991.

[12] J. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[13] T. Back and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *Evolutionary Computation*, vol. 1, no. 1, pp. 1–23, 1993.

[14] I. Rechenberg, *Evolution Strategy: Optimization of Technical Systems by Means of Biological Evolution*. Stuttgart,Germany: Fromman-Holzboog, 1973.

[15] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008.* IEEE, 2008, pp. 1–8.

[16] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[17] J. Wang, Q. Chen, and C. Lee, "Design and implementation of a virtual reconfigurable architecture for different applications of intrinsic evolvable hardware," *IET Computers and Digital Techniques*, vol. 2, no. 5, pp. 386–400, 2008.

[18] K. Price, R. Storn, and J. Lampinen, *Differential Evolution - A Practical Approach to Global Optimization*. Berlin, Germany: Springer, 2005.

[19] Y. LeCun and C. Cortes, "Mnist handwritten digit database," *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2010.

[20] C. L. Blake and C. J. Merz, "Uci repository of machine learning databases, university of california," *www http://www.ics.uci.edu/mlearn/MLRepository.html*, 1989.