Improving an Artificial Neural Network Model to Predict Thyroid Bending Protein Diagnosis Using Preprocessing Techniques

Younis Elhaddad
University of Benghazi

Improving an Artificial Neural Network Model to Predict Thyroid Bending Protein Diagnosis Using Preprocessing Techniques

Aiman S. Gannous, Younis R. Elhaddad

Abstract**—**Artificial Neural Networks (ANN) is a popular artificial intelligence model used to acquire knowledge from datasets in different domains by applying learning techniques which work as estimators between the available inputs and the desired outputs. Using this aspect of ANN, the diagnosis of a disease can in time be predicted from the operating data with accuracy. But, the accumulated operating data used in ANN training may contain corrupt and noisy data records. Therefore, to enhance the reliability of the trained ANN, a data preprocessing technique is necessary for preparing the training and testing data set. In this study, several data preprocessing steps are applied to a Thyroid data set before starting the training process to build an expert medical diagnosis system. Several experiments were done and the results show good improvement in predicting the diagnosis of protein bending in the thyroid.

Keywords**—**Artificial Neural Networks, back propagation algorithm, data pre-processing, machine learning.

I. Source of the Data Set

THE database directory in the university of California machine learning repository contains a thyroid database, corresponding test set, and corresponding documentation. The chosen thyroid data set contains attributes which could be used to classify the thyroid binding protein to three states:

a) Negative (class 1),

b) Increased binding protein (class 2),

c) Decreased binding protein (class 3).

ANN with back propagation will be applied to the same data set to construct a classifier model which could be used to predict these three states using an unseen data set, and to compare the accuracy of each model in order to predict the diagnosis. Several experiments will be conducted; in each experiment we will apply a preprocessing technique, observe the behavior of the model, and compare the results of the experiments.

Thyroid disease records supplied by the Garavan Institute and J. Ross Quinlan, New South Wales Institute, Sydney, Australia. 1987. Experiments are performed on the Thyroid datasets that are available at the university of California archive [3] for researchers to study and carry on such experiments in the artificial intelligence field.

II. General Steps for Preparing the Dataset

This part is important because the neural network model is only as good as the quality of the data set; in this part all the missing values are processed, and the outlier values are detected. Preparing the input data is often the most complicated part of using a neural network. Part of the complexity is the challenge of choosing the right data and the right examples for the training phase. The data must be processed before applying the ANN approach because the data set used contains missing values, and from the eye check, there are some modifications which must be made to the data set to make it perform better in the training phase.

III. Reduction of Ineffective Attributes

Examining, checking, and understanding the data set has revealed that the attribute (TBG measured) has the same value for all records. Thus, it cannot be useful for ANN learning because it will not affect or contribute to better classification. Therefore, we can delete this attribute from the data set. We can also delete the attribute (TBG) because it is correlated to the value of (TBG measured). That is, if the (TBG measured) value is (f), then it follows that the patient did not do this test, so the values of the (TBG) attribute will be considered missing values and thus are not useful because all values of (TBG measured) attribute have (f) value.

IV. Dealing with Missing Values

We do not remove all records containing missing data values within some attributes from the beginning; instead, we try to replace missing values with special outlier values first, in order to make full use of the training records. The missing values in the data set are replaced with (-1).

V. Coding Attributes

Any attributes with text values should be coded into numeric values, for example the "class" attributes are coded as follows:

TABLE I
CODING THE CLASS ATTRIBUTE

Class	Description	Code
Class 1	Negative	1
Class 2	Increased binding protein	2
Class 3	Decreased binding protein	3

Aiman S. Gannous With Department of Bioinformatics, Faculty of Public Health, University of Garyounis, Benghazi, Libya, gannous@yahoo.com, Aiman_gannous@garyounis.edu

Younis R. Elhaddad With Department of Computer Science, Faculty of Information Technology, University of Garyounis, Benghazi, Libya, Younis_haddad@garyounis.edu

VI. Designing the Network

The network was designed according to feed forward structure as shown in Fig1

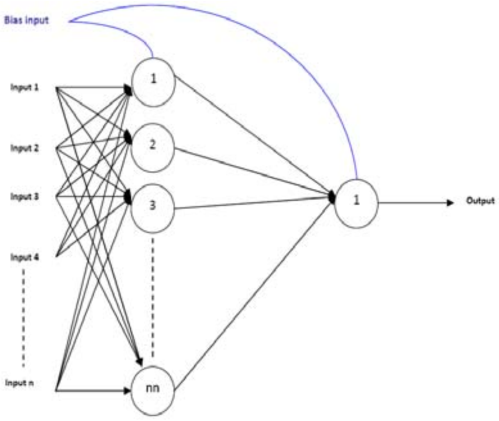

Fig. 1 Feed forward neural network structure

VII. The training phase

Training the network was done using Back Propagation Algorithm. The training was conducting with different transfer functions, a linear function in the first layer, and a nonlinear logarithmic sigmoid function in the second layer; we start the training with a unity array of weights. All of the dataset is used, consisting of 2800 records and 27 attributes after eliminating two attributes, (TBG measured) and (TBG), and replacing all the missing values with -1.

VIII. First experiment

In the first training experiments, the focus was on choosing the best learning rate. The test set consists of 972 records. Based on Table 2, there was no preferred learning rate to choose because in 20000 iterations the reached error varied from one experiment to another. Therefore, the allowed error is determined by 0.004 in order to recognize the differences in changing the learning rate from 0.0001 to 0.00001.

TABLE II
Basic experiment

No. of Experiment	Number of Neurons	Learning rate	Allowed Error	Reached Error	Number of iterations in thousands	M(ssec) in the training set	Miss classified in the training set		M(ssec) in the test set	Miss classified in the test set	
								error %			error %
1	10	0.001	0.001	NaN	_	_	_	_	_	_	_
2	10	0.0001	0.001	0.0032	20	0.0038	2719	97.11	0.007	958	98.56
3	10	0.00001	0.001	0.0037	20	0.0037	2576	92.00	0.0065	908	93.42
4	20	0.0001	0.001	0.0033	20	0.0033	2601	92.89	0.0061	912	93.83
5	20	0.00001	0.001	0.0037	20	0.0038	2746	98.07	0.0063	964	99.18
6	30	0.0001	0.001	0.0042	20	0.0042	2624	93.71	0.0081	930	95.68
7	30	0.00001	0.001	0.0035	20	0.0035	1320	47.14	0.0068	433	44.55

Table III shows that a learning rate of 0.00001 leads to better results in testing both the training set and the test set, therefore it will be fixed in following experiments, the number of neurons will be increased, and the best results observed during the testing phase. The number of iterations is defined by observing a decrease in error very slowly over 10000 iterations. In order to avoid consuming excess time, 20000 iterations will be acceptable as a termination condition.

Table IV shows the effect of the number of neurons used for training.The results obtained using 20 neurons produced a 12.5 % error rate on evaluating the training set, and 12.2 % error rate on evaluating the test set. One can see that the results of the learning process are not good; accordingly, more preprocessing techniques applied to the training data may improve learning in the ANN.

IX. Second experiment

In this experiment, we try to change the output (class attribute) of the training set into a binary number consisting of three digits as follows:

100 instead of 1
110 instead of 2
111 instead of 3

We use the same training parameters that were used in the first experiment and the structure of the ANN was changed as shown in Fig 2.

TABLE III
DETERMINING THE BEST VALUE OF LEARNING RATE

No. of Experiment	Number of Neurons	Learning rate	Allowed Error	Reached Error	Miss classified in the training set		M(ssec) in the test set	Miss classified in the test set	
						error %			error %
1	10	0.0001	0.004	0.004	2783	99.39	0.0067	960	98.77
2	10	0.00001	0.004	0.004	2726	97.36	0.0067	955	98.25
3	20	0.0001	0.004	0.004	2668	95.29	0.0068	935	96.19
4	20	0.00001	0.004	0.004	2297	82.04	0.0074	816	83.95
5	30	0.0001	0.004	0.004	2766	98.79	0.0067	963	99.07
6	30	0.00001	0.004	0.004	1758	62.79	0.0069	623	64.09

TABLE IV
DETERMINE THE BEST NUMBER OF NEURONS

No. of Experiment	Number of Neurons	Learning rate	Allowed Error	Reached Error	Number of iterations in thousands	M(ssec) in the training set	Miss classified in the training set		M(ssec) in the test set	Miss classified in the test set	
								error %			error %
1	10	0.00001	0.001	0.0034	20	0.0035	362	12.9	0.0060	131	13.5
2	20	0.00001	0.001	0.0034	20	0.0035	349	12.5	0.0066	119	12.2
3	30	0.00001	0.001	0.0034	20	0.0034	364	13.0	0.0059	124	12.8
4	40	0.00001	0.001	0.0034	20	0.0034	382	13.6	0.0060	132	13.6
5	50	0.00001	0.001	0.0034	20	0.0034	390	13.9	0.0060	133	13.7
6	60	0.00001	0.001	0.0034	20	0.0034	391	14.0	0.0060	132	13.6
7	70	0.00001	0.001	0.0034	20	0.0034	386	13.8	0.0060	130	13.4
8	80	0.00001	0.001	0.0034	20	0.0034	364	13.0	0.0060	125	12.9

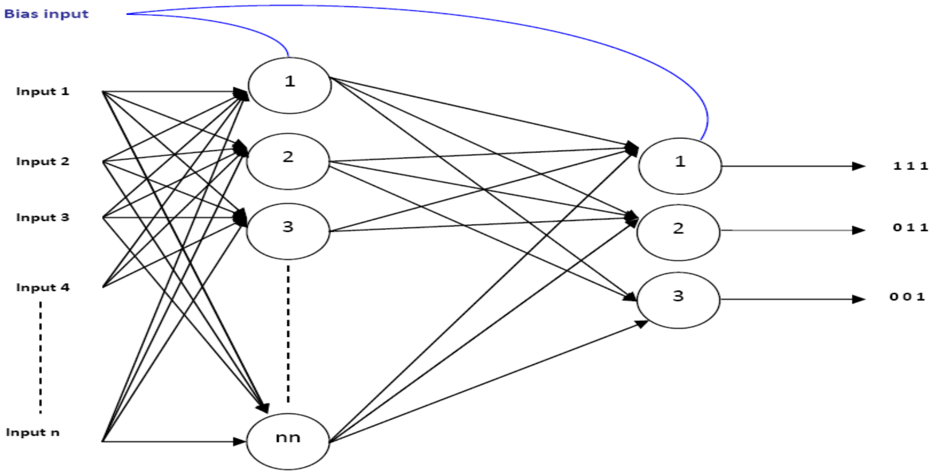

Fig. 2 Designed network for training with binary output

The results as shown in Table 5 are much better than the first experiment, so changing the output to binary has improved the classification and prediction process. The best result was obtained by using 10 neurons in the ANN. Therefore, coding the outputs in binary numbers will be fixed in the next experiments.

TABLE V
TRAINING WITH BINARY OUTPUT

No. of Experiment	Number of Neurons	Learning rate	Allowed Error	Reached Error	Number of iterations in thousands	M(ssec) in the training set	Miss classified in the training set		M(ssec) in the test set	Miss classified in the test set	
								error %			error %
1	10	0.00001	0.001	0.00138	20	0.0014	152	5.4	0.0020	36	3.7
2	20	0.00001	0.001	0.00114	20	0.0011	358	12.8	0.0018	113	11.6
3	30	0.00001	0.001	0.00106	20	0.0011	380	13.6	0.0019	132	13.6
4	40	0.00001	0.001	0.00105	20	0.0011	374	13.4	0.0019	138	14.2
5	50	0.00001	0.001	0.00106	20	0.0011	361	12.9	0.0018	122	12.6
6	60	0.00001	0.001	0.00106	20	0.0011	364	13.0	0.0018	125	12.9
7	70	0.00001	0.001	0.00106	20	0.0011	363	13.0	0.0018	124	12.8
8	80	0.00001	0.001	0.00106	20	0.0011	362	12.9	0.0018	124	12.8

X.THIRD EXPERIMENT

In this experiment the training was conducted after removing all the missing data, so the data set size was reduced to 1947 records with 22 attributes, and training was started with the same parameters from the last two experiments to determine whether there would be any further improvement of the results. The results as shown in table 6 were a little bit better than the first and the second experiments; removing all the records which contained missing values had an effect on the classification and prediction process.

The best result was using 10 neurons. But as shown in table 6, there is no effect on the learning rate by the number of neurons; also, the error decreased very slowly and was almost fixed as compared to the beginning of the training., moreover training was conducted after rearranging the input values between 0 and 1, and the results were exactly the same and show no improvement, also training was conducted using high order inputs by squaring and tripling inputs, so there were 66 inputs but the results were exactly the same and showed no improvement.

TABLE VI
TRAINING AFTER REMOVING ALL MISSING DATA

No. of Experiment	Number of Neurons	Learning rate	Allowed Error	Reached Error	Number of iterations in thousands	M(ssec) in the training set	Miss classified in the training set		M(ssec) in the test set	Miss classified in the test set	
								error %			error %
1	10	0.00001	0.001	0.0017	20	0.0018	106	3.8	0.0020	30	3.1
2	20	0.00001	0.001	0.0017	20	0.0018	106	3.8	0.0020	30	3.1
3	30	0.00001	0.001	0.0017	20	0.0018	106	3.8	0.0020	30	3.1
4	40	0.00001	0.001	0.0017	20	0.0018	106	3.8	0.0020	30	3.1
5	50	0.00001	0.001	0.0017	20	0.0018	106	3.8	0.0020	30	3.1
6	60	0.00001	0.001	0.0017	20	0.0018	106	3.8	0.0020	30	3.1
7	70	0.00001	0.001	0.0017	20	0.0018	106	3.8	0.0020	30	3.1
8	80	0.00001	0.001	0.0017	20	0.0018	106	3.8	0.0020	30	3.1

XI. Conclusions and Future Work

Three experiments were conducted in order to obtain the best weights for ANN that demonstrate a classification model that will predict the diagnosis of thyroid binding protein. The first experiment was a direct training without applying any series preprocessing technique. The best training, however, was on 10 neurons that produced a 12.2% error rate, and an error rate of 12.5% when evaluating the test set. In the second experiment was conducted after coding the class attribute values to binary code, and this technique showed an improvement in the training that reduced the error rate to 3.7% as shown in Fig 3 and a 5.4% error rate in the evaluation of the test set. In the second experiment, we started the training after removing all the records that contained missing values. This technique also showed an improvement of the results, producing a 3.1% error rate in the training process and 3.8% in the evaluation of the test set. This empirical study shows that preprocessing techniques are needed before starting any training, and involves constructing a predicting model based on real life data. Our future work will be to demonstrate and test more known preprocessing techniques on different diagnosis datasets in the medical domain, in order to determine the best road map that could be used as proven steps to begin training, to construct these types of ANN models.

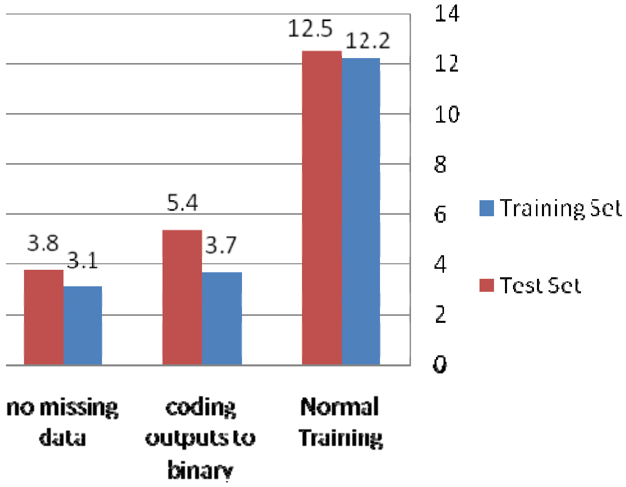

Fig. 3 Comparison of the results of the three experiment

References

[1] Thomas G. Dietterich, Hermann Hild and Ghulum Bakiri (1995), A Comparison of ID3 and Backpropagation for English Text-to-Speech Mapping Machine Learning, Kluwer Academic Publishers, Boston.

[2] Jacek M. Zurada (1992), Introduction to Artificial Neural Systems, West Publishing Company.

[3] The UCI KDD archive. Irvine, University of California, Department of Information and Computer Science, http://kdd.ics.uci.edu. Last access September 2007.

[4] Sushmita Mitra, Tinkuacharya (2003), Data mining multimedia, soft computing, and Bioinformatics, John Wiley & Sons, Inc.

[5] Daniel T. Larose (2005), Discovering Knowledge in data, an introduction to data mining, John Wiley & Sons. Inc.

[6] Michael J.A. Berry Gordon S. Linoff, Data mining Techniques (2004) Second Edition, John Wiley & Sons. Inc.

[7] Deaho Cha, Michael Blumenstein, Hong Zhang, and Dong-Sheng Jeng (July, 2006), Improvement of an Artificial Neural Network Model using Min- Max Preprocessing for the Prediction of Wave-induced Seabed Liquefaction, 2006 International Joint Conference, Vancouver, BC, Canada.