Dimension Reduction by Principal Component Analysis for Evolutionary and Back-propagation Neural Networks

Jung-Guk Park
Dept. of Computer Engineering
Sejong University
98 Gunja-Dong, Gwangjin-Gu, Seoul
Republic of Korea
prfirst@sju.ac.kr

Kyung-Joong Kim
Dept. of Computer Engineering
Sejong University
98 Gunja-Dong, Gwangjin-Gu, Seoul
Republic of Korea
kimkj@sejong.ac.kr

ABSTRACT

In machine learning, back-propagation neural network and evolutionary neural network have been widely used to solve classification problems. However, evolving neural network suffers from high dimensional space which is proportional to the dimension of data. Back-propagation neural network is often stuck into local optima. These are related to the curse of dimensionality decreasing the performance of classification. Principal Component Analysis (PCA) is widely used to reduce data dimension and our work shows how can PCA affects the learning of two representative neural networks. In this paper, our work presents the relationship between data dimension and classification methods. Our work used six UCI benchmark data sets and our empirical studies showed that PCA affects differently on the performance of back-propagation neural network and evolving neural network.

Keywords

Principal Component Analysis, Evolutionary Neural Network, Back-propagation Neural Network, Data Analysis, Classification

1. INTRODUCTION

Solving classification problem is one of the main issues in pattern recognition and popular domain in machine learning. Evolving Neural Network (ENN) [1] has been researched in the last 20 years since ENN solves well the optimization problems [2],[3], and yields good performance in classification problems [4]. There are a lot of research on ENN [5],[6] because GA has potential for searching globally, but it struggles for the convergence of a solution in case of large dimensionality. On the other hand, back-propagation [7] yields a great convergence to a solution while there is a risk of possibly stucking in local optima. Randall *et al.* [4] shows that ENN yields better accuracy than back-propagation neural network. Unlike back-propagation neural network, as the properties of genetic algorithm, ENN can come out the local optima. However, ENN searches for the optimal weights in random space which is exponentially proportional to the number of weights of neural network. It is described as the curse of dimension [8]. Our work focuses on data dimension effect on the learning of ENN and BPNN.

Principal Component Analysis [12] has been widely used to reduce data dimension with statistical method. Mayank,.A. *et al*. [10] applied PCA for face recognition, Javed, K. *et al*. [9] used PCA for gene expression profile classification and Francisco, C. *et al*. [11] used it for ECG signal processing. In this paper, our work shows that how dimension of PCA effects the learning of ENN comparing BPNN relative to the data dimension.

2. Methodologies

2.1 Principal Component Analysis

Principal Component Analysis is a useful method for signal processing [11], face recognition [10], data analysis [12] and the work with high dimensional data [9].

PCA yields a projection axis which reduces the original data dimension within separated data. Let the sample data as a vector $\mathbf{x}_n=\{x_{n1},x_{n2},x_{n3},\ldots,x_{nD}\}^T$, D is the dimension of data and N $(1<n<N)$is the number of sample data. $\mathbf{y_n}=\{y_{n1},y_{n2},y_{n3},\ldots y_{nD'}\}^T$ is a reduced vector($D>D'$) from $\mathbf{x_n}$. Let

$$\mathbf{y} = A^T\mathbf{x} \tag{1}$$

The matrix A is a D x D' transform matrix. The aim of PCA is to find a new projection axis to separate the input data. The sample data are subtracted from the mean of the data set so that samples have zero mean. It is needed to minimize the mean squared error of approximation of the data.

To separate, data the variance of data should be maximum.

$$\sigma^2 = \frac{1}{N}\sum_i^N (y_i - \bar{y})^2 = \frac{1}{N}\sum_i^N (A^t x_i - A^t \bar{x})^2 \tag{2}$$

Where $\bar{y}$ and $\bar{x}$ denote the mean of x and y , respectively. (2) can be solved by Lagrange multipliers [8] constrained condition $A^TA=1$

$$L = \frac{1}{N}\sum_{i}^{N}(A^t x_i - A^t \bar{x})^2 + \Lambda(1 - A^t A) \qquad (3)$$

Where Λ denotes Lagrange multipliers and $\frac{\partial L}{\partial A}$ allows $\Sigma A = \lambda A$. The eigenvectors of covariance Σ is a unit axis on which the data variance is the maximum. So, the matrix A consists of $\{\phi_1, \phi_2, \ldots \phi_{D'}\}$, where ϕ_n $(1 \le n \le D)$ is the eigenvectors following the nth λ eigenvalue of the covariance matrix from the data set ($\lambda_1 > \lambda_2 > \ldots \lambda_D$).

2.2 Evolutionary Neural Network

Kim *et al.* [5] represented ENN architecture using a matrix based encoding. It is straightforward to represent a node and easy to implement (Figure 1). In this case, it is a conventional one hidden layered-feedforward neural network which consists of two input nodes, two hidden nodes and one output node. In Figure 1, right-upper of the matrix '1' means a connection is established, '0' means not connected between the two index nodes. Left-lower of the matrix denotes the weight value between the two index nodes.

The output of ENN is as follows

$$out_k = f\left(\sum_j w_{jk} f\left(\sum_i w_{ij} x_i\right)\right) \qquad (4)$$

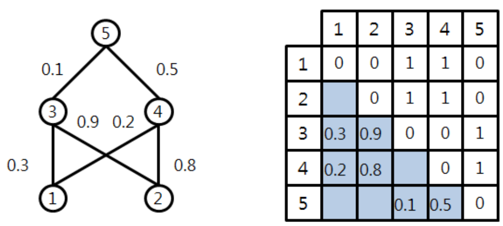


Figure 1. Matrix based encoding for Evolving Artificial Neural Network

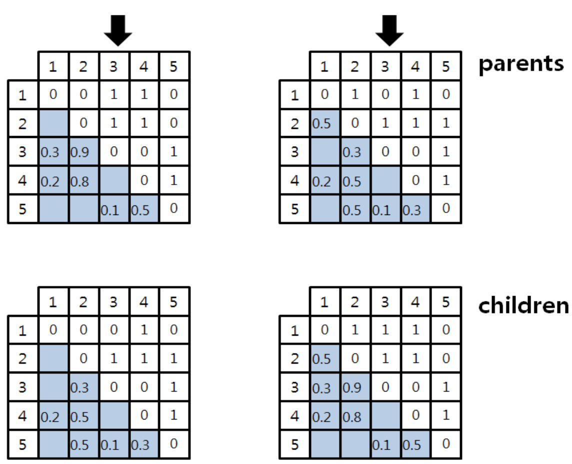


Figure 2. Crossover (the arrow denotes the crossover point)

Where x_i is input value, $f(x)$ is an activation function for each node. i , j , k denote the number of nodes of input layer, hidden layer, and output layer, respectively. To optimize the weights, it uses genetic operators such as crossover, selection and mutation. The operation of crossover and mutation modifies a weight value. The mutation operation perturbs the weight by using a random function. Figure 2 shows the crossover operation. The selection operation yields the diversity of population.

2.3 Back-propagation Neural Network

Back-propagation (BP) neural network is widely used in pattern classification since it is simple and its generalization ability is robust. Our BP architecture consists of one input layer, one hidden layer and one output layer called fully linked multiple layered perceptron. Training rule follows the Least Squared Method (LSM) and the update of weights is as follows

$$w(n) = w(n-1) + \alpha\frac{\partial E}{\partial w}, \qquad E_i = \sum_j \frac{1}{2}(t_{ij} - o_{ij})^2 \qquad (5)$$

Where W is a weight value, n denotes weight update iternation, α and E denote learning rate and a cost function for the learning. t and o denote target value and jth output of neural network at ith sample data.

3. Experimental Results

This section shows experimental results between back-propagation neural network and evolving neural network with PCA. We used six UCI data sets for classification and used PCA for data reduction. The description of data sets is shown in Table I. 70% of data was used for training and remaining are test sets. PCA axes are obtained from the training set. Each neural network uses a sigmoid function. The number of nodes of hidden layer is the same to the number of nodes of input layer for two neural networks, the number of nodes of output layer is the same to the number of data class.

In ENN population size, crossover rate and mutation rate are 100, 0.08 and 0.01. ENN used roulette wheel selection. In back-propagation, learning rate is 0.9. Each neural network updates the weights for 1000 epochs and 1 epoch means the total iteration of training data.

Table II shows experimental results of whole data sets with 10 runs and Figure 3 shows the comparison of accuracy between back-propagation neural network and evolving neural network with different dimension reduction rate. The x-axis of Figure 3 is dimension of searching space given data. The term of searching dimension denotes the searching space for the optimization solution. Two neural networks has the same searching space as

Searching Dimension = data dimension^2 + *data dimension* x *the number of class.*

Figure 4 shows that the classification boundary of ENN(a) and BPNN(b) in the diabetes data set which is 75% reduced by PCA, and the boundary of ENN is more appropriate than BPNN. The axes are two dimensional input values for neural networks.

Data Set	The number of data	Data dimension	Class
Breast Cancer	699	9	2
Australian	690	14	2
Soybean Large	307	35	19
Diabetes	768	8	2
Iris	150	4	3
Glass	214	9	6

Table I. The description of data sets

4. Conclusion

This paper presents the comparison of classification by ENN and BPNN with PCA. BPNN outperforms ENN in high dimension and ENN shows better accuracy rate than BPNN in low dimension. BPNN has good generalization ability in high dimension while ENN struggles for searching the solution in the large dimension problem. However, ENN shows better generalization than BPNN in low dimension. In 75% reduced dimension(notably below 28 searching dimension), ENN outperforms BPNN in which original data dimension is low. It implies that when ENN searches for the solution in high dimensional space, it faces with exponentially increasing of search space. In BPNN, data reduction with the loss of information shows low accuracy when data reduction is higher.

5. Acknowledgement

This work was supported by Standard Technology Advancement Program from Ministry of Knowledge Economy, Korea.

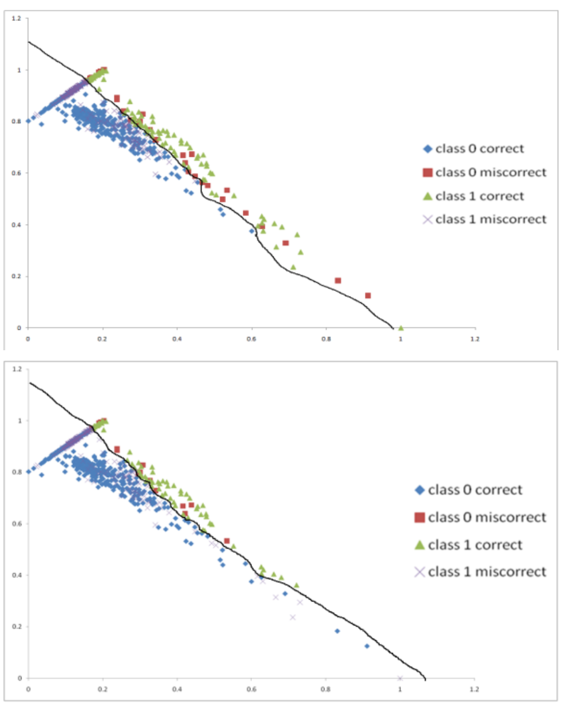


Figure 4. The classification boundary between back-propagation neural network and evolving neural network

No dimension reduction

Data Set(# of dimension)	ENN	BPNN
Breast Cancer(9)	98.53±0.37	98.04±0.81
Australian(14)	84.08±1.57	81.00±1.31
Soybean Large(35)	34.94±6.82	85.65±1.59
Diabetes(8)	77.57±2.20	77.96±1.42
Iris(4)	92.26±2.00	95.55±0.0
Glass(9)	54.00±7.73	48.30±4.57

25% reduction

Data Set(# of dimension)	ENN	BPNN
Breast Cancer(7)	98.58±0.34	98.14±0.29
Australian(10)	73.12±2.06	79.13±1.10
Soybean Large(27)	29.24±4.53	88.47±1.47
Diabetes(6)	73.46±2.45	75.71±0.78
Iris(3)	87.11±7.42	97.77±0.0
Glass(7)	51.53±7.91	50.61±1.74

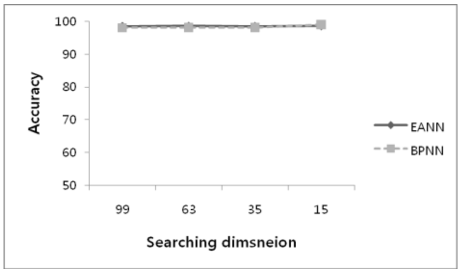
50% reduction

Data Set(# of dimension)	ENN	BPNN
Breast Cancer(5)	98.53±0.30	98.09±0.59
Australian(7)	76.92±1.56	81.15±0.19
Soybean Large(18)	26.45±6.19	82.71±3.55
Diabetes(4)	74.45±2.18	74.37±1.09
Iris(2)	93.11±4.60	97.77±0.0
Glass(5)	48.15±5.50	51.84±3.15

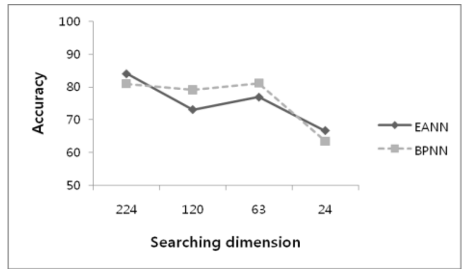
75% reduction

Data Set(# of dimension)	ENN	BPNN
Breast Cancer(3)	98.58±0.14	98.97±0.14
Australian(4)	66.68±2.51	63.46±0.0
Soybean Large(11)	29.56±3.37	72.29±2.19
Diabetes(2)	75.45±3.40	73.76±0.21
Iris(1)	78.44±5.53	71.11±0.0
Glass(3)	49.07±8.21	38.76±9.30

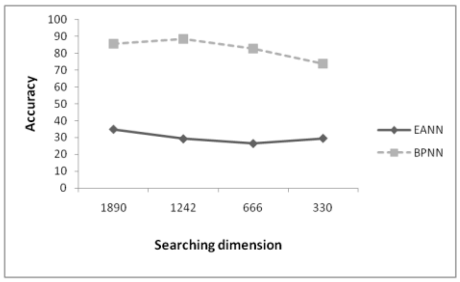
Table II. The accuracy of data set with different dimension reduction rate (average of 10 runs)



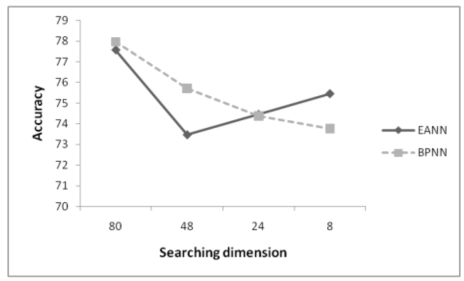
Breast Cancer



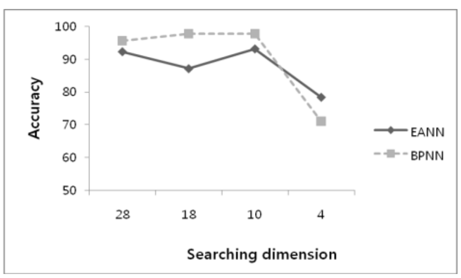
Australian



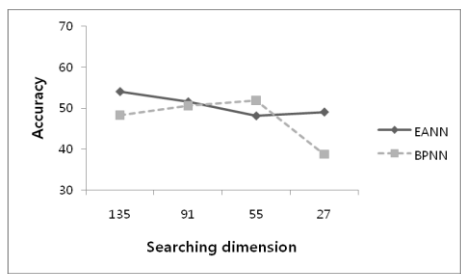
Soybean Large



Diabetes



Iris



Glass

Figure 3. The accuracy comparison between back-propagation neural network and evolving neural network

6. REFERENCES

[1] Xin, Y. 1999. Evolving artificial neural networks, *Proceedings of the IEEE* , vol. 87, no. 9, pp. 1423-1447, 1999

[2] Ajith, A. 2004. Meta learning evolutionary artificial neural networks, NeuroComputing, vol. 56, pp. 1-38, 2004

[3] Kim, K. –J., and Cho, S.-B. 2008. A genetic algorithm approach to optimal ensemble classifiers for DNA microarray data analysis," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 3, pp. 377-388, 2008

[4] Randall, S. S., Robert, E. D.2000. Reliable classification using neural networks: a genetic algorithm and backpropagation comparison, Decision Support Systems, Springer, vol. 30 No. 1, pp. 11-20, 2000

[5] Kim, K.-J. and Cho, S.-B.2009. Evaluation of Distances for Speciated Evolutionary Neural Network in Pattern Classification Problem, Proceedings of the 16th International Conference on Neural Information Proceeding, LNCS, vol. 5864, pp. 630-637, 2009

[6] Bruno, S. et al.1998. Fitness Sharing and Niching Methods Revisited, IEEE Transactions on Evolutionary Computation , vol.2, no.3, 1998

[7] Simon, H.1998. Neural Networks 2/E: A comprehensive foundation, Prentice Hall, pp. 161-175, 1998

[8] Christopher, M. B.1996. Neural Networks for Pattern Recognition, pp. 7-9, 448-450, Oxford, 1996

[9] Javed, K. et al.2001. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, Nature Medicine 7, pp. 673-679, 2001

[10] Mayank, A., Himanshu, A., Nikunj, J. and Manish, K.2010. Face recognition using principle component analysis, eigneface and neural network, Signal Acquisition and Processing International conference on, pp. 310-314, 2010

[11] Francisco, C. et al.2007. Principal component analysis in ECG signal processing, EURASIP Journal on Applied Signal Processing, no.1, 2007

[12] Hervé A. et al.2010. Principal Component Analysis, Wiley Interdisciplinary Reviews : Computational Statistics , vol. 2, no. 4, pp. 433-459, 2010