NST PROCEEDINGS
OPEN ACCESS

Conference Paper

Analysis of Supervised Learning Methods on Artificial Neural Networks

[1] Fetty Tri Anggraeny *, [1]Intan Yuniar Purbasari, [1] Radical Rakhman Wahid

[1] Department of Informatics System, Faculty of Computer Science, Universitas Pembangunan Nasional "Veteran" Surabaya, East Java

Abstract

Artificial intelligence applications are now highly developed, where hardware is given the ability to think and act like humans. One method for building artificial intelligence applications is artificial neural networks (ANN). ANN is currently highly developed in areas of application that start as diverse as classification, prediction, games, robots, IoT, object/sound recognition, and so on. Unlike the human brain, which is designed to solve many problems, ANN can only solve a specific problem. To make ANN intelligent, learning methods are needed to train ANN to be able to solve certain problems with a good degree of accuracy. There are 3 learning methods commonly used to make ANN intelligent, are perceptron, backpropagation, and extreme learning method. Perceptron is the simplest learning method for the simplest ANN structure. Backpropagation was found to answer the weaknesses of perceptron which cannot solve nonlinear problems. And finally ELM exists to overcome the problem of long computational time in perceptron and backpropagation learning. This research will conduct an analysis of the three ANN learning methods using several problems. Problems used include AND logic, OR logic, XOR logic, iris datasets and MNIST digit handwritten digits. Because the ANN learning method is based on random parameters, each case will be trained 10 times for each method. After testing it is concluded that ELM has a very good speed with a pretty good degree of accuracy.

Keywords: analysis, supervised learning, artificial neural networks

Introduction

Artificial intelligence applications are now highly developed, where there is hardware that is compatible with human capabilities and needs. In making this application, an artificial intelligence method is needed, one of which is an artificial neural network (ANN). ANN is currently very developed to overcome weaknesses, including development in the training process, combining with optimization methods, and application fields that start as diverse as classification, games, robots, IoT, object / sound discovery, and so on.

Before ANN can be used to solve a problem, the best structure of ANN structure configuration must be done by going through the training process. There are 2 types of learning, namely supervised and unsupervised learning. A supervised learning algorithm takes a known set of input dataset and its known responses to the data (output) to learn the regression/classification model. A learning algorithm then trains

Corespondence Author
Email Address: fettyanggraeny.if@upnjatim.ac.id

How to cite this article: Anggraeny, F. T., Purbasari, I. Y. and Wahid, R. R. (2019). Analysis of supervised methods on artificial neural networks. *4th International Seminar of Research Month*. NST Proceedings. pages 125-134.doi: 10.11594/nstp.2019.0418.

a model to generate a prediction for the response to new data or the test dataset. Supervised learning uses classification algorithms and regression techniques to develop predictive models (Rao & Gudivada, 2018). Unsupervised learning studies how systems can learn to represent particular input pat-terns in a way that reflects the statistical structure of the overall collection of input pat-terns. By contrast with Supervised Learning, there are no explicit target outputs or environmental evaluations associated with each input (Dayan et all., 1999). Three supervised learning methods that will be discussed in this study include perceptron, backpropagation, and extreme learning machine (ELM). Some papers that discuss the perceptron method can be seen in (Stephen, 1990) and (Freund and Schapire, 1999). Some papers that discuss the backpropagation method can be seen in (Sukhbaatar & Fergus 2016), (Cilimkovic, 2015; Anggraeny, 2009; Anggraeny & Purbasari, 2015; Anggraeny et all., 2018). The backpropagation training process takes quite a long time and the process is quite high, so many studies that optimize the training process of backpropagation ANN (Ren et all., 2014; Das et all., 2014; Purbasari & Anggraeny, 2015). In addition, several studies also modified backpropagation ANN (Lin et all., 2015; Rodger, 2014; Huang et all., 2004) introduced the Extreme Machine Learning Method (ELM) which overcomes the main problem in the backpropagation method. Some implementations of this method include (Tang et all., 2016; Wan et all., 2014; Anggraeny & Purbasari, 2016).

As previously stated, there are many variations in the application of ANN, various methods of the training process, so this study will compare the performance of the three methods. Each method will be tested on 2 types of problems, namely linear and nonlinear problems.

Research Method

There are 3 methods of supervised learning artificial neural networks used in this study, namely perceptron, backpropagation and Extreme Learning Machine (ELM). Each will be explained as follows.

Neural Network

In artificial neural networks, neurons will be collected in layers (layers) called neuron layers (neuron layers). Information given to the neural network will be propagated layer to layer, starting from the input layer to the output layer through another layer, known as the hidden layer. The input layer is analogous to sensory nerve cells in biological nerve tissue, for example nerve cells in the skin as a touch sensor, on the nose odor sensor, on the taste sensor tongue, on the light sensor eye, on the sound sensor ear. This layer serves to receive input data from outside and forwarded to other neurons in the next layer in the artificial neural network. Output layer such as motor nerve cells in biological nerve cells. This layer functions to distribute the output signals from network processing. This layer is analogous to the connecting nerve cells of sensory nerve cells and motor nerve cells in the biological nerve layer. This layer is able to improve the ability of artificial neural networks in solving problems. The amount of this layer in biological nerve tissue is very much, for example the data that enters through the sensory nerve cells of the skin in the hand will be transferred cells to cells to the central cell (brain) and then sent back cells to cells to motor nerve cells.

Neurons in Artificial Neural Networks are the same as neurons in living things which are nerve cells that will transform / change the information received into information passed on to other neurons. The electrochemical process that occurs in neurons of living things is replaced by the activation function of human neurons. An illustration of human neural networks compared to artificial neural networks can be seen in Figure-1.

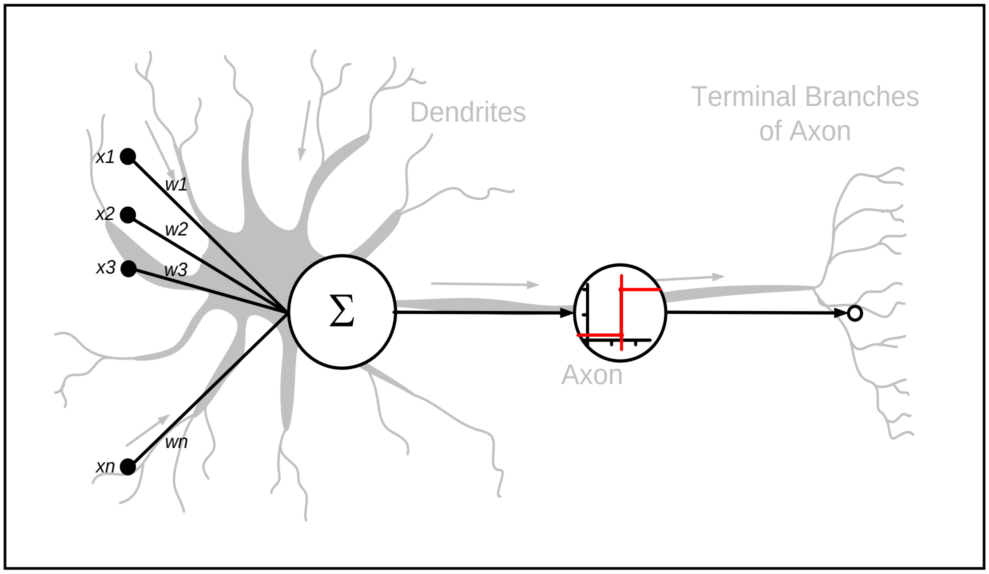


Figure 1. Artificial Neural Network Neurons.

The simplest supervised learning ANN architecture is a single layer neural network, in addition there is a multilayer neural network architecture, with 1 hidden layer or n hidden layer. Pictures of each supervised learning artificial neural network architecture can be seen in Figure-2.

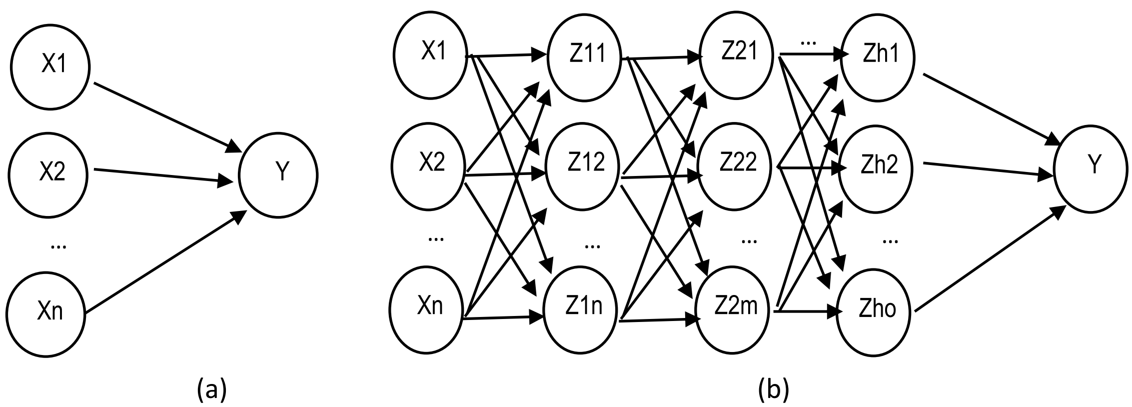


Figure 2. Artificial Neural Network Architecture, (a) single layer neural network, (b) multi layer perceptron neural network.

Perceptron

The perceptron training algorithm is used to train a single layer neural network (see Figure-2a). Perceptron is good for solving problems that can be separated by linear lines, namely problems where different data classes can be separated by straight lines. Figure-3 illustrates an example of data that can be solved using perceptron. It appears that there is a strict linear line that is used as a boundary between class A data (black dots) and class B data (gray dots).

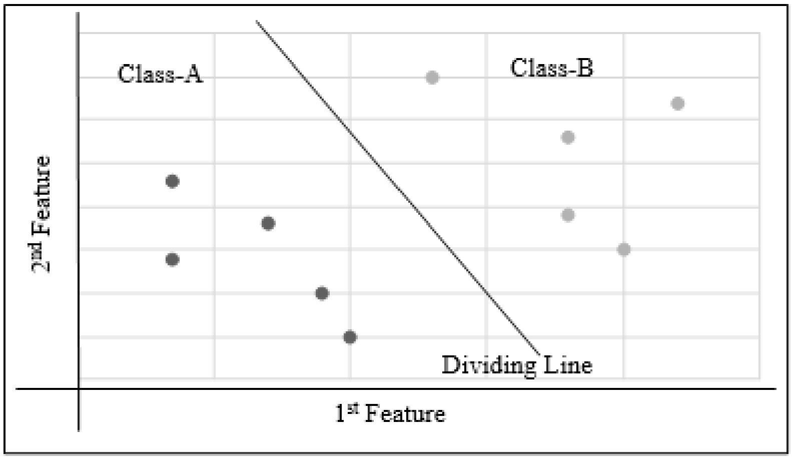


Figure 3. Linear Problem

The dividing line function between class A and class B has the following equation:

$$w_1x_1 + w_2x_2 + w_0 = 0 \quad (1)$$

So, the inequality of class A regions which lies below the dividing line function:

$$w_1x_1 + w_2x_2 < -w_0 \quad (2)$$

And the inequality of the class B area located above the dividing line function:

$$w_1x_1 + w_2x_2 \geq -w_0 \quad (3)$$

The learning process of artificial neural networks is basically used to find the weight value (w) that connects between layers in order to obtain test results with minimum misclassification (error). The learning process is done repeatedly to improve the weight value until it reaches the expected minimum error. The parameters that also play a role in the learning process are the learning rate parameters commonly symbolized by alpha (α). The perceptron learning algorithm is as follows:

Step 1. Prepare training data

Storage of training data into separate variables between feature variables and target variables, $k = 1,2,3,\cdots,n$.

Step 2. Initialization

a. If the network architecture has m input nodes and 1 output nodes, the initial weight value initialization can be given a value of 0 or random for each link between the input node and the output node, the weight value between the node-i in the input layer and node-j in the layer output (w_{ij})
b. Each output node is equipped with a bias value (w_{0j}) that can be assigned a random value
c. learning rate (α) is worth in the range [0,1]
d. maximum epoch (maxEpoch)
e. initial epoch is equal to 0 (epoch = 0)
f. the value of the stop condition is the same as FALSE

Step 3. As long as the stop condition is FALSE, start the k=1 data do:

a. Calculate the response of each output node (y_j), $j = 1,2,3,\cdots,q$

$$S_{kj} = \sum_{i=0}^{m} w_{ij} x_{ki} + w_{0j} \quad (4)$$

b. Apply activation function:

$$O_{kj} = f(y_{in_{kj}}) \quad (5)$$

c. Fix weight and bias:

$$w_{ij} = w_{ij} + \alpha(T_{kj} - O_{kj})x_{ki} \quad (6)$$

$$w_{0j} = w_{0j} + \alpha(T_{kj} - O_{kj}) \quad (7)$$

d. Change the value k = k + 1, and proceed to the next data by returning to Step 3.a

Step 4. Test the stopping condition by calculating the SSE

$$SSE = \sqrt{\sum_{k=1}^{n} (T_k - O_k)^2} \quad (8)$$

If the SSE value> threshold value and epoch value <MaxEpoch, then repeat Step 3, otherwise the stop condition is TRUE, and the training process is complete.

Backpropagation

Backpropagation is a supervised learning algorithm with multiple layers to change the weights associated with neurons in hidden layers (see Figure-2b). Learning methods on neural networks are called supervised if the expected outputs are known beforehand. Backpropagation learning is suitable for solving problems that cannot be separated by linear lines (nonlinear), ie problems where different data classes cannot be separated by one straight line. Figure-4 illustrates examples of data that can be solved using back propagation. It appears that there is no linear line that can be used as a boundary between class A data (black dots) and class B data (gray dots).

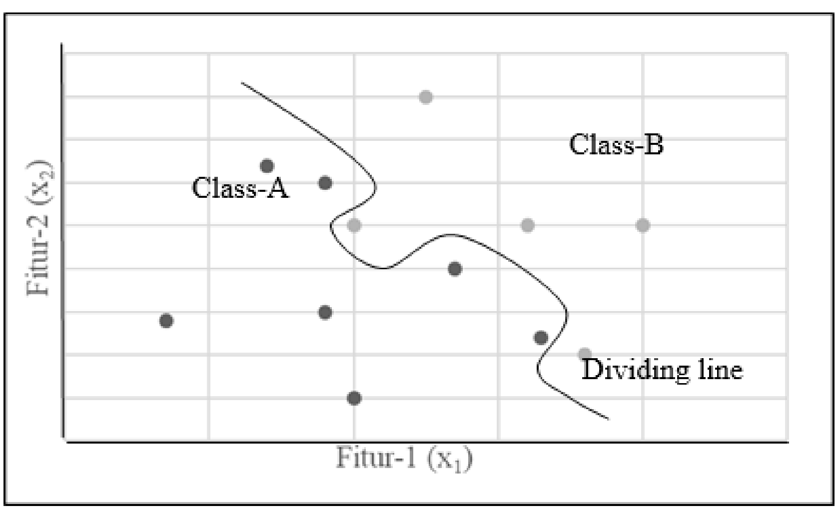


Figure 4. Nonlinear Problem

Artificial neural network with back propagation method consists of 2 stages, namely the forward stage and backward propagation stage. This stage is intended to make artificial neural networks become "intelligence" by entering some data that has been proven true. "intelligence" in this case is to configure the weight value so that the network can produce output that matches/approaches the target.

Step 1 Initialize the weight and bias values can be set with any number (random) Initialization learning rate, maximum literation and error tolerance.
Step 2 Do it while the stop condition is still not fulfilled, start the feed forward step.
Step 3 Each input unit (x_i, i = 1, ..., m) receives the input signal and spreads it to all hidden units.
Step 4 Each hidden unit (z_j, j = 1, ..., p) will calculate the input signals with their weights and biases.

$$z\_in_j = v_{0j} + \sum_{i=1}^{m} x_i v_{ij} \tag{9}$$

Then using the predetermined activation function, an output signal from the hidden unit is obtained.

$$z_j = f(z\_in_j) \tag{10}$$

Step 5 Each unit of output (y_k, k = 1,...,q) will count the signals from the hidden unit with their weights and biases.

$$y\_in_k = w_{0k} + \sum_{j=1}^{p} z_j w_{jk} \tag{11}$$

Then using the predetermined activation function the output signal from the output unit is obtained.

$$y_k = f(y\_in_k) \tag{12}$$

Step 6 Calculate the error correction factor (δo_k) at the output layer, and begin the preparation of the backward flow (bacpropagation) stage.

$$\delta o_k = (t_k - y_k)\, f'(y\_in_k) \tag{13}$$

Error correction factor is used to calculate error correction (ΔW_{jk}) to update W_{jk}.

$$\Delta W_{jk} = \alpha\, \delta o_k\, z_j \tag{14}$$

$$\Delta W_{0k} = \alpha\, \delta o_k \tag{15}$$

Step 7 Each hidden unit (Z_j, j = 1, ..., p) will calculate the weight sent by the m output unit. If the first iteration condition is the formula used.

$$\delta\_in_j = \sum_{k=1}^{m} \delta o_k w_{jk} \tag{16}$$

Then the results are multiplied by the derivation of the activation function to get the error correction factor.

$$\delta h_j = \delta\_in_j\, f'(z\_in_j) \tag{17}$$

$$\Delta V_{ij} = \alpha\, \delta_j\, x_i \tag{18}$$

$$V_{0j} = \alpha\, \delta_j \tag{19}$$

Step 8 Each output unit (Yk, k = 1, ..., m) will update the weight of each hidden unit.

$$W_{jk}(baru) = W_{jk}(lama) + \Delta W_{jk} \tag{20}$$

Likewise each hidden unit (Zj, j = 1, ..., p) will renew the weight of each input unit.

$$V_{ij}(baru) = V_{ij}(lama) + \Delta V_{ij} \tag{21}$$

Step 9 Check stop condition

Extreme Learning Machine

Extreme Learning Machine (ELM) is a new learning method for artificial neural networks discovered by Huang et al. (Huang et all., 2004). ELM is a learning method for a single layer feed forward network architecture that is able to solve problems that arise in backpropagation learning. As we have learned about backpropagation above. It is stated that to produce a good neural network structure requires many iterations. With more and more iterations, the longer the time needed for the learning phase. Backpropagation learning requires a longer time compared to ELM even though it uses the same artificial neural network structure. Backpropagation uses a learning method based on a gradient algorithm that runs very slowly and several parameters (weights) must be changed in each iteration until the stop condition is reached (Tang et all., 2015). The stop condition in back propagation learning can be based on 2 things: the maximum number of iterations and the minimum error value. Whereas ELM only requires 1 iteration to do learning on the specified neural network structure.

The ELM algorithm consists of 2 stages, namely random mapping features and linear completion parameters. The first step is to initialize random weights and biases at the hidden layer node to map input feature data to the hidden layer using a probability distribution algorithm. The second step is calculating the weight value that connects the hidden layer node to the output layer node by minimizing the output error.

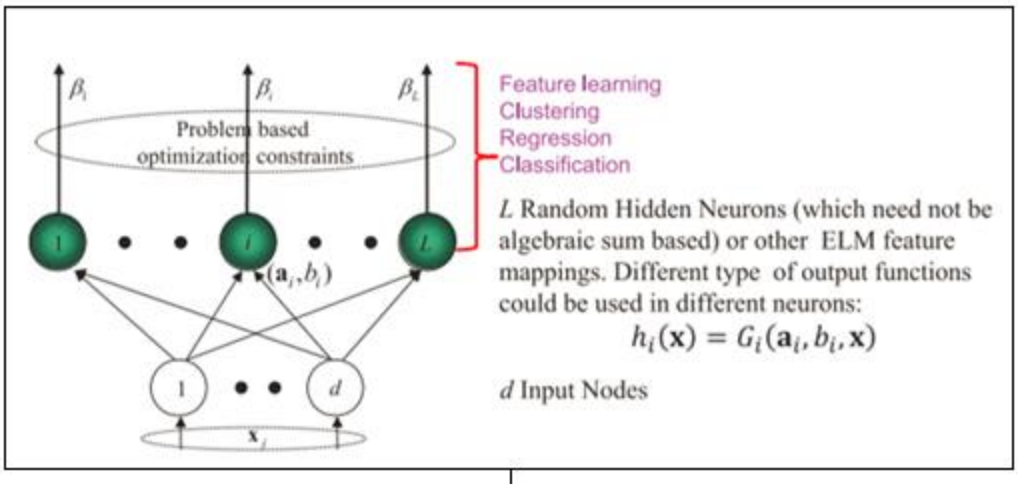


Figure 5. Structure of artificial neural networks for ELM learning models (Tang et all., 2015).

The structure of artificial neural networks using the ELM learning method is illustrated in Figure-5. An artificial neural network consisting of *d* input nodes in the input layer corresponds to the number of features used, *L* hidden nodes in the hidden layer, and *m* nodes in the output layer according to the number of problem classes.

Step 1 Mapping feature values to the hidden layer so as to produce a nonlinear ELM feature. This stage is done by calculating the output of each node in the hidden layer using the following formula:

$$h_i(x) = F(a_i, b_i, x), a_i \in R^d, b_i \in R \quad (22)$$

$$F(a_i, b_i, x) = F\left(\sum_{j=1}^{d} a_i . x_j + b_i\right) \quad (23)$$

Where $h_i(x)$ is the output of the *i*-node the hidden layer, a_i is the weight value from the input node to the *i*-node hidden, b_i is the value of the bias weight of the node in the hidden layer, and x_j the value of the input feature data. $F(z)$ is the activation function of each node in the hidden layer and each node can use a different activation function.

Step 2. Calculate the final output of the ANN structure using the following formula:

$$f_L = \sum_{i=1}^{L} \beta_i h_i(x) = h(x)\beta \qquad (24)$$

Where $\beta = [\beta_1, \ldots, \beta_L]^T$ is the weight between the *L*-hidden layer node and the *m*-output layer node, while $h(x) = [h_1(x), \ldots, h_L(x)]$ is an ELM nonlinear mapping feature.The weight connecting the hidden layer with the output layer is formed with the aim of minimizing errors.

$$min_{\beta \in R^{Lxm}} \|h\beta - T\| \qquad (25)$$

Where *h* is the hidden layer value that results from the process step 1, *T* is the target, and $\|.\|$ is Frobenius norm.

Dataset

There are 5 data used inthis research, the characteristic shown in Table-1, the first three is the simple one. The data divided into 2 problems, are linear and nonlinear problem. Linear problem means that the data can divided using straight lain, otherwise in nonlinear problem. Example of linear problem are AND logic and OR Logic, while XOR Logic, Iris, and MNIST digit are nonlinear preoblems. Detail of each dataset shown in Figure-6.

Table 1. Characteristics of the dataset

Problem	#Input	#Class	#Instance
AND Logic	2	2	4
OR Logic	2	2	4
XOR Logic	2	2	4
Iris Dataset	4	3	150
Digit Handwritten Dataset	784	10	42000

x_1	x_2	x_1 AND x_2
0	0	0
0	1	0
1	0	0
1	1	1

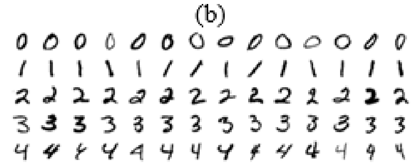
(a)

x_1	x_2	x_1 OR x_2
0	0	0
0	1	1
1	0	1
1	1	1

(b)

x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	1
1	1	0

(c)



(d)

Sepal Length	Sepal Width	Petal Length	Petal Width	Kelas
5.1	3.5	1.4	0.2	0
4.9	3	1.4	0.2	0
4.7	3.2	1.3	0.2	0
4.6	3.1	1.5	0.2	0

(e)

Figure 6. Dataset sample.

Result and Discussion

Because there are 3 kind of unsupervised learning methods in artificial neural network and 5 datasets, then we have 15 of artificial neural network structure. Each structure give 2 performance, accuracy and iteration number, see Table-2 and Table-3. Linear problem have good accuracy in perceptron or backpropagation, and quite good in ELM. Otherwise, nonlinear problem have good accuracy in backpropagation or ELM.

As explain in methodology, ELM did not need any iteration process, it only process once and done. In iteration number comparation, ELM is the winner. All learning method can solve linear problem, but it needs more iteration in backpropagation learning. And this phenomena also shown by nonlinear problem in perceptron, it needs many of iteration or difficult to achieve stopping condition.

Table 2. Accuracy performance.

Problems	Perceptron	Backpropagation	ELM
AND logic	100	100	75
OR logic	100	100	75
XOR logic	~ 50	100	50
Iris Dataset	51.11	91.11	100
Digit Handwritten	86,65	50	100

Table 3. Iteration Number Performance.

Problems	Perceptron	Backpropagation	ELM
AND logic	6	172	1
OR logic	4	408	1
XOR logic	~ (>1000)	1107	1
Iris Dataset	~(>1000)	4	1
Digit Handwritten	~(>1000)	99	1

Conclusion

The results of the study can be concluded that:

1. Perceptron good for solving linear problems and do not use it for nonlinear problems because it needs many iteration
2. Backpropagation good for solving nonlinear problems, because needs less iteration than solving linear problem
3. ELM good for both linear and nonlinear problems and good in running time (no iteration) and the performance is very good for large dataset than the simple one

Suggestion

For the best result, generate more ANN structure for each dataset and learning methods. And add more dataset in linear problems and nonlinear problem, so we can get conclude in general manner.

Acknowledgment

The author would like to thank University of Pembangunan Nasional "Veteran" Jawa Timur for the support and opportunities provided in the 2019 funding Research Program in Peningkatan Mutu Pembelajaran (PMP) scheme with title *Jaringan Saraf Tiruan dan Modifikasinya Menggunakan Supervised Learning*.

.

References

Anggraeny, F. T., Muttaqin, F., & Munir, M. S. (2018). Modeled early detection of pregnancy risk based on Poedji Rochjati score card using relief and neural network. *International Conference on Science and Technology (ICST)*: Denpasar.

Anggraeny, F.T, Purbasari, I.Y. (2015). Prediksi Kelompok Usia Manusia Berdasarkan Rasio Garis Keriput pada Area Dahi Menggunakan Citra Digital. *Seminar Nasional Pengembangan Aktual Teknologi Informasi (SENA BAKTI):* Surabaya,

Anggraeny, F.T., Purbasari, I.Y. (2016). Modification of Hidden Layer Weight in Extreme Learning Machine Using Gain Ratio. *MATEC Web of Conferences*, 58: 03010.

Anggraeny, F. (2009). Prediksi Prestasi Akademik Mahasiswa Dengan Metode Jaringan Saraf Tiruan. *Seminar Nasional LPPM UPN "Veteran" Jawa Timur*: Surabaya,

Cilimkovic, M. (2015). *Neural networks and back propagation algorithm*. Dublin: *Institute of Technology Blanchardstown*

Das, G., Pattnaik, P. K., & Padhy, S. K. (2014). Artificial neural network trained by particle swarm optimization for non-linear channel equalization. *Expert Systems with Applications*, 41(7), 3491-3496.

Dayan, P., Sahani, M. and Deback, G. (1999). *Unsupervised learning.* USA:The MIT encyclopedia of the cognitive sciences.

Freund, Y. and Schapire, R.E. (1999). Large margin classification using the perceptron algorithm. *Machine learning*, 37(3), pp. 277-296.

Huang, G.B., Zhu, Q.Y. and Siew, C.K. (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. *Neural networks,* 2: 985-990.

Lin, Y. Y., Chang, J. Y., & Lin, C. T. (2015). A TSK-type-based self-evolving compensatory interval type-2 fuzzy neural network (TSCIT2FNN) and its applications. *IEEE Transactions on Industrial Electronics,* 61(1),): 447-459.

Purbasari, I. Y., & Anggraeny, F. T. (2015). Perbandingan Algoritma Klasifikasi untuk Huruf Tulisan Tangan. *SCAN-Jurnal Teknologi Informasi dan Komunikasi*, 10(2): 55-62.

Rao, C.R. and Gudivada, V.N. (2018). *Computational Analysis and Understanding of Natural Languages: Principles, Methods and* Applications. Holland: Elsevier.

Ren, C., An, N., Wang, J., Li, L., Hu, B., & Shang, D. (2014). Optimal parameters selection for BP neural network based on particle swarm optimization: A case study of wind speed forecasting. *Knowledge-Based Systems,* 56: 226-239.

Rodger, J. A. (2014). A fuzzy nearest neighbor neural network statistical model for predicting demand for natural gas and energy cost savings in public buildings. *Expert Systems with Applications,* 41(4): 1813-1829.

Stephen, I. (1990). Perceptron-based learning algorithms. *IEEE Transactions on neural networks, 50(2):*. 179.

Sukhbaatar, S., & Fergus, R. (2016). Learning multiagent communication with backpropagation. *Advances in Neural Information Processing Systems*, 1: 2244-2252.

Tang, J., Deng, C. and Huang, G.B. (2015). Extreme learning machine for multilayer perceptron. *IEEE transactions on neural networks and learning systems,* 27(4): 809-921.

Tang, J., Deng, C., & Huang, G. B. (2016). Extreme learning machine for multilayer perceptron. *IEEE transactions on neural networks and learning systems*, 27(4): 809-821.

Wan, C., Xu, Z., Pinson, P., Dong, Z. Y., & Wong, K. P. (2014). Probabilistic forecasting of wind power generation using extreme learning machine. *IEEE Transactions on Power Systems,* 29(3): 1033-1044.