The Application of Neural Networks in Prediction Problems

Rohitash Chandra [1], Godfrey Owubolu[2]

[1] *School of Science and Technology, The University of Fiji, Lautoka, Fiji.*

[2]*School of Engineering and Physics, The University of the South Pacific, Suva, Private Bag, Fiji*

rohitashc@unifiji.ac.fj, onwubolu_g@usp.ac.fj,

Abstract. *This work presents the application of neural networks to real-life prediction problems. Neural networks are trained to predict three real world application problems given data for training and testing. We pre-process the actual output of the dataset by converting the values to integers and later translate them to binary strings. The sigmoidal output neurons of the feed-forward architecture predicts the output as binary values which are then translated back to integers to further compare the predicted output values of the network with the desired or actual output values from the dataset. The results for two, out of the three problems solved show that neural networks can be used to predict real-life problems similar to other inductive modeling approaches. A neural network can therefore be classified as an inductive modeling approach since it can be used for prediction, given the desired output . Finally, the performance of neural networks are then compared to GMDH.*

Keywords

Prediction, artificial neural networks, inductive modelling

1 Introduction

Artificial neural networks are artificial intelligence paradigms; they are machine learning tools which are loosely modelled after biological neural systems. They learn by training from past experience data and make generalization on unseen data. They have been applied as tools for modelling and problem solving in real world applications such as speech recognition, gesture recognition, financial prediction, and medical diagnostics [1, 2, 3 and 4]. Backpropagation employs gradient descent learning and is the most popular algorithm used for training neural networks. Neural networks were recently viewed as 'black boxes' as they could not explain how they arrived at a particular solution. Knowledge extraction is the process of extracting valuable information from trained neural networks in the form of 'if-then' rules as shown in [5, 6]. The extracted rules describes the knowledge acquired by neural networks while learning from examples. A neural network can be classified as an inductive modeling approach since it can be used for prediction, given the desired output. In order to use neural networks for prediction, we need to convert the predicted output value into binary strings. The prediction made by the network is compared by the actual prediction during learning. Training is terminated once the network can predict all training samples within a given threshold.

We first use neural networks to predict the amount of *vitamin B_2* per gram of turnip greens as a function of radiation in relative gram calories per minute during preceding half-day of sunlight, average soil moisture tension, and air temperature in degrees Fahrenheit. We then apply neural networks to the *end-milling tool wear experiment*. Finally, we apply them to the application of *gas furnace process* prediction problem. We end this paper with conclusion from our work and possible directions for future research.

2 Definition and Methods

2.1 Artificial Neural Networks

Artificial Neural networks learn by training on past experience using an algorithm which modifies the interconnection weight links as directed by a learning objective for a particular application. A *neuron* is a single processing unit which computes the weighted sum of its inputs. The output of the network relies on cooperation of the individual neurons. The learnt knowledge is distributed over the trained networks weights. Neural networks are characterized into feedforward and recurrent neural networks. Neural networks are capable of performing tasks that include pattern classification, function approximation, prediction or forecasting, clustering or categorization, time series prediction, optimization, and control. Feedforward networks contain an input layer, one or many hidden layers and an output layer. Fig. 1 shows the architecture of a feedforward network. Equation (1) shows the dynamics of a feedforward network.

$$S_j^l = g_j\left(\sum_{i=1}^{m} S_i^{l-1} w_{ji}^l - \theta_j^l\right) \tag{1}$$

where S_j^l is the output of the neuron j in layer l. S_i^{l-1} is the output of the neuron i in layer l-1 (containing m neurons) and w_{ji}^l the weight associated with that connection with j. θ_j^l is the internal threshold/bias of the neuron and g_j is the sigmoidal discriminant function.

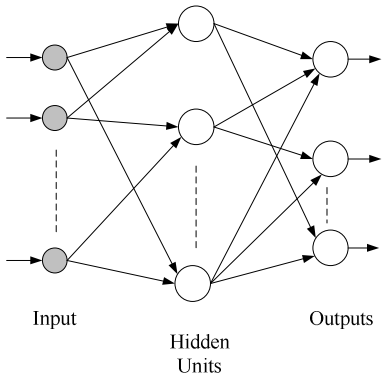

Fig. 1. The architecture of the feedforward neural network with one hidden layer. The diagram shows the interconnections of neurons from one layer to another by links known as weights. Dashed lines indicate that more neurons can be used in each layer depending on the application.

Backpropagation is the most widely applied learning algorithm for neural networks. It learns the weights for a multilayer network, given a network with a fixed set of weights and interconnections. Backpropagation employs gradient descent to minimize the squared error between the networks *output values* and *desired values* for those outputs. The goal of gradient descent learning is to minimize the

sum of squared errors by propagating error signals backward through the network architecture upon the presentation of training samples from the training set. These error signals are used to calculate the *weight* updates which represent the knowledge learnt in the network.

The performance of backpropagation can be improved by adding a momentum term and training multiple networks with the same data but different small random initializations prior to training. In gradient descent search for a solution, the network searches through a weight space of errors. A limitation of gradient descent is that it may get trapped in a local minimum easily. This may prove costly in terms for network training and generalization performance.

In the past, research has been done to improve the training performance of neural networks which has significance on its generalization. Symbolic or expert knowledge is inserted into neural networks prior to training for better training and generalization performance as demonstrated in [5]. The generalization ability of neural networks is an important measure of its performance as it indicates the accuracy of the trained network when presented with data not present in the training set. A poor choice of the network architecture i.e. the number of neurons in the hidden layer will result in poor generalization even with optimal values of its weights after training. Until recently neural networks were viewed as black boxes because they could not explain the knowledge learnt in the training process. The extraction of rules from neural networks shows how they arrived to a particular solution after training.

3 Results and Discussion

3.1 The Description of the dataset from three application problems

Experiment 1: Milligrams of Vitamin B_2 Experimentation

This study was an initial use of the NN to investigate the milligrams of vitamin B_2 per gram of turnip greens as a function of radiation in relative gram calories per minute during preceding half-day of sunlight, average soil moisture tension, and air temperature in degrees Fahrenheit. The data for this problem were taken from [7] who in turn took them from Draper and Smith's textbook as shown in Table 7. The problem solved here is to find the model that predicts the output variable.

Experiment 2: End-milling Tool Wear Experiment

The end-milling tool wear experiment was carried out on the Seiki Hitachi milling machine at the University of the South Pacific. A 16mm Co-high speed (HSS) Kobelco Hi Cut brand new end mill cutter was used to machine the work-piece. The end mill cutter had four flutes and the flute geometry was 30 degrees spiral. The overall length of the cutter was 77mm and the flute length was 26.5 mm. The work-piece machined was mid steel blocks which had a constant length of 100 mm for each trial. The machining was done under dry conditions. The work-piece used was mild steel which had a Brinell hardness number of 128.

Experiment 3: Gas Furnace Process

In this section we illustrate the performance of the neural network by experimenting with data of the gas furnace process which have been intensively studied as a benchmark problem in the previous literature [8]. For the design of experiment, the delayed term of the observed gas furnace process data, $y(t)$ is used as system input variables made up of six terms given as follows: $u(t-3), u(t-2), u(t-1), y(t-3), y(t-2)$ and $y(t-1)$. The processed data $y(t)$ (which is the output) resulted in 296 rows and 6 columns input variables of nodes in the first layer of the neural network structure.

3.2 The Performance of Artificial Neural Networks in Prediction Problems

3.2.1 Finding the Optimal Parameters for Training Neural Networks

We ran our experiments by using 60% of the data for training and the remaining 40% for testing. The first part was to determine the optimal values for the number of hidden neurons and the learning rate. All neural networks were trained until one of the three following stopping criteria was satisfied:

1. On 100% of the training examples, the prediction of the network is within 5% of the desired output, or
2. a network had been trained for 1000 epochs, or
3. the threshold of the root mean squared error has been reached.

The neurons in the network had a sigmoidal discriminant function and all networks were trained using the standard quadratic error function. The results for each experiment for the particular prediction problem are given in Table 1, Table 2 and Table 3, respectively.

Tab. 1 Trial runs for determining the optimal parameters of the network for Experiment 1: *Milligrams of Vitamin B_2 Experimentation.*

Trial No.	Network Topology	Learning Rate	Training Performance (RMSE)	Generalisation Performance (RMSE)	Training Time (epochs)
1	**3--3--7**	**0.3**	**6.94**	**8.24**	**585**
2	3--3--7	0.6	7.05	8.65	250
3	3--3--7	0.9	6.90	8.71	175
4	3--5--7	0.3	6.95	8.57	488
5	3--5--7	0.6	6.95	8.56	239
6	3--5--7	0.9	6.87	8.57	211
7	3--7--7	0.3	7.02	8.56	485
8	3--7--7	0.6	6.92	8.57	250
9	3--7--7	0.9	6.56	9.39	394

Tab. 2 Trial runs for determining the optimal parameters of the network for Experiment 2: *End-milling Tool Wear Experiment.*

Trial No.	Network Topology	Learning Rate	Training Performance (RMSE)	Generalisation Performance (RMSE)	Training Time (epochs)
1	3--3--7	0.3	0.79	3.09	339
2	3--3--7	0.6	1.13	3.00	233
3	3--3--7	0.9	1.17	3.00	236
4	**3--5--7**	**0.3**	**0.68**	**2.95**	**337**
5	3--5--7	0.6	0.90	3.00	216
6	3--5--7	0.9	0.95	3.00	196
7	3--7--7	0.3	0.76	2.89	358
8	3--7--7	0.6	0.82	3.01	218
9	3--7--7	0.9	0.87	3.12	142

Tab. 3 Trial runs for determining the optimal parameters of the network for Experiment 3: *Gas Furnace Process.*

Trial No.	Network Topology	Learning Rate	Training Performance (RMSE)	Generalisation Performance (RMSE)	Training Time (epochs)
1	6--3--7	0.3	1.45	2.47	169
2	**6--3--7**	**0.6**	**1.42**	**2.53**	**105**
3	6--3--7	0.9	1.76	2.51	1000
4	6--5--7	0.3	1.46	2.55	125
5	6--5--7	0.6	1.55	2.78	197
6	6--5--7	0.9	1.52	2.89	109
7	6--7--7	0.3	1.57	2.48	147
8	6--7--7	0.6	1.56	2.92	122
9	6--7--7	0.9	1.55	2.80	104

The above results show the performance of the network architecture with different number of hidden neurons and different learning rate values. We used the best values for each application as highlighted in the tables. We have used optimal values in the experiments to follow. Please note that RMSE is the square root of the mean squared error (MSE) in all the experiments. The network topology 6—3—7 means that 6 neurons are used in the input layer, 3 neurons in hidden and 7 neurons in the output layer, respectively.

3.2.2 Training Neural Networks for Prediction Problems

We used the optimal values for each prediction problem application as highlighted in Tables 1, 2 and 3. The neural network training procedure was the same as discussed in 3.2.1. We ran 10 major trial experiments for each prediction problem using different random weight initialisation for training in each trial.

Tab.4. The performance of neural networks in prediction problem of Experiment 1: *Milligrams of Vitamin B_2 Experimentation.*

Trial No.	Training Performance (RMSE)	Generalisation Performance (RMSE)	Training Time (epochs)
1	6.94	8.24	585
2	9.38	8.79	105
3	**7.08**	**8.15**	**496**
4	9.40	8.79	100
5	7.01	8.15	511
6	7.68	8.24	1000
7	7.05	8.24	528
8	7.05	8.43	535
9	7.05	8.15	511
10	6.91	8.43	548

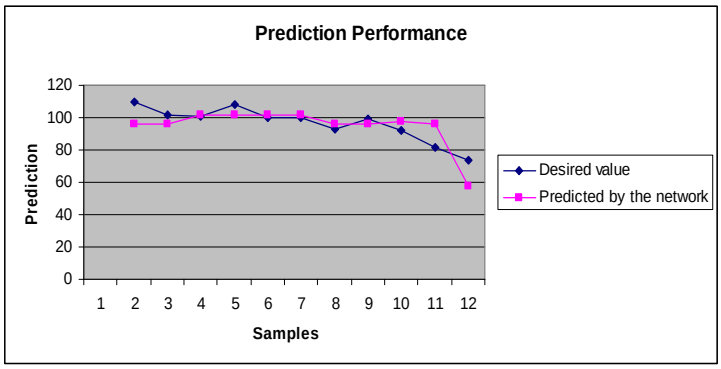

Fig. 2. Network prediction performance from best experiment highlighted in Table 4

Tab. 5. The performance of neural networks in prediction problem of Experiment 2: *End-milling Tool Wear Experiment.*

Trial No.	Training Performance (RMSE)	Generalisation Performance (RMSE)	Training Time (epochs)
1	1.71	3.68	1000
2	1.42	2.53	105
3	1.43	2.43	103
4	**1.42**	**2.47**	**105**
5	1.46	2.43	105
6	1.42	2.53	105
7	1.73	3.68	1000
8	1.43	2.43	105
9	1.43	2.54	107
10	1.42	2.47	104

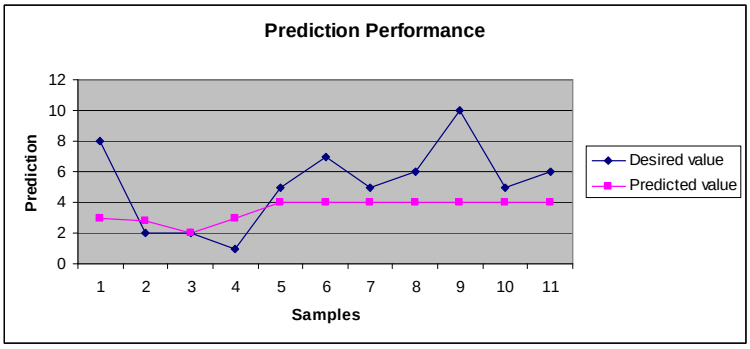

Fig. 3. Network prediction performance from best experiment highlighted in Table 5

Tab. 6. The performance of neural networks in prediction problem of Experiment 3: *Gas Furnace Process.*

Trial No.	Training Performance (RMSE)	Generalisation Performance (RMSE)	Training Time (epochs)
1	1.70	3.68	1000
2	1.42	2.53	105
3	**1.43**	**2.43**	**103**
4	1.42	2.47	105
5	1.46	2.43	105
6	1.41	2.54	105
7	1.72	3.68	1000
8	1.42	2.43	105
9	1.42	2.54	107
10	1.42	2.47	104

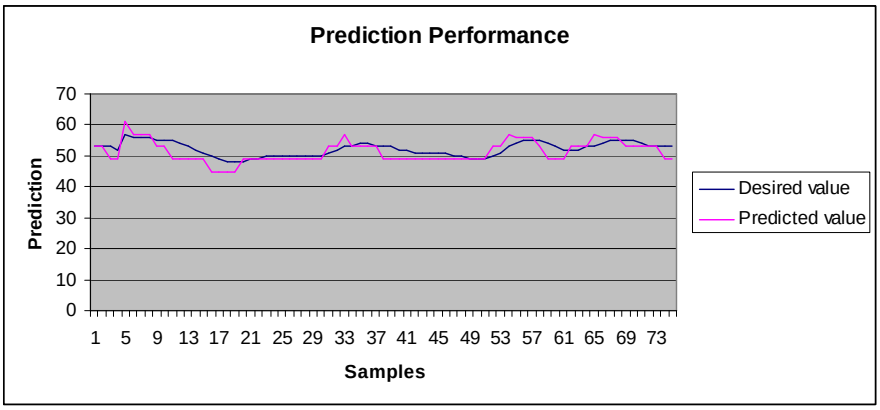

Fig. 4. Network prediction performance from best experiment highlighted in Table 6

Figure 2, 3 and 4 shows the prediction performance of neural networks compared with the desired values as graphed. Figure 2 and 4 show fairly good results while the performance of neural networks in the application in End-Milling tool wear experiment as graphed in Figure 3 show poor results. Although the results reported in this paper for these three problems using neural network are not as competitive as those obtained using the group method for data handling (GMDH), we have demonstrated that neural networks can be used to solve real-life problems and their results could be enhanced by incorporating more design features.

3.2.2 Comparison of Neural Networks to GMDH

Table 7 shows the performance of neural networks when compared to an enhanced version of GMDH.

Tab. 7. Comparison of ANN to GMDH

	Experiment 1 Milligram of B2		Experiment 2 Tool Wear		Experiment 3 Gas Furnace	
	Train error	Test error	Train error	Test error	Train error	Test error
BPN	7.08	8.15	1.42	2.47	1.43	2.43
GMDH	2.48	2.75	0.033	0.154	0.00058	0.00053

The table shows that the performance of GMDH is better when compared to ANN. Plans are on the way to improve the performance of ANN for this class of real world data and hence it is candidate for prediction

4 Conclusion

This paper presents the application of neural networks to real-life prediction problems. The vitamin B2 problem shows good results while the result for the gas furnace problem is reasonable. The tool wear solution does not seem to generalize well. This means that two, out of the three problems solved show that neural networks can be used to predict real-life problems similar to other inductive modeling approaches. A neural network can therefore be classified as an inductive modeling approach since it can be used for prediction, given the desired output. The pre-processing of desired output of the dataset by converting the values as integers and later translating it to binary strings facilitates the use of neural networks for dealing with real-life problems. The sigmoidal output neurons of the feed-forward architecture predicts the output as binary values which are then translated back to integers to further compare the actual output value of the network with the desired value from the dataset. Although this process of forward translation into binary values and then backward translation into integer values after prediction facilitates the use of neural networks for prediction, there is some loss of accuracy due to truncation.

Therefore one area of further research is to convert the given actual output values which may be floating points or integers directly into binary values; this will ensure that the original values are utilized. Furthermore, it will be useful to investigate the whether neural networks can be applied to modeling.

5 Acknowledgements

The authors acknowledge the support received from both The University of Fiji and The University of the South Pacific regarding this collaborative research work.

References

[1] Robinson A.J.: An application of recurrent nets to phone probability estimation, *IEEE Transactions on Neural Networks*, 5(2), pp. 298-305, 1994.

[2] Giles C.L., Lawrence S., Tsoi A.C.: Rule inference for financial prediction using recurrent neural networks, *Proceedings of the IEEE/IAFE Computational Intelligence for Financial Engineering. New York City*, USA, pp. 253-259, 1997.

[3] Marakami K., Taguchi H.: Gesture recognition using recurrent neural networks, *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through Technology*, Louisiana, USA, pp. 237-242, 1991.

[4] Omlin C.W., Snyders S.: Inductive bias strength in knowledge-based neural networks: application to magnetic resonance spectroscopy of breast tissues, *Artificial Intelligence in Medicine*, 28(2), 2003.

[5] Chandra R., Omlin C. W.: Knowledge discovery using artificial neural networks for a conservation biology domain, *Proceedings of the 2007 International Conference on Data Mining*, Las Vegas, USA, In Press, 2007.

[6] Fu L.: Rule generation from neural networks, *IEEE Transactions on Systems, Man and Cybernetics*, 24(8), pp. 1114-1124, 1994.

[7] Farlow S. J.: Self-organizing methods in Modeling GMDH-type Algorithms, Marcel Decker, N.Y., 1984.

[8] G. E. P. Box, F. M. Jenkins.: Time Series Analysis: Forecasting and Control, second edition, Holden-Day, San Francisco, CA, 1976.