STUDY ON OPTIMIZATION OF ARTIFICIAL NEURAL NETWORK GENERALIZATION POWER BASED ON ARCHITECTURE

[1]MARY ANNE ROA, [2]PROCESO FERNANDEZ JR.

[1,2]Department of Information Systems and Computer Science, Ateneo de Manila University, Philippines
E-mail: [1]mary.roa@obf.ateneo.edu, [2]pfernandez@ateneo.edu

Abstract- Artificial Neural Networks (ANNs) have attracted considerable attention from researchers in many fields, including economics, medicine, data processing, robotics, quality control, optimization and security, and have been used to solve a wide range of problems. The discovery of numerous applications and promising benefits to a diverse field encourages studies in this area. Architecture of the network is one of the major challenge in the field of Artificial Neural Network. Hidden layers plays a vital role in the performance of ANN causing the problem of either underfitting or overfitting under certain circumstances. Overfitting is the phenomenon wherein the network fits the training data set so well that even noise of the training data are memorized causing performance to drop when it is tested against unknown data set. Underfitting, the opposite of over-fitting, occurs when the model is incapable of capturing the variability of the data and the resulting model will have suboptimal predicative ability. The structure of the network also has direct effect on training time and classification accuracy. Although there are some discussions in the literature of the impact of network structure on the performance of the ANN, there is no standard method or approach for selecting the optimum structure. In this paper, the relationship between the number of hidden layers, nodes (neurons) and the accuracy of the classification is investigated. The result of this study can be used as a guideline for the selection of design parameter in setting artificial neural networks.

Keywords- Artificial Neural Network, Network Architecture, Hidden Layer, Neurons.

I.INTRODUCTION

The field of computation has a lot to gain from neural networks. The capability of neural networks to learn by example makes them extremely adaptable and robust. A neural network typically involves a large number of interconnected processing nodes working together to solve a specific problem. These nodes receive information from neurons of the previous layer and pass them on to the next layer. Deciding the number of hidden layers and number of neurons in each hidden layer is a challenging issue in considering any problem. Figure 1 shows a neural network having three layers, called the input layer, hidden layer, and output layer. Each layer consists of one or more nodes, represented in this diagram by the small circles. The arrows between the nodes indicate the flow of information from one node to the next.

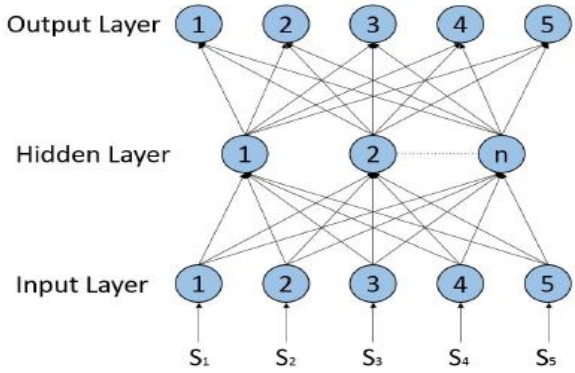


Fig.1. ANN Architecture

1.1. Input Layer

The Input layer is a layer which communicates with the external environment that presents a pattern to the neural network. The number of neurons comprising the input layer is equal to the number of features in the training data set. The nodes of the input layer are passive and do not modify the data. They receive a single value on their input, and duplicate the value to their multiple outputs [1].

1.2 Output Layer

The output layer is directly related to the type of work that the neural network is to perform. The number of output neurons is typically equal to the number of elements in the target vector - one output neuron for each group that input instance may be assigned into. The output layer can be configured to report a wide range of information such as yes/no or classification.

1.3 Hidden Layer

The hidden layer is an intermediate layer between the input layer and the output layer. Neural networks can have any number of layers, and any number of nodes per layer [1]. The actual processing is done via a system of weighted connections of nodes in this layer [2]. The hidden layers then link the processed information to the output layer.

Questions arise as to how many hidden layers has to be used and the number of neurons in each hidden layer as the ability of the neural network to provide useful data manipulation lies in the proper selection of the network parameters. If the data is linearly separable than there is no need to use hidden layer. But in case of problems which deals with arbitrary decision boundary then one has to use two or three hidden layer [3]. Also, the number of neurons that should be kept in each hidden layer need to be carefully selected. The generalization power of an ANN largely depends on the hidden layer architecture [4]. Generally, too many hidden neurons over-train

the network whist too few exhibits insufficient degrees of freedom [5,6]. Determining the most suitable architecture is very important in artificial neural networks. However, until today there is no exact solution for optimized solution on hidden layer architecture in multilayer artificial neural networks [7]. As such, many researches have been carried out to better understand the effect of hidden layer design in optimum network performance.

The power of a neural network depends on how well it generalizes on new data after training. Generalization is the ability of the neural network to interpolate and extrapolate data that it has not seen before [8]. The generalization capabilities of ANNs largely depends on their architecture. When there are too few neurons in the hidden layers as equated to the complexity of the problem then underfitting usually occurs. The number of hidden layers and the number of nodes in each hidden layer depend on the complexity of the problem [9].

Generally, a network with large number of hidden nodes is able to learn fast and avoids local minima as when a network has too many free parameters such as weights and/or nodes, local minima are more easily avoided [10, 11]. However too large a network yields more nodes, more weights and more layers than sometimes necessary; and thus results in unnecessary arithmetic calculations and high computation cost. Also if unnecessary more neurons are present in the network it is likely to overfit the training data and have poor generalization ability for the testing data [12]. Better generalization performance can be achieved by smaller networks but their training may require more effort since these networks may not have enough processing elements [13]. Moreover, from the implementation point of view, small networks only require limited resources whereas large networks may require otherwise. Both large and small networks exhibit certain advantages and disadvantages. The optimal architecture is a network that is large enough to learn the problem and is small enough to generalize well [14].

The study aims to comprehend the optimum internal structure determination for the neural networks. Various network structures were trained and tested in order to find out the relationship between the size of the neural network and the accuracy of the resulting classification.

II. DETAILS EXPERIMENTAL

The dataset used in the experiments is the Breast Cancer Wisconsin Dataset of Dr. William H. Wolberg available from the UCI Machine Learning Repository [15]. This dataset can be used to classify tumor as either benign or malignant depending on the characteristics of sample biopsies and is composed of 699 sample cases for which there are 9 attributes each (1. Clump thickness 2. Uniformity of cell size 3. Uniformity of cell shape 4. Marginal Adhesion 5. Single epithelial cell size 6. Bare nuclei 7. Bland chomatin 8. Normal nucleoli 9. Mitoses) and the corresponding classification as benign or malignant. 66% of the instances belongs to the majority class of benign. A representative sample from the data set is listed in Table 1.

Table1: Breast Cancer Wisconsin Dataset

Input (Biopsy attributes)									Target	
A1	A2	A3	A4	A5	A6	A7	A8	A9	B	M
0.2	0.1	0.1	0.1	0.2	0.1	0.2	0.1	0.1	1	0
0.2	0.1	0.1	0.1	0.2	0.1	0.3	0.1	0.1	1	0
0.5	0.1	0.1	0.1	0.2	0.1	0.2	0.1	0.1	1	0
0.5	0.4	0.6	0.8	0.4	0.1	0.8	1	0.1	0	1
0.5	0.3	0.3	0.1	0.2	0.1	0.2	0.1	0.1	1	0
0.2	0.3	0.1	0.1	0.3	0.1	0.1	0.1	0.1	1	0
0.3	0.5	0.7	0.8	0.8	0.9	0.7	1	0.7	0	1
1	0.5	0.6	1	0.6	1	0.7	0.7	1	0	1
1	0.9	0.8	0.7	0.6	0.4	0.7	1	0.3	0	1
0.4	0.1	0.1	0.1	0.2	0.1	0.3	0.1	0.1	1	0
0.5	0.1	0.1	0.1	0.2	0.1	0.3	0.1	0.1	1	0
0.8	1	1	0.1	0.3	0.6	0.3	0.9	0.1	0	1
0.1	0.1	0.3	0.1	0.2	0.1	0.2	0.1	0.1	1	0
0.1	0.1	0.1	0.2	0.1	0.1	0.1	0.1	0.1	1	0
0.3	0.4	0.5	0.2	0.6	0.8	0.4	0.1	0.1	0	1
0.4	0.3	0.3	0.1	0.2	0.1	0.3	0.3	0.1	1	0

The input vectors and target vectors were randomly divided, with 70% used for training, 15% for validation and 15% for testing as shown in Figure 2.

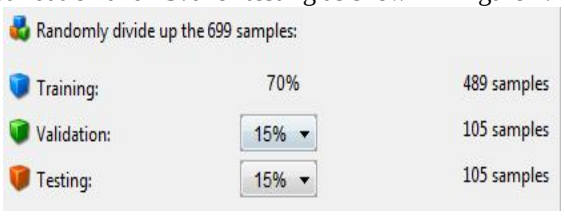


Fig.2. Division of data set

In order to partition data without losing significant modelling or testing capability, a fair way is to use cross-validation as a powerful general technique [16]. Figure 3 shows four-fold cross validation implemented.

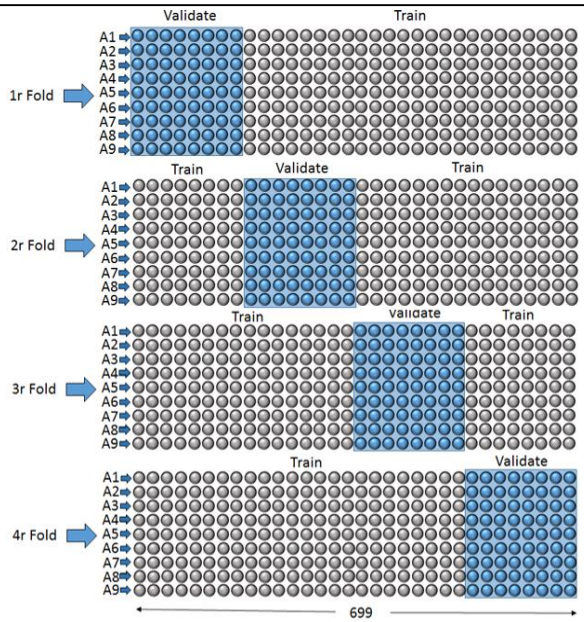


Fig.3. Four-fold Cross Validation

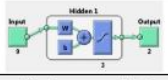
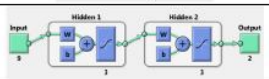
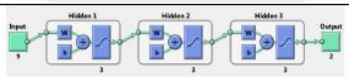
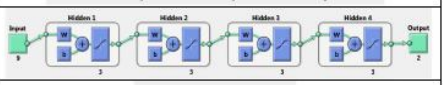
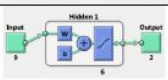
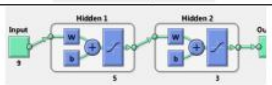
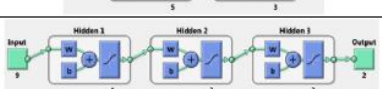
Training activity was carried out in the cancer dataset with a cross entropy error maintained below zero for all network design - retraining the network until desired CE was achieved. Cross entropy is MATLAB default performance function for feed forward networks returning penalties for inaccurate outputs [17].

Eight network structures, shown in Table 2, have been trained and tested in order to find out the relationship between the size of the neural network and the accuracy of the resulting classification. The first four networks aims to illustrate the effect of growing the network by adding hidden layers having the same neuron count; while the last four network aims to demonstrate how networks with more layers perform as compared to networks with few layers when the total degrees of freedom are kept similar. For sufficient results, a good and popular rule of thumb suggested by Prof. Yaser S. Abu-Mostafa [18] constrained all network choices. Following this, the number degrees of freedom of the network should be significantly greater than or at least ten times the number of training data. A network with less than that will tend to memorize the data and may not generalize well. Having 699 samples only allows for 69 weights or degrees of freedom at maximum. Based on this constraint, 8 possible neural network architecture were investigated in this study, as summarized in Table 2.

III.RESULTS AND DISCUSSION

The result obtained by the simulation is illustrated in Table 3. An epoch is a measure of the number of times all of the training vectors are used once to update the weights. Training stopped and weights are saved at an epoch when the validation error cease to decrease as the network starts overfitting the training data and this is monitored during the training process.

Table 2: Neural Network Models

Network	Architecture
9-5-2	
9-5-5-2	
9-5-5-5-2	
9-3-3-3-3-2	
9-6-2	
9-5-3-2	
9-4-3-2-2	
9-3-2-2-3-2	

Training is stopped when the validation error increased for six iterations as shown in one performance plot (ANN 9-5-2) in Fig. 4.

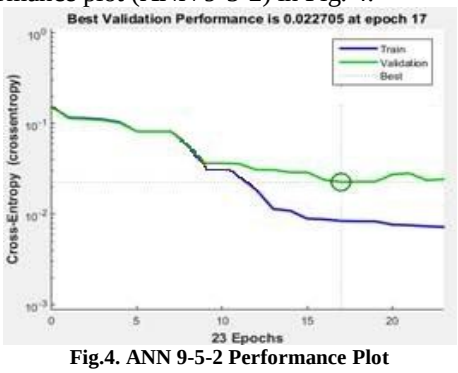


Fig.4. ANN 9-5-2 Performance Plot

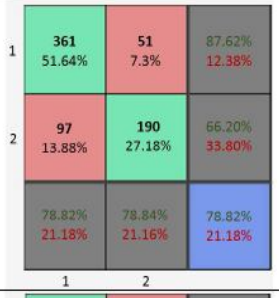
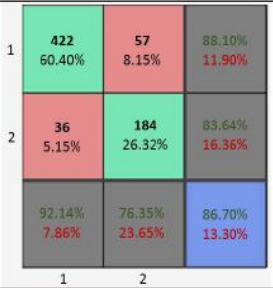
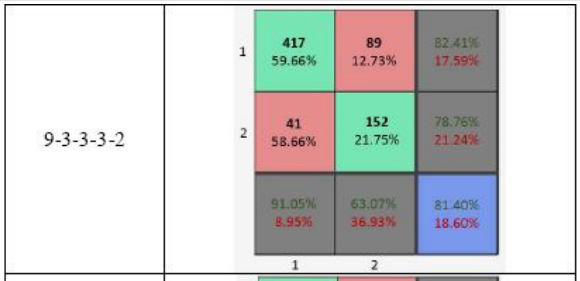
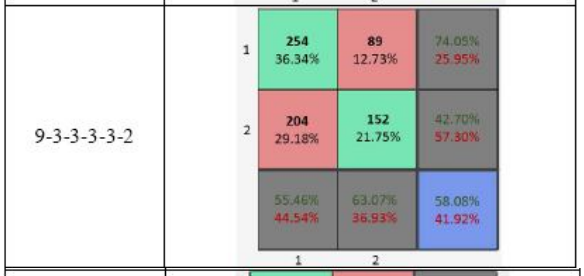
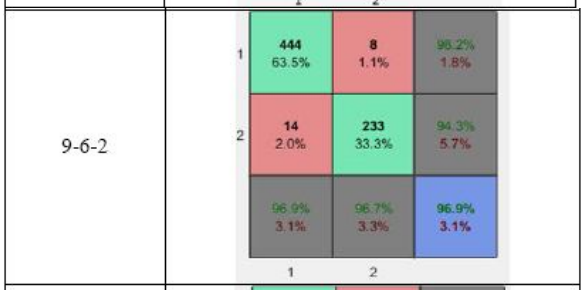
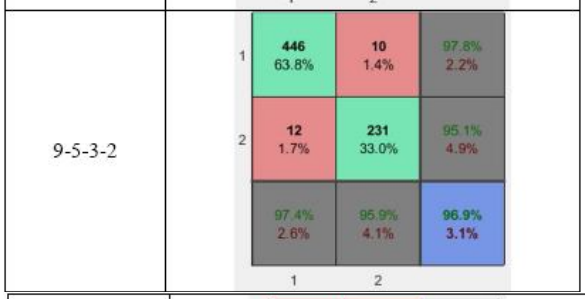
The total number of epochs (t) consumed and the error (ϵ) at that last epoch for each network per cross validation are listed in Table 3. It can be derived that that the structure of the network has a direct impact on learning time. From the first subset, it is evident that large networks take a longer time to learn the characteristics of the data. Networks with 1 and 2 hidden layers can be trained with less than 20 training cycles. Notice that for ANN having 3 and 4 hidden layers, the training cycles required are much greater than that. The improvement of the performance is insignificant in relation with the time resource put to train these networks. Conversely, a network having a structure simpler than necessary may not give good approximations even for patterns included in its training set as seen from the error at the last epoch of ANN 9-6-2 in comparison to ANN 9-5-3-2 and ANN 9-4-3-2-2. Increasing the number of layers, improves generalization ability of the network. This behavior is also evident when the networks were tested against the test data set.

Table 3: Simulation Results

ANN	1r		2r		3r		4r	
	t	ε	t	ε	t	ε	t	ε
9-3-2	1	0.135	3	0.125	1	0.146	2	0.143
9-3-3-2	15	0.168	19	0.016	14	0.020	18	0.022
9-3-3-3-2	35	0.023	26	0.021	20	0.020	25	0.017
9-3-3-3-3-2	77	0.017	87	0.013	89	0.018	71	0.014
9-6-2	11	0.015	10	0.017	13	0.012	13	0.19
9-5-3-2	21	0.015	22	0.020	27	0.016	20	0.017
9-4-3-2-2	38	0.016	32	0.017	40	0.017	40	0.020
9-3-2-2-3-2	59	0.020	44	0.017	47	0.032	41	0.020

The third subset ie. test data set, was not used during learning but it was used to compare the different networks designed. The performance of the eight neural networks set up in this study was validated using a confusion matrix. In the field of machine learning and specifically the problem of classification, confusion matrix is often used to evaluate the quality of the output of an algorithm [18]. Table 4 shows the consolidated confusion plot for each of the four fold cross validation implemented per network. The diagonal cells (green) show the number of cases that were correctly classified, and the off-diagonal cells (red) show the misclassified cases. The column on the far right of the plot shows the accuracy for each predicted class, while the row at the bottom of the plot shows the accuracy for each true class. The blue cell in the bottom right shows the total percent of correctly classified cases (green) and the total percent of misclassified cases (red).

Table 4: Consolidated Confusion Plot

Network	Confusion Matrix
9-3-2	
9-3-3-2	
9-3-3-3-2	
9-3-3-3-3-2	
9-6-2	
9-5-3-2	
9-4-3-2-2	
9-3-2-2-3-2	

The more internal structure a network has, the better that network may be at representing the problem. When the number of hidden layer increased, the network can learn more complex data by locating decision boundaries more accurately in the feature space. The addition of one more hidden layer resulted in more accurate predictions for ANN 9-3-3-2 (86.7%) against 9-3-2 (78.82%). Also, from ANN 9-

6-2 and ANN 9-5-3-2, notice that a small two hidden layer network can be used to achieve comparable performance where a single hidden layer network would require large number of nodes. However, these do not suggest that there is always a benefit in having more hidden layer. The use of unnecessary additional hidden layers (ANN 9-3-3-3-2 and ANN 9-3-3-3-3-2) can make the network too specific to the training data and, hence, give poor performance for new data set.

CONCLUSIONS

In this paper, the determination of optimum network structure using tumor biopsy dataset has been investigated. Analysis was performed to understand the effect of various internal parameters setting to accuracy and learning time of a feed forward neural networks. It was shown that the generalization power of an ANN improves for networks having more internal structure. It is also evident than this increases the number of epochs required to complete the training. However, it is important to note that large networks do not always improve the accuracy of the classification. Adding more hidden layers achieved better accuracy up to some extent only but causes complexity and training time to increase several folds. Large networks take long to learn the characteristics of the data whilst small networks may not learn at all.

For future works, several factors such as activation function and the characteristics of the data, may be studied as these also play very important role to get a network with high generalization capabilities.

ACKNOWLEDGMENTS

The author is supported by a grant from the Engineering Research and Development for Technology (ERDT), Philippines. She wishes to express deep gratitude to Prof. Processo Fernandez (PhD Adviser) and Prof. Rosula Reyes (MS Adviser) for their guidance, motivation and support throughout her graduate study. Thanks also to colleague, Christian Kevin Villanueva for his valuable inputs whenever needed.

REFERENCES

[1] http://www.dspguide.com/ch26/2.htm.
[2] http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html.
[3] S. Karsoliya. “Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture” International Journal of Engineering Trends and Technology-Department of Computer Science & Engineering, Maulana Azad National Institute of Technology, Volume3Issue6-ISSN: 2231-5381, Bhopal, India, 2012.
[4] S. Lawrence, C. Giles, A. Tsoi. “Lessons in Neural Network Training: Overfitting May be Harder than Expected” Proceedings of the Fourteenth National Conference on Artificial Intelligence, AAAI-97, AAAI Press, pp. 540–545. Copyright AAAI, Menlo Park, California, 1997.
[5] A. N. Burkitt. "Optimization of the Architecture of Feed-forward Neural networks with hidden layers by Unit Elimination," Complex Systems 5, pp. 371-380, 1991.
[6] J. Clerk Maxwell. A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, pp.68–73, 1892.
[7] J. Amini, "Optimum Learning Rate in Back-Propagation Neural networks for Classification of Satellite Images (IRS-1D)," Scientia Iranica, vol. 15, pp. 558 To 567, 2008.
[8] N. M.Wagarachchi. “Optimization of Multi-layer Artificial Neural Networks Using Delta Values of Hidden Layers” Dept. of Computational Mathematics, University of Moratuwa Sri Lanka Conference Paper · IEEE, April 2013.
[9] P. M. Atkinson, A. R. L. Tatnall, Neural networks in remote sensing, Int. J. Remote Sens. 18 (4), 699, 1997.
[10] O. Aran, O. T. Yildiz, E. Alpaydin. “An incremental framework based on cross validation for estimating the architecture of a multilayer perceptron” Int. J. Pttern. Recogn.Artif. Intell. 23 (2), 159–190, 2009.
[11] J. Xua, D. W. C. Hob. “A new training and pruning algorithm based on node dependence and Jacobian rank deficiency” Neurocomputing 70, 544–558, 2006.
[12] B. Choi, J. HongLee, D.-H. Kim. “Solving local minima problem with large number of hidden nodes on two layered feedforward artificial neural networks” Neurocomputing 71, 3640–3643, 2008.
[13] D. Sabo, X.-H. Yu. “A new pruning algorithm for neural network dimension analysis” IJCNN 2008, IEEE World Congress on Computational Intelligence, In Proc. of IEEE Int. Joint Conference on Neural Networks, Hong Kong, 1-8 June, 3313–3318, 2008.
[14] R. Reed. Pruning algorithms a survey, IEEE T. Neural Networ. 4 (5), 740–747, 1993.
[15] M. Gethsiyal Augasta, T. Kathirvalavakumar. Pruning algorithms of neural networks - a comparative Study Central European Journal of Computer Science, Department of Computer Applications, Sarah Tucker College,Tirunelveli, 627007, TN, India, Department of Computer Science, V.H.N.S.N College, Virudu Nagar, 626001, TN, India 06 September 2013.
[16] Murphy,P.M., Aha, D.W. UCI Repository of machine learning databases, Irvine, CA: University of California, Department of Information and Computer Science, 1994.
[17] R. Grossman, G. Seni, J. Elder, N. Agarwal and H. Liu. “Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions. Morgan & Claypool” 2010.
[18] G. Nasr, E. Badr and C. Joun, “Cross Entropy Error Function in Neural Networks: Forecasting Gasoline Demand” FLAIRS-02 Proceedings, 2002.
[19] D. Powers. "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation" Journal of Machine Learning Technologies 2, 37–63, 2011.

★★★