International Journal of u- and e- Service, Science and Technology
Vol.8, No. 11 (2015), pp.317-326
http://dx.doi.org/10.14257/ijunesst.2015.8.11.31

The Genetic Convolutional Neural Network Model Based on Random Sample

You Zhining and Pu Yunming*

School of Computer Technology, Jimei University, Xiamen 361021, China
**yunmingpu@163.com*

Abstract

Convolutional neural network (CNN) -- the result of the training is affected by of initial value of the weights. It is concluded that the model is not necessarily the best features of expression. The use of genetic algorithm can help choosing the better characteristics. But there almost was not literature study of the combining genetic algorithm with CNN. So this research has a lot of space and prospects. GACNN convolution genetic neural network model based on random sample has a better solution to obtain the unknown character expression. CNN individual training set uses a random data set. At the same time, the crossover and the mutation genetic algorithm bring random factors. There may are unknown feature expressions that may be appropriate. Experiments are based on accepted MNIST data sets, and the experimental results proved the advantages of the model.

Keywords*: convolutional neural network (CNN); genetic algorithm; random sample; generalization*

1. Introduction

Convolutional neural network is inspired by biological neural network. There is a weight sharing network structure to make it more similar to the biological neural networks, reducing the complexity of the network model and the number of weights. CNN has a local perception area, hierarchical, feature extraction and classification process combined with the characteristics of the global training. It is obtained widespread application in the field of image recognition.

The evolution of the nature is composed of population genetic evolution and individual learning. Congenital gene and acquired learning decides the individual fitness to the environment. Because the individual living environment is different, the individual learning samples are different. Regard Each CNN as an individual and seeing from the perspective of individual learning, the parameters of the initial value selection is a historical problem. People often give initial value in the form of experience. This approach is often subjective. If it is able to use genetic algorithm to obtain the optimal individual parameters of the initial value, this is an exciting and worth exploring direction to try. Previously, there is not document mentioned combination of genetic algorithm with CNN on experiment. This paper presents an improved genetic algorithm. Each individual sample order of training data set is random. This leads to an advantage to improve the generalization of the model. Because a random sample data set is equal to the introduction of the random factors, to improve generalization.

In this article, we design a model of a hybrid genetic algorithm and CNN. The model will select the characteristic of CNN by crossover and mutation based on genetic algorithm.

* Corresponding Author

ISSN: 2005-4246 IJUNESST

These experiments use a 0-9 handwritten numerals MNIST dataset. Character recognition is one of the challenging problems in pattern recognition. The use of actual application including bank check recognition, postal address, *etc*... Because of the different writing style, an effective character recognition system construction is still a challenging task. Character recognition includes three steps: preprocessing, feature extraction and classification. The focus of this article is classified. Classification method is mainly divided into four categories, namely pattern matching, neural networks, statistical and structural techniques [8]. Pattern matching technology is based on the similarity in the feature space vectors. Statistical and structural are classified by using statistical function, such as K - nearest neighbor classifier (K - NN), bayes classifier and support vector machine (SVM), and the hidden markov model (HMM) [8]. And the classification of the artificial neural network is the most popular computing model. In order to improve the generalization of neural network, different strategies have been proposed such as regularization [8, 9] or minimization methods [10, 11]. Neuro-evolution is a class of artificial neural networks in which evolution is another form of adaptation in addition to learning. Neuro-evolution can be achieved according to different ways. The most popular forms include evolution of neural networks' structures, their connection weights, their training parameters, and its training data. This article GACNN model is a form of connection weights' evolution.

2. Algorithm Constitute

This article GACNN model is consisted of CNN parts and GA algorithm. CNN model exists as individual in the GA algorithm. GA algorithm's fitness function accords to the validation set S_{valid} error rate instead of the training set S_{train} error rate. Its advantage is that model generalization is better; Crossover operation's exchanging is based on feature unit. It matches the purpose to define each convolution kernels that is all down connecting weights and threshold value for a feature. The GA algorithm can choose the better expression of characteristics. During each individual CNN model train, the order of sample is random. Regarding The CNN model as individual existence in the GA algorithm, the relevant literature study is less, and there is almost no more models applied to the large number samples. This paper hopes to get actual experimental results of GACNN model under the large sample, verifying the performance of the GACNN model compared with pure CNN model.

2.1. CNN

Convolution neural network (CNN) [5] is a multilayer supervised learning framework. It can be seen as two parts: an automatic feature extraction and a neural network training classifier. Feature extractor includes feature mapping layer and retrieval from the original image recognition function through the two operations: convolution filtering and sampling. In the convolution model of this paper, the characteristics of the nuclear map size are 5 * 5 pixels, and the subsampling size are 2 * 2. Classifier is implementated by softmax function and argmax function.

This article does not use complex CNN structure [6], but refer to simplify the structure of the CNN [7]. Our CNN structure is shown in Figure 1.There are seven layers.

CNN training algorithm uses CUDA GPU stream processors equipment to parallel convolution of the neural network calculation [convolution neural network recognition algorithm based on CUDA technology].The purpose is to improve the training speed.

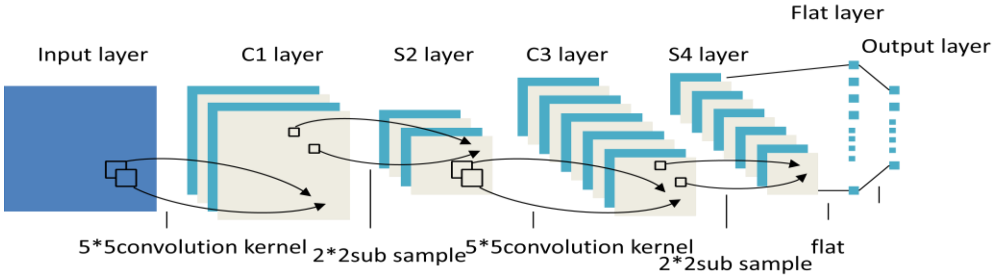

Figure 1. CNN Construction

A) The size of the input image is 32 x32.C1 layer is the convolution layer. The characteristics of each neuron in the graph are connected to the input 5 * 5 in the neighborhood. Figure for the size of the 28 * 28, this can prevent the input connection falling beyond the boundaries. It is the size of C1 is 28*28. Setting 20 different layers of C1 here, each weight in same C1 layer is the same.

B) S2 layer is the sub sampling layer. In short, 2*2 = 4 points sample to one point, it is the weighted average of the 4 points.

C) C3 is the convolution layer, it convolutes layer S2 through 5*5 convolution kernel. Then getting the characteristics of the map are only 14*14 neurons. But it has 15 different convolution kernels. So there are 15 feature maps.

D) S4 is the sub sampling layer. It is figured by 15 characteristic of 7*7. Features diagram of each unit are connected with C3 in the corresponding figure of 2*2.

E) Flat layer corresponds to flatter neurons of S4 maps, such as a feature layer S4 figure map flatter to a linear 49 neurons. All S4 figure will be changed into the linear layer of neurons.

F) The entire flat linear neurons will be connected to the output layer neurons. This layer is the output of the network layer. If it is to identify 0-9, the layer has 10 nodes.

Output vector can be get like this:$o(x) = G(b + Wh(x))$, $h(x)$ is the output of the flat floor for the input x. Function G can be defined as a function of softmax function:

$$P(Y = i \mid x, W, b) = \mathrm{softmax}_i(Wx + b) = \frac{e^{W_i x + b_i}}{\sum_j e^{W_j x + b_j}} \quad (1)$$

The function can get the probability value for each category (on the MNIST data set, the Numbers 0 to 9). So you can use argmax function to achieve classification probability value to obtain the biggest class labels:

$$y_{pred} = \mathrm{argmax}_i P(Y = i \mid x, W, b) \quad (2)$$

CNN Training:

BP (backpropagation algorithm) algorithm was used to study the parameter set (W, b).

2.2. GA Algorithm

GA algorithm is mimic natural biological genetic and evolutionary process. In 1975, it is first introduced by Holland JH [8]. It is mainly used in the use of computers to solve optimization problems. What is the typical application of genetic algorithm is to search the large "space" of possible solutions of the problem effectively [9-11]. GA uses the generations of evolutionary random individual to do searching. A new generation of members of the individual comes from the previous generation of crossover and mutation, and the concept of the imitation selection comes from Darwin's natural selection [12]. GA randomly generates candidate solution to form initial population [13-15]. In every generation, all individuals are ranked. The most possible to adapt to the environment in probability of the individual is selected as the "father". Forming on the basis of future population (or the next "generations"), it is to replace the current population [16]. Repeating this process, there will produce more and more individual that have the ability to solve the problem [17]. The genes of the new generation individuals are created by genetic operations of crossover and mutation [18]. The specific implementation of Crossover and mutation depends on the solution which is selected and said to be the effective solution of application field. In this paper, the improvement to classical GA algorithm is focused on the design of fitness function, crossover operation with convolution kernels as cross genes, each individual operation using random order sample set. The following is the algorithm description:

The GA algorithm can be defined as a space of seven dimensions:

$$\mathrm{GA} = (\mathrm{M}, \mathrm{F}, \mathrm{s}, \mathrm{c}, \mathrm{m}, \mathrm{pc}, \mathrm{pm})$$

We regard M as the population size, F as individual fitness evaluation function, and *s* as selecting operator, c as crossover operator, m as the mutation operator, the pc as the proportion of crossover operation, pm as the proportion of variation. GA algorithm defines a pool to hold the new generation of the next generation of individuals. The size of the pool is 2M. We compute the individual fitness within pool, and converts individual fitness to individuals selected probability value according to certain probability transformation formula, and select M individuals to form the next generation of population by the probability value.

Fitness is to measure the degree of a species to adapt to living environment. On the survival environment, the species of high fitness will get more chances. And those species with low fitness to adapt to the living environment will get less chance. What’s worse, they will extinct gradually. In this paper, the fitness function of GACNN model is:

$$\mathrm{F} = \begin{cases} \left(\frac{1}{\mathrm{e}}\right)^{\alpha}, & \mathrm{e} > 0 \\ \mathrm{FMAX}, & \mathrm{e} = 0 \end{cases} \tag{3}$$

The e is the error rates for the individual model on validation set $\mathrm{S}_{\mathrm{valid}}$ .You can see the smaller error rate, the bigger the fitness.

Selection is the certain probability operation from a selected number of individuals in a population. This paper model uses function Q_{i} and R_{i}:

$$\mathrm{Q}_{\mathrm{i}} = \begin{cases} \mathrm{F}_{\mathrm{i}}, & \mathrm{F}_{\mathrm{i}} > 0 \\ 0, & \mathrm{F}_{\mathrm{i}} = 0 \end{cases} \tag{4}$$

$$\mathrm{R}_{\mathrm{i}} = \mathrm{Q}_{\mathrm{i}} - \mathrm{Q}_{\mathrm{min}} \tag{5}$$

When each individual's R_{i} are normalized to R'_{i}, and we set $\mathrm{R}''_{\mathrm{i}} = \mathrm{R}'_{\mathrm{i}} + \beta$. Then each individual's $\mathrm{R}''_{\mathrm{i}}$ are normalized to get the probability P_{i} of each individual value. According to the probability values we define the each individual range $\tau_{\mathrm{i}} \in \left[\sum_{\mathrm{k}=0}^{\mathrm{i}-1} \mathrm{P}_{\mathrm{k}}, \sum_{\mathrm{k}=0}^{\mathrm{i}} \mathrm{P}_{\mathrm{k}}\right)$of probability. According to the random function to obtain the random value, if the random value falls on the middle of τ_{i}, the individual *i* is selected.

Crossover is the exchanged operation at the same location between two chromosomes. After exchanging, it becomes two new chromosomes. It is also called genetic recombination. In CNN model, we only exchange layer weights and threshold value of C1 and C3. Each convolution kernels contains weights which are down connection and threshold. We can regard each convolution kernels as a feature mapping. We cross to the individual. That is the exchange of the characteristics. So every time when we need to exchange, we can only exchange a convolution kernel connection weights and threshold nodes.

Variation is that when the cell is to reproduce, there could be some small probability replication errors. Thus there occurs some mutations in the DNA, and new chromosomes appear. To individual CNN model, we randomly select a convolution kernels, and then modify the convolution kernels all the weights that are down connection and thresholds.

2.3. The Algorithm after Combining with Two Parts:

A) Set the initial value, generation $g = 0$.

B) If g is greater than the maximum evolution generation *gmax*, then output the best individual, and quit, if not then continue.

i) *M*pc* times to select two individuals from the population, then cross, then generate *2*M*pc* CNN individuals in the pool.

ii) *M*pm* times to choose an individual, then mutate, and generate *M*pm* CNN individuals in the pool.

iii) *2M-2*M*pc-M*pm* times to select from a population in the individual, then directly put into the pool.

iv) For each CNN individual in the pool repeats the following:

- Adjusting sample order of train set.
- Training individual CNN model with the adjusted train set.
- Calculating the error rate *misclass* with CNN model trained in the valid set.
- According to the *misclass*, calculate individual fitness and record the best individual.

v) generating a new generation of population from the pool.

vi) B)

3. Introduction of the Experiment

3.1. In this paper, the experiment use MNIST data set. It is composed of handwritten digital image. It is divided into 60000 train data and 10000 test data. Training data is further divided into 50000 train training data set and 10000 valid validation data set. In order to facilitate the selection of model parameters, all the images did the standardized processing. The size of each image is 28 * 28. In the original data, the image pixel into common gray image (gray level range 0 ~ 255).

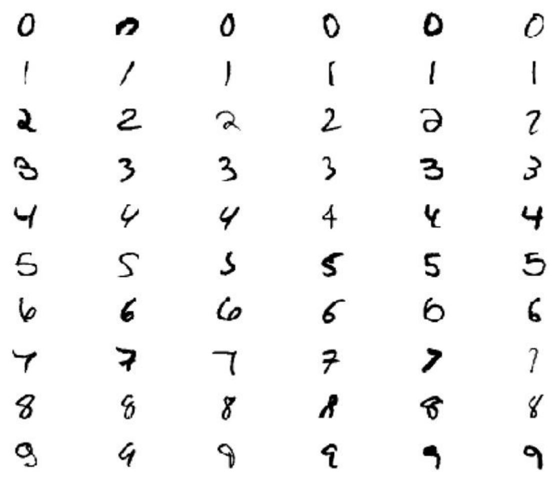

Figure 2. The Samples of MNIST Data Set

3.2. The seven dimensions of GA algorithm that used in GACNN model set to $\mathrm{GA} = (30, \mathrm{F}, \mathrm{s}, \mathrm{c}, \mathrm{m}, 0.5, 0.45)$. In other words, the population size is 30, the pool size is 60, the proportion of crossover operation is 0.5, and the proportion of the mutation operation is 0.45.

3.3. The experimental model of CNN has input layer, C1, the S2 layer, C3, S4, flat layer and output layer. The map from the input layer to C1 nuclear layer sets 20 characteristics. The map from S2 layer to C3 sets 15 characteristics of the nucleus. Output layer has 10 nodes. Such model is denoted as CNN (20-15).

3.4. The experiment training pure CNN model alone. The results were compared with the corresponding GACNN model, namely: a) Defining CNN (20 -15); b) Defining that the map from the input layer to C1 nuclear layer set 40 characteristics, the map from S2 layer mapped to C3 set 30 characteristic, denoted as CNN(40-30); c) Defining that the map from the input layer to C1 nuclear layer set 80 characteristics, the map from S2 layer mapped to C3 set 60 characteristics, denoted as CNN(80-60). The initial value of weights for each CNN is random, each CNN models randomly generated three times, there are nine models, using the train data set for training, according to the valid data set to get the best, and verify the result on the test data set.

3.5. We also define two models of large number characteristic nuclear which are CNN (240-180) and CNN (350-250). We also use the train data set for training. According to the valid data set to get the best, and verify the result on the test data set, the results were compared with the corresponding GACNN model.

4. The Experimental Results

Experiment platform is a GPU server. The GPU chip is nVIDIA GeForce GTX 770, the number of CUDA Stream processor is 1536, GPU memory size is 2G, 8G server memory, operating system is 64-bit Windows 7, the algorithm using CUDA library, make full use of the GPU parallel computing ability.

We regard CNN(20-15) as individual GACNN model. The genetic evolution generation is 61 generations. The error rate of the optimal individual obtained in valid data set is 0.00739999767393.The best individual in the test data set on the error rate is 0.00879999715835.The experiment result is shown in the figure below:

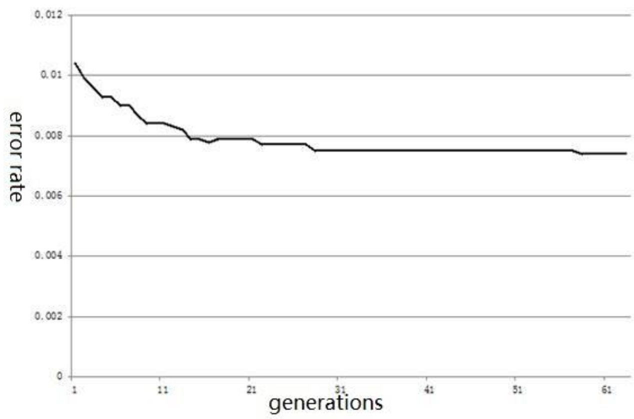

Figure 3. The Error Rate Changes of Best Individual on Valid Data Set

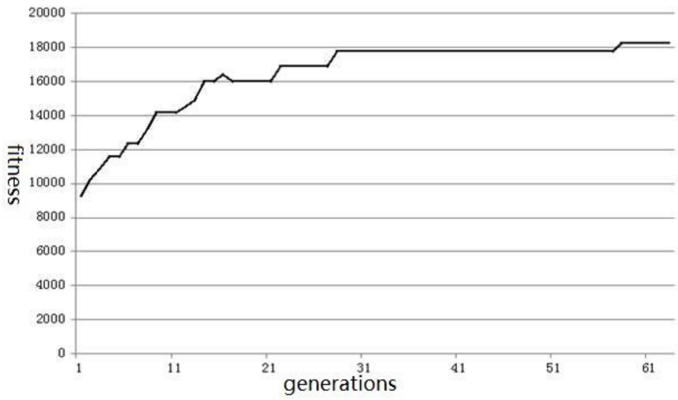

Figure 4. The Best Individual Fitness

The experimental results of 2.4 are as follows:

Table 1. The Training Results of Nine Random Initial Value's CNN Models

Model		CNN(20-15)	CNN(40-30)	CNN(80-60)
The random initial value1	The error rate on valid set	0.0136999944225	0.0136999944225	0.0122999940068
	The error rate on test set	0.0124999973923	0.0141999963671	0.0112999966368
The random initial value2	The error rate on valid set	0.0138999940827	0.0111999958754	0.0118999956176
	The error rate on test set	0.0134999938309	0.0133999977261	0.0104999979958
The random initial value3	The error rate on valid set	0.0157999955118	0.0134999947622	0.0117999957874
	The error rate on valid set	0.0156999975443	0.0125999944285	0.0104999989271
Mmathematical expectation	Valid set	0.0145	0.0128	0.0120
	The error rate on test set	0.0139	0.0134	0.0107
standard deviation	The error rate on valid set	0.000946926	0.001134	0.000216
	The error rate on test set	0.001336663	0.000653	0.000383

The experimental results of 2.5 are as follows:

Table 2.The Training Results of CNN (240-180) (350-250) and CNN

CNN(240-180)		CNN(350-250)	
The error rate on valid set	The error rate on test set	The error rate on valid set	The error rate on test set
0.01029999740 42	0.009499997831 88	0.009799995459 62	0.009199997410 18

5. Analysis of Experimental Results

According to the above experimental data, we get the following conclusions:

A. Comparing the experiment results of the GACNN model with the experiment results of the CNN (20-15) model, we discovered GACNN model's performance is better than that of pure CNN model. The best individual error rate is reduced on the valid data set. The reason is that GACNN model is optimized by genetic algorithm.

B. From the GACNN model experiment results, we can see that the more iterative generations, the better. Due to the role of genetic algorithm, when we are searching for the optimal model, there always is the opportunity to jump out of local optimum and approach to the global optimal. But with the evolution being, the best individual error rate declining will become more and slower.

C. According to table 1, we can analyze groups of experimental data, and calculate mathematical expectation and variance. In probability theory and mathematical statistics, the standard deviation is a measure that decides dispersion degree of a set of data .A larger standard deviation means that most of the value is more different from the average; a smaller standard deviation means that the numerical value is close to the average. We can from table 1 where the standard deviation value is small, experimental results of the other random initial model is close to mathematical expectation. With this conclusion, we can further approximate think that the results of CNN (240-180) and CNN (350-250) is the same as the mathematical expectation of CNN (240-180) and CNN (350-250).

D. According to table 1, we compare three groups of experimental data. We found that the more the number of convolution kernels is, the result of mathematical expectation is more optimal. That is to say that the relations of the error rate's mathematical expectation of the CNN(20-15), (40-30) and CNN (80-60) on valid set is as follows: 0.0145 > 0.0128 > 0.0120, and the relation of the error rate's mathematical expectation on the test set is as follows: 0.0139 > 0.0134 > 0.0108. The more the convolution kernel is, the more characteristic is expressed. So the performance is better. Comparing the CNN (40-30) and CNN (80-60) with GACNN experimental results, we found that although the CNN (40-30), CNN (80-60) convolution kernels is more and GACNN convolution kernel number is less at the same time, GACNN is the optimal characteristics of genetic algorithm selected. So the model has good generalization and better performance.

E. When performing CNN(350-250) training, because the convolution kernels number is too much, 2G of GPU memory is nearly depleted. The size of GPU memory is the key factor in parallel computing. When more convolution neural network of convolution kernels parallel computing, the model is restricted by the GPU memory size. But GACNN model chooses less number of convolution kernels; therefore it is constrained by the GPU memory less. According to the data in table 2 (the approximate mathematical expectation) and the results of model GACNN we found that GACNN model performance is much better than convolution model. So the universality of GACNN model is better.

6. Sum Up the Main Points

The training result of Convolutional neural network (CNN) is affected by the weights of initial value largely. At the same time there is no better way to get better features of expression. This article choose less convolution kernels CNN model as the genetic individual, and use genetic algorithm to selection of convolution kernels. The evolutionary model has better characteristics of expression. And every training in the process of genetic evolution, the data set is random. It is like the introduction of random factor. At the same time, the crossover and mutation of genetic also bring random factors. To some extent, it overcome the over fitting and improves the generalization. So the model has good universality. The experimental comparison results show that the characteristics of the individual's model which comes from genetic screening are some better characteristics. Its performance has large improvement. The next research direction is to combine deep belief networks [19, 20] with genetic algorithm. The purpose is to improve the performance of deep belief networks.

Acknowledgments

The research work was supported by Natural Science Foundation of Fujian Provincial under Grant No. 2013J01243 and Science Project Foundation of Fujian Provincial under Grant No. 2013H0032.

References

[1] P. Gallinari and T. Cibas, "Practical Complexity Control in Multilayer Perceptrons", Signal Process, vol. 74, no. 1, **(1999)**, pp. 29–46.
[2] T. Simila and J. Tikka, "Combined Input Variable Selection and Model Complexity Control for Nonlinear Regression", Pattern Recognition Letter, vol. 30, no. 3, **(2009)**, pp. 231-236.
[3] A. Abraham and D. Steinberg, "Is Neural Network a Reliable Forecaster on Earth? A MARS Query!", Lecture Notes Computer Science, vol. 2085, **(2001)**, pp. 679–686.
[4] T. Gagnon and R. Lefebvre, "A Neural-network Approach for Pre-classification in Musical Chords Recognition", Proceeding 37th Asilomar conference on signals, systems and computers, Asilomar, Pacific Grove, **(2003)**.
[5] Y. LeCun, L. Bottou and Y. Bengio, "Gradient-based Learning Applied to Document Recognition", Proceedings of the IEEE, **(1998)**.
[6] F. Lauer, C. Y. Suen and G. Bloch, "A trainable Feature Extractor for HandwrittenDigit Recognition", Pattern Recognition, vol. 40, no. 6, **(2007)**, pp. 1816–1824.
[7] W. Pan, T. D. Bui and C. Y. Suen, "Isolated Handwritten Farsi Numerals Recognition Using Sparse and Over-Complete Representations", Proceedings of the International Conference on Document Analysis and Recognition, **(2009)**; Barcelona, Spain.
[8] J. H. Holland, "Adaptation in Natural and Artiðcial Systems", Ann Arbor: University of Michigan Press, **(1975)**, p. 2.
[9] S. N. Kumar and R. Panneerselvam, "A Survey on the Vehicle Routing Problem and Its Variants", Intelligent Information Management, vol. 4, no. 3, **(2012)**, pp. 66 -74.
[10] P. Gemperline, A. Niazi and R. Leardi, "Genetic Algorithms in Chemometrics", Journal of Chemometrics, vol. 26, no. 6, **(2012)**, pp. 345-351.
[11] H. Khajemohammadi, A. Fanian and T. A. Gulliver, "Efficient Workflow Scheduling for Grid Computing Using a Leveled Multi-objective Genetic Algorithm", Journal of Grid Computing, vol. 12, no. 4, **(2014)**, pp. 637-663.
[12] J. J. McDowell and A. Popa, "Toward a Mechanics of Adaptive Behavior: Evolutionary Dynamics and Matching Theory Statics", Journal of the Experimental Analysis of Behavior, vol. 94, no. 2, **(2010)**, pp. 241-260.
[13] D. Álvarez, R. Hornero and J. V.Marcos, "Feature Selection from Nocturnal Oximetry Using Genetic Algorithms to Assist in Obstructive Sleep Apnoea Diagnosis", Medical Engineering & Physics, vol. 34, no. 8, **(2012)**, pp. 1049-1057.
[14] H. Tian, C. Liu and X. D. Gao, "Optimization of Auto-induction Medium for G-CSF Production by Escherichia coli Using Artificial Neural Networks Coupled with Genetic Algorithm", World J Microbiol & Biotechnol, vol. 29, no. 3, **(2012)**, pp. 505-513.
[15] P. Qiao, L. Zheng and L. Ma, "Research on a Niche Genetic Algorithm", Journal of Harbin University of Science and Technology, vol. 16, no. 1, **(2011)**, pp. 90-93.

[16] F. Naznin, R. Sarker and D. Essam, “Vertical Decomposition with Genetic Algorithm for Multiple Sequence Alignment”, BMC Bioinformatics, vol. 12, no. 353, **(2011).**
[17] A. Wongsarnpigoon and W. M. Grill, “Energy-efficient Waveform Shapes for Neural Stimulation Revealed with a Genetic Algorithm”, Journal of Neural Engineering, vol. 7, no. 4, **(2010)**, pp. 046009
[18] N. Zaki, S. Bouktif and S. L. Molnar, “A Combination of Compositional Index and Genetic Algorithm for Predicting Transmembrane Helical Segments”, PLoS One, vol. 6, no. 7, **(2011)**, pp. 21821.
[19] G. E. Hinton and S. Osindero, “A Fast Learning Algorithm for Deep Belief Nets”, Neural Computation, vol. 18, no. 7, **(2006)**, pp. 1527-1554.
[20] G. E. Hinton, P. Dayan and B. Frey, “The Wake-sleep Algorithm for Unsupervised Neural Network”, Science, vol. 268, no. 5214, **(1995)**, pp. 1158-1161.