Initialization of the Nguyen-widrow and Kohonen Algorithm on the Backpropagation Method in the Classifying Process of Temperature Data in Medan

M R Wayahdi¹, M Zarlis², P H Putra³

Department of Information Technology, Faculty of Computer Science and Information Technology, University of Sumatera Utara, Medan 20155, Indonesia

¹muhammadrhifkywayahdi@gmail.com, ²m.zarlis@yahoo.com, ³purwahasanputra21@gmail.com

Abstract. In this paper, we propose the initialization of the Nguyen-widrow and Kohonen algorithm on the Backpropagation Neural Network in the classification of temperature in Medan. Initialization of Nguyen-widrow and Kohonen weight in Backpropagation could accelerate the training process of temperature data compared to Backpropagation with random weight. The experiment reaches target error 0.007 at 30 epoch. The result of testing show that the initialization of Nguyen-widrow and Kohonen weight in Backpropagation could recognize the test data reaches 96.52% accuracy.

Keyword: Artificial Neural Network, Backpropagation, Nguyen-widrow, Kohonen

1. Introduction

Air temperature is the average kinetic energy size of molecular movement. Temperature is a non linear condition in 24 hours can change. Selection of the appropriate method in determining level of the temperature of a region is the activities that are recently done by research. Artificial Neural Networks has matured to a great extent over a past few years. Neural network provides methodology of solving highly non-linear problems. Inspired by brain as ANN is interconnection of highly non-linear neuron. The neuron are connected to each other via link. This network is trained using Backpropagation algorithm which follows Gradient Descent Method [1].

Nguyen-widrow is used a method for initialization of the weights of neural networks to reduce training time [2], with the addition of the Nguyen-widrow method in determining the weight of the data generated which will be adjusted to its weight [3]. While the Kohonen Self-Organizing Map is a neural network-based clustering approach where it assigns input data to a series of partitions in its output neurons layer by the similarity of their vectors expression to reference vectors or weights that are defined for each partition [4], Kohonen can be trained in a short period of time with a few optimization techniques such as “winning” neurons search scope limit [5]. In this research, the writer is going to optimize Backpropagation method by adding Kohonen and Nguyen-widrow algorithm in process of training of the air temperature in Medan. The writer wants to know what Kohonen and Nguyen-widrow algorithm can reduce training time in recognizing the pattern of temperature data provided.
2. Related Research
Neupane N and Shakya S explained the neural networks can characterize examples, and consequently can be utilized as a part of intrusion detection system for attack classification. Backpropagation calculations are used well for neural network preparation for promising elements [6]. Kharola M and Kumar D used the training process can be done quickly. The result are more accurate to predict future weather when the number of iterations increases [7].

Andayani U et al. explains in the advanced phase (feed forward), the initial weighting method Nguyen-widrow able to give values close to target error, which affects the value of renewal weight to the backward phase (backward feed). At the backward phase, adaptive learning rate parameter capable of affecting the value of the weighting and reduced the number of iteration which that mean to get high performance with the built network was optimal [3]. Mishra K, et al. explains after using Nguyen-widrow in multilayer feed forward artificial neural network with conjugate gradient it improve execution time [2]. Khvorostukhina E et al. in their research modifies Kohonen’s learning so it can significantly reduce time grouping of objects and can be recommended for effective user to analysis a large amount of information [5]. Prokofiev A O et al. explains that since the Kohonen self-organizing maps are a tool for clustering, the experiment on clustering the full-scale results of stochastic transformations takes place [8].

3. Research Method
3.1. Artificial Neural Network
Artificial neural network is a technique that inspired by biological neuron model. In artificial neural network numbers of highly non-linear neurons are interconnected forming a network. The network consists of three layers: input, hidden, output. These neurons are connects by links which consists of weight, weights are the connection strength which exists between the neurons in the network. Basically ANN is the system that receives the input, process the data and then gives output with respect to input. Multilayer neural networks consist of input layers, one or more hidden layers, and output layers [1].

Artificial neural networks are relatively similar with electronic networks of "neurons" which is assumed based on the neural structure of the brain. For example the errors which obtained from initial classification of the first record is fed back into the network, and then modify the networks algorithm for the second time around, and so on for many iterations [2].

3.2. Backpropagation
Backpropagation is the one of the most popular algorithm which utilized in neural network framework. The Backpropagation need training to learn calculation process could be completed through propagation and weight update process [6]. Backpropagation algorithm consists of two passing processes: pass forward and pass backward. Pass forward calculates the output network and compares this with actual output to obtain error. Pass backward propagates the error back to each neuron in the previous layers to update the weighted matrices in order to minimize the error. A cycle of one pass forward and one pass backward over the network is known as one epoch [9].

The steps in the Backpropagation algorithm as follows:

a. Initialize weight (take a fairly small random value).

b. Forward propagation

1) Each input unit (X_i, i=1,2,3,…,n) receives the signal x_i and forward the signal to all unit in the hidden layer.

2) Each hidden unit (Z_j, j=1,2,3,…,p) sums the weight of the input signal, shown the equation (1).

\[ z_{jn} = v_{nj} + \sum_{i=1}^{n} x_i v_{ij} \]  

(1)

And apply the activated function to calculate its output signal, indicated by equation (2).

\[ z_j = f(z_{inj}) \]  

(2)
The activated function used is the sigmoid function, then sends the signal to all output units.

3) Each unit of output \( Y_k, k=1,2,3,\ldots,m \) sums the weight of the input signal, indicated by equation (3).

\[
y_{\text{in}k} = w_{2k} + \sum_{i=1}^{p} z_i w_{jk}
\]  

(3)

And apply the activated function to calculate its output signal, indicated by equation (4).

\[
y_k = f(y_{\text{in}k})
\]  

(4)

c. Back propagation

1) Each unit of output \( Y_k, k=1,2,3,\ldots,m \) receives the suitable pattern of target with the input pattern training, then calculates error, indicated by equation (5).

\[
\delta_k = (t_k - y_k)f'(y_{\text{in}k})
\]  

(5)

\( f' \) is a derivation of the activated function.

Then calculate the correlated weight, shown by equation (6).

\[
\Delta w_{jk} = \alpha \delta_k z_j
\]  

(6)

And count the bias correction, shown by equation (7).

\[
\Delta w_{bk} = \alpha \delta_k
\]  

(7)

All sending \( \delta_k \) to the units of the rightest layer.

2) Each hidden unit \( Z_j, j=1,2,3,\ldots,p \) adds its input delta (from units that are in the right layer), shown by equation (8).

\[
\delta_{\text{in}j} = \sum_{k=1}^{m} \delta_k w_{jk}
\]  

(8)

To calculate error information, multiply this value with an instance of its activated function, shown by equation (9).

\[
\delta_j = \delta_{\text{in}j} f'(z_{\text{in}j})
\]  

(9)

Then calculate the weight correction, indicated by equation (10).

\[
\Delta w_{ij} = \alpha \delta_j x_i
\]  

(10)

After that, calculate also the bias correction, shown by equation (11).

\[
\Delta \theta_j = \alpha \delta_j
\]  

(11)

d. Stage of changing weight and bias.

1) Each unit of output \( Y_k, k=1,2,3,\ldots,m \) are done the change of weight and bias \( j=0,1,2,\ldots,p \), shown by equation (12).

\[
w_{jk}(\text{new}) = w_{jk}(\text{old}) + \Delta w_{jk}
\]  

(12)

Each hidden unit \( Z_i, i=1,2,3,\ldots,p \) is done the changing and bias \( i=0,1,2,\ldots,n \), shown by equation (13).

\[
\nu_{ij}(\text{new}) = \nu_{ij}(\text{old}) + \Delta \nu_{ij}
\]  

(13)

2) The test of condition stops.

3.3. Nguyen-widrow

Nguyen-widrow is a method for initialization of the weights of neural networks to reduce training time. In weight initialization, some small number of random values for weight is assigned for Backpropagation network. If the network using large weight in initial weight, the input signals to each hidden or output unit will fall in saturation region where the derivative of sigmoid has very small value \( f'(\text{net}) = 0 \). Whereas, if weights are too small, the net input to hidden unit or output unit will reach zero point cause learning process is slow.

In order to obtain the best result, the initial weights and biases are set to random numbers between -0.5 and +0.5 or -1 and +1. Thus initial weights and biases is executed randomly so that using it to improve weight initialization method [2].
The steps in the Nguyen-widrow algorithm as follows:

a. Set:
   - \( n \) = number of input units
   - \( p \) = number of hidden units
   - \( \beta = \text{scale factor} = 0.7(p)^{1/n} = 0.7 \sqrt[p]{p} \)

b. For each hidden unit (\( j=1,\ldots,p \)), does step (c) – (f)

c. For \( i=1,\ldots,n \) (all input units), \( v_{i,j}(\text{old}) = \text{random number number between -0.5 and 0.5} \)

d. Calculate the value \( ||v_{i,j}(\text{old})|| \)

e. Reinitializing the weights of the input unit (\( i=1,\ldots,n \))

f. Bias used as initialization:
   - \( v_{o,j} \) = random numbers between \( -\beta \) and \( \beta \).

3.4. Kohonen

Kohonen self-organizing maps is a separate class of neural networks used for a variety of tasks such as clustering, classification and their derivatives [5]. This algorithm is also capable of processing the high dimension data because it is designed to group data into clusters that exhibited some similarities. Each group with similar features is projected onto the same node on the map. Otherwise, the dissimilarity increases with the distance that separates two objects on the map. Generally, in KSOM, the distance is calculated using the Euclidean Distance (ED) and it is commonly used to measure the distance between the input and output nodes in the network [4][10].

Training Kohonen algorithm is given below:

a. Select initial weights \( W_{i,j} \) random values from input vector range and learning rate \( \lambda \in [0,1] \).

b. Apply step 3-7, upto stopping criteria is false, stopping criteria may be number of iteration or learning parameter is sufficiently small (say \( \epsilon \)).

c. Apply euclidean measure from input vector and weight vector, for \( j=1,\ldots, m \).

d. Calculate winning node say index \( J \), so that \( D(J) \) is minimum.

e. \( J \) within a specified neighbourhood of \( j \) and for all \( i \), calculate new weights as in equation.

f. Update new weight using:
   - \( W_{i,j}(\text{new}) = (1 - \lambda) W_{i,j}(\text{old}) + \lambda x_i \) \hspace{1cm} (15)

g. Update learning rate \( \lambda \) using:
   - \( \lambda(t + 1) = 0.5 \lambda(t) \) \hspace{1cm} (16)

h. Calculate error and test stopping criteria of network.

4. Discussion

In this research, author using two data for the classification of temperature, namely training data and testing data. The data for training is used is data from 1998 to 2012 and data for testing is used is data from 2013 to 2017. Network architecture uses 8 input layers, 6 hidden layers, and 1 output layer.

Firstly, data is normalize to obtain a good data. Normalization is done for output network according to the activated function used. The data is normalized within interval \([0, 1]\). Transformation of data is done at smaller intervals \([0.1; 0.8]\), shown by equation (17).

\[
x' = \frac{0.5(x - a)}{b - a} + 0.1
\]

where \( a \) is the minimum data, \( b \) is the maximum data, \( x \) is the data to be normalized, and \( x' \) is data that has been transformed.

After get the normalized data, data through the training process to recognize the given data pattern. Training is perform with using several different target errors. Then, initialization of weights is also use
to begin the training process. The author propose combination of the Nguyen-widrow and Kohonen algorithms to initialize the weights on the Backpropagation method in order to recognize the patterns of data to be trained. The results of the training can be seen in Table 1.

Table 1. Training Result

| #   | Error (epoch) |
|-----|---------------|
|     | 0.02  | 0.01  | 0.0095 | 0.009  | 0.0085 | 0.008  | 0.0075 | 0.007  |
| B   | 0.02   | 0.01   | 0.0095 | 0.009  | 0.0085 | 0.008  | 0.0075 | 0.007  |
| B + K | 0.013  | 0.011  | 0.008  | 0.013  | 0.011  | 0.008  | 0.010  | 0.010  |
| B + N | 0.015  | 0.013  | 0.009  | 0.015  | 0.013  | 0.009  | 0.011  | 0.011  |
| B + K + N | 0.016  | 0.010  | 0.007  | 0.016  | 0.010  | 0.007  | 0.008  | 0.008  |

B = Backpropagation  
K = Kohonen  
N = Nguyen-widrow

In table 1 we can see the results of training with weight initialization combination Nguyen-Widrow and Kohonen in Backpropagation more quickly reach the target initialization error compared to the Backpropagation training with random weight initialization. Quadratic error reduction in Backpropagation method training with target error 0.007 can be seen in figure 1.

Figure 1. Decrease Error Value

In Figure 1 we can see the decrease of error value with initialization of Nguyen-widrow weight and Kohonen in Backpropagation can reach target error of 0.007 at 30 epoch, whereas Backpropagation with random weight reach target error 0.007 at 55 epoch. Then data were analyzed using Backpropagation random weight and Backpropagation with Nguyen-widrow and Kohonen initialization. The results of the test can be seen in Table 2.
Table 2. Testing Result

| #     | Error (%) |
|-------|-----------|
|       | 0.02 | 0.01 | 0.0095 | 0.009 | 0.0085 | 0.008 | 0.0075 | 0.007 |
| B     | 94.27 | 95.32 | 95.42   | 95.63 | 95.80   | 95.88 | 96.27   | 96.38 |
| B + K + N | 94.30 | 95.40 | 95.62   | 95.75 | 95.87   | 96.12 | 96.28   | 96.52 |

B = Backpropagation
K = Kohonen
N = Nguyen-widrow

In Table 2 we can see the results with the initialization of the Nguyen-widrow and Kohonen combination in the Backpropagation better than the random weight initialization. Backpropagation with Nguyen-widrow and Kohonen initial weights can recognize the audit data reaches 96.52% accuracy.

5. Conclusion
In this study, initialization of the Nguyen-widrow and Kohonen weights in Backpropagation which the author proposes that it can accelerate the training process temperature data, with a target error of 0.007 data identified at 30 epoch, while Backpropagation with weight of random data is identified at 55 epoch. And the testing process of the temperature data with initialization of the weight of Nguyen-widrow and Kohonen can achieve a higher accuracy, with a target error of 0.007 data identified with 96.52% accuracy, while Backpropagation with weight of random data is identified with 96.38% accuracy. This shows the initialization of Nguyen-widrow and Kohonen weights in Backpropagation is better in terms of training speed and accuracy than Backpropagation with random weight.

References
[1] Naik A R and Pathan S K 2012 Weather Classification and Forecasting using Backpropagation Feed-forward Neural Network. IJSRP, Vol. 2, Issue. 12, ISSN 2250-3153, pp 1-5.
[2] Mishra K et al. 2014 Image Compression using Multilayer Feed Forward Artificial Neural Network with Nguyen Widrow Weight Initialization Method. IJETAE, Vol. 4, Issue 4, ISSN 2250-2450, ISO 9001:2008 Certified Journal, pp 475-480.
[3] Andayani U et al. 2017 Optimization Backpropagation Algorithm Based on Nguyen-widrow Adaptive Weight and Adaptive Learning Rate. ICIEA, pp 363-367.
[4] Ahmad A and Ismail M N 2017 Clustering the Imbalanced Datasets using Modified Kohonen Self-Organizing Map (KSOM). Computing Conference 2017, pp 751-755.
[5] Khvorostukhina E et al. 2017 Performance Improvements of a Kohonen Self-Organizing Training Algorithm. pp 456-458.
[6] Neupane N and Shakya S 2017 Comparative Analysis of Backpropagation Algorithm Variants for Network Intrusion Detection. ICCCA, pp 726-729.
[7] Kharola M and Kumar D 2014 Efficient Weather Prediction by Backpropagation Algorithm. IOSR-JCE, Vol. 16, Issue 3, Ver. IV, e-ISSN: 2278-0661, p-ISSN: 2278-8727, pp 55-58.
[8] Prokofiev A O et al. 2018 A Method of Cryptostability Analysis of Stochastic Transformations, Based on the Kohonen Self-Organizing Maps. pp 351-354.
[9] Latumakulita L A and Usagawa T 2017 A Combination of Backpropagation Neural Network on Fuzzy Inference System Approach in Indonesia Scholarship Selection Process. ICNC-FSKD, pp 1309-1314.
[10] Singh U P et al. 2017 Kohonen Neural Network Model Reference for Nonlinear Discrete Time Systems. IEEE-CICT, pp 1-5.