International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 18 (2015) pp 38829-38833

An Enhanced Data Cleaning Approach for Data mining Using Neural Network Methods

Agusthiyar.R

Research Scholar, Anna University Chennai, ramagusthiyar@gmail.com

Dr. K . Narashiman

Director, AUTVS - Center for Quality Management & IQAC Anna University, Chennai . knman@annauniv.edu

Abstract
In recent years, data cleaning solutions are very essential for huge data handling organizations. The huge volume of data is analyzed to predict and improve the profits of the organization's business and development. Most of the data cleaning methodologies are based on clustering, classification and association rule mining techniques of traditional methods. The results of the techniques are suited for any one of the situation and dataset types. But the neural network has many features like robustness, parallelism and creates the noise-free environment. It leads very effective data cleaning in large and real-time databases. This paper proposed an enhanced framework for data cleaning using neural networks methods and implement an algorithm for detect and replace the missing values in an appropriate location and values. There are two hidden layers are used in this framework. The first one is FFNN for train the prediction array and the second one BPNN are used to find the prediction of the value and location of the missing values. Then the SQL query is used to replace the missing values. By the experiment result, this algorithm has been checked and verified for EEG brain signal datasets. This proposed algorithm improves the accuracy and completeness of the missing value datasets.

Keywords: Data Cleaning, Neural Network, Feed Forward Neural Network (FFNN), Back Propagation Neural Network (BPNN)

1. INTRODUCTION

In data mining, the recent research problem is data cleaning in large data sets. The data cleaning methods for the business intelligence is very much need nowadays. The business strategy and development of the company is purely depends on good decision making, the data cleaning is the only cause whether the decision is good or bad. For that the past two decades there are' n' numbers of solutions are derived for data cleaning in large data sets by the knowledge engineers. All the classical methods were used for data cleaning by the researchers in different scenario's and different data sets. The traditional data mining algorithms are hard to apply on noisy data, redundant information, incomplete data and sparse data in database, or the application effects are not good. But the neural network approach for data cleaning is very narrow and cleans the noisy data effortlessly. So it is very effective on data mining in large and real datasets. [5] This paper proposed an enhanced framework and methodology for data cleaning using neural networks methods and implement an algorithm for detect and replace the missing values in an appropriate location and values. There are two hidden layers are used in this purpose. The first one is FFNN; it is for train the prediction array of inputs and the second one, BPNN are used to find the prediction of the value and location of the missing values. Then there are some SQL statements are used to replace the missing values according to the findings. By the experiment results, this algorithm is checked and verified. This proposed algorithm improves the accuracy and completeness of the datasets with efficient manner.

2. NEURAL NETWORKS TECHNIQUES

Neural networks are a new method of programming computers for machine learning. They are exceptionally good at performing pattern recognition and other tasks that are very difficult to program using conventional techniques. Programs that employ neural nets are also capable of learning on their own and adapting to changing conditions. The most basic method of training a neural network is trial and error. If the network isn't behaving the way it should, change the weighting of a random link by a random amount. If the accuracy of the network declines, undo the change and make a different one. It takes time, but the trial and error method does produce results.

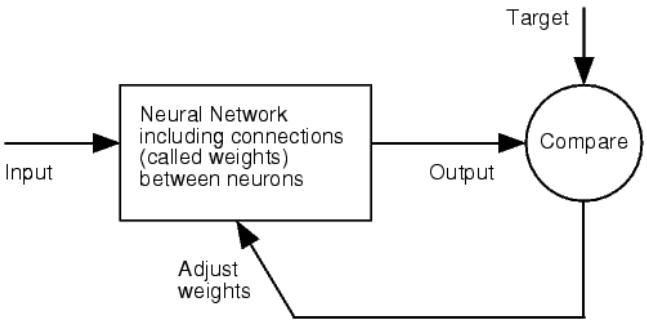


Figure1. A simple Neural Network architecture

2.1 Feed Forward Neural Network (FFNN)

Feed-forward Neural Networks allow the signals to travel one way only; from input to output. Data enters at the inputs and

passes through the network, layer by layer, until it arrives at the outputs. During normal operation, that is when it acts as a classifier, there is no feedback between layers. This is why they are called feed forward neural networks. The output of any layer does not affect that same layer. Feed-forward Neural Networks tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition.

2.2 Back propagation Neural Network (BPNN)

The basic back propagation algorithm is based on minimizing the error of the neural network using the derivatives of the error function. For hidden layers, we must propagate the error back from the output nodes. We want to train the multi-layer feed forward network by gradient descent to approximate an unknown function, based on some training data consisting of pairs (x, t). The vector x represents a pattern of input to the network, and the vector t the corresponding target (desired output). The overall gradient with respect to the entire training set is just the sum of the gradients for each pattern.

• the **error** signal for unit j:	$\delta_j = -\partial E/\partial net_j$
• the (negative) **gradient** for weight w_{ij}:	$\Delta w_{ij} = -\partial E/\partial w_{ij}$
• the set of nodes **anterior** to unit i:	$A_i = \{j : \exists w_{ij}\}$
• the set of nodes **posterior** to unit j:	$P_j = \{i : \exists w_{ij}\}$

The gradient, we expand the gradient into two factors by use of the chain rule

$$\Delta w_{ij} = -\frac{\partial E}{\partial net_i}\frac{\partial net_i}{\partial w_{ij}}$$

The first factor is the error of unit i.
The second is

$$\frac{\partial net_i}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}}\sum_{k\in A_i} w_{ik}\, y_k = y_j$$

Putting the two together, we get

$$\Delta w_{ij} = \delta_i\, y_j.$$

To compute this gradient, we thus need to know the activity and the error for all relevant nodes in the network.

Forward activation:
The activity of the input units is determined by the network's external input **x**. For all other units, the activity is propagated forward:

$$y_i = f_i(\sum_{j\in A_i} w_{ij}\, y_j)$$

Note that before the activity of unit i can be calculated, the activity of all its anterior nodes (forming the set A_i) must be known. Since feed forward networks do not contain cycles, there is an ordering of nodes from input to output that respects this condition.

Calculating output error:
Assuming that we are using the sum-squared loss

$$E = \frac{1}{2}\sum_o (t_o - y_o)^2$$

The error for output unit o is simply

$$\delta_o = t_o - y_o$$

Error back propagation:
For hidden units, we must propagate the error back from the output nodes (hence the name of the algorithm). Again using the chain rule, we can expand the error of a hidden unit in terms of its posterior nodes:

$$\delta_j = -\sum_{i\in P_j}\frac{\partial E}{\partial net_i}\frac{\partial net_i}{\partial y_j}\frac{\partial y_j}{\partial net_j}$$

Of the three factors inside the sum, the first is just the error of node i. The second is

$$\frac{\partial net_i}{\partial y_j} = \frac{\partial}{\partial y_j}\sum_{k\in A_i} w_{ik}\, y_k = w_{ij}$$

3. NEURAL NETWORKS FOR DATA MINING IN DATA CLEANING

In a decade, neural networks techniques are very useful for the field of data mining and data warehousing. In data mining, neural network is used for efficient data cleaning and it will increase the accuracy and completeness of the data. For example, Data mining on financial application of credit card mining problems including filling missing data and predicting the target value could be achieved by neural network machine learning algorithms. This work to fill the missing data in credit cards record set with more accurate and reliable data, which found a very solid foundation on credit card approval and fraud detection research [1]. The financial and the non-financial ratios in the financial statement could be used the BPN and the clustering model to compare the performance of the financial distress predictions, in order to find a better early-warning method. The BPN approach obtains better prediction accuracy than the DM clustering approach in developing a financial distress prediction model, with the exception that the accuracy rate [2]. Even the Neural Networks has complex structure, consumes more learning time and difficult to understand the representation of results, it have more acceptance ability to clean impure data with more precise and accuracy in pre-processing which results in efficient data pre-processing for Data Mining [4]. The literature survey shows that there is a research gap to detect inaccurate and inconsistent values in the datasets and the determination of attribute weight.

4. PROPOSED RESEARCH WORK

This proposed research has the framework and algorithm for data cleaning using neural network methods. The framework has the following four stages of processes. 1. Get the data source of noisy data, 2. Apply this data into the FFNN for train the prediction array of inputs. 3. Then BPNN are used to

find the prediction of the value and location of the missing values. 4. Execute the SQL query to replace the missing values according to the findings. In this paper, the EEG brain signal data have been applied in the proposed algorithm and verified the performance of the algorithm on missing and inconsistent data of data source. It has improved the accuracy and completeness of the dataset in the better way.

This research work has the procedure of neural network construction and training; in this phase, the construction of network and training of 'n' layer based on the input from the databases have done. After the construction of the network, the datasets will apply into the FFNN for train the prediction array of inputs and in the network pruning stage, BPNN are used to find the prediction of the value and location of the missing values, finally in the rule extraction stage, execute the SQL query statements to replace the missing values according to the findings.

4.1 Framework

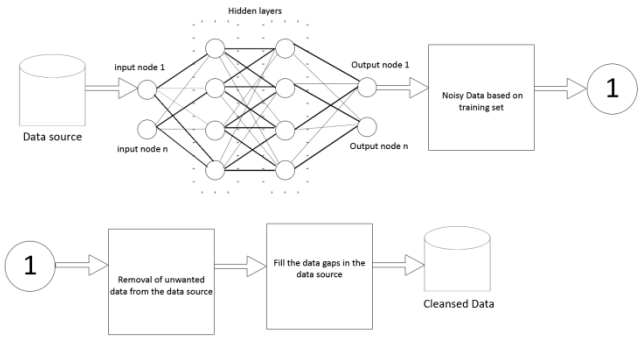


Figure2. Proposed Data cleaning Framework using Neural Network Methods

4.2 The data cleaning algorithm has been implemented in the following steps.

1. Declare the input array;
2. Declare the training set array detect the training set using feed forward algorithm.
3. Declare the prediction array ;
4. Initialize the sum, avg and count=0 variable.
5. Declare the output array
6. Initialize for loop and initialize the second nested for loop and store the location of the missing value.
7. After this process detect the values that are needed for the prediction of the values.
8. After the prediction process store the location and predicted value in the 2-dimentional array.
9. Declare the feed forward function and calculate input to the hidden layer and calculate the hidden to output layer.
10. Declare the back propagation function and calculate output layer error.
11. Update the weights for the output layer
12. Update the weights for the hidden layer.
13. Do the database operation for filling the data gaps and replacing the inappropriate value.
14. Complete the operation.

4.3 Proposed Algorithm for Data cleaning using Neural Network Methods

```
//This is an algorithm for data cleaning using Neural Networks methods of feed forward and back propagation networks.
arrayInput_Neurons = {values to be inputted};
arrayTraining_REPS = {Training sets};
array p = {train according to the training set}
int sum;
intavg;
int count = 0;
public function neuralnet()
{
feedforward();//function
array out = { };
for each(i=0;i<4;i++)
{
for(j=0; j<t.count; j++)
{
if(a[i] == t[j]){
out[i] = a[i] + t[j];
//train the prediction array
for(f=0; f<t.count; f++)
{
if(t[f] == ' ' or t[f] == 0)
{
sum += 0;
}
else
{
sum += t[f];
}
avg = sum/t.count;
count += 1
for(d=0; d< count; d++)
{
backpropagate();
//p[d] = avg;
//the prediction of the value and the location of the missing values are stored in p[i][d]
p[i][d] = {out[i], avg}
}

}
}
}

private static void backPropagate()
{
// Calculate the output layer error (step 3 for output cell).
for(int out = 0; out < OUTPUT_NEURONS; out++)
{
erro[out] = (target[out] - actual[out]) * sigmoidDerivative(actual[out]);
}

// Calculate the hidden layer error (step 3 for hidden cell).
for(int hid = 0; hid < HIDDEN_NEURONS; hid++)
{
errh[hid] = 0.0;
for(int out = 0; out < OUTPUT_NEURONS; out++)
```

```
{
errh[hid] += erro[out] * who[hid][out];
}
errh[hid] *= sigmoidDerivative(hidden[hid]);
}

// Update the weights for the output layer (step 4).
for(int out = 0; out < OUTPUT_NEURONS; out++)
{
for(int hid = 0; hid < HIDDEN_NEURONS; hid++)
{
who[hid][out] += (LEARN_RATE * erro[out] * hidden[hid]);
} // hid
who[HIDDEN_NEURONS][out] += (LEARN_RATE * erro[out]); // Update the bias.
} // out

// Update the weights for the hidden layer (step 4).
for(int hid = 0; hid < HIDDEN_NEURONS; hid++)
{
for(intinp = 0; inp< INPUT_NEURONS; inp++)
{
wih[inp][hid] += (LEARN_RATE * errh[hid] * inputs[inp]);
} // inp
wih[INPUT_NEURONS][hid] += (LEARN_RATE * errh[hid]); // Update the bias.
} // hid
return;
}
private static void feedForward()
{
double sum = 0.0;

// Calculate input to hidden layer.
for(int hid = 0; hid < HIDDEN_NEURONS; hid++)
{
sum = 0.0;
for(intinp = 0; inp< INPUT_NEURONS; inp++)
{
sum += inputs[inp] * wih[inp][hid];
} // inp

sum += wih[INPUT_NEURONS][hid]; // Add in bias.
hidden[hid] = sigmoid(sum);
} // hid

// calculate the hidden to output layer.
for(int out = 0; out < OUTPUT_NEURONS; out++)
{
sum = 0.0;
for(int hid = 0; hid < HIDDEN_NEURONS; hid++)
{
sum += hidden[hid] * who[hid][out];
} // hid

sum += who[HIDDEN_NEURONS][out]; // Add in bias.
actual[out] = sigmoid(sum);
} // out
return;
}

private static double sigmoid(final double val)
{
return (1.0 / (1.0 + Math.exp(-val)));
}

private static double sigmoidDerivative(final double val)
{
return (val * (1.0 - val));
}
```

// after this operation the location of the missing values and the prediction are send to the query operation and the database operations are performed.

//Query Operation

query = UPDATE `table_name` SET ` p[i]' = p[d] WHERE user_id = p[i]

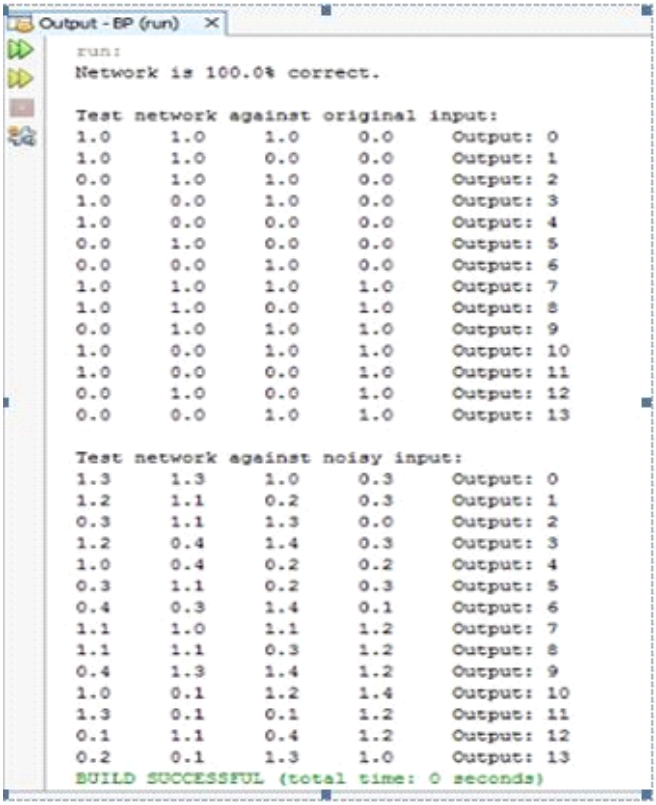
//The above code will update the empty dataset with the predicted value

sql.query(query);

//The above code will execute the query and will update the dataset.

//The table will be assigned with the cleaned data

4.4 Experiment Result



5. CONCLUSION

This neural network based data cleaning approach consists of four stages of processes. In first, it will get the data source of noisy data for training then in the second stage; this data will apply into the FFNN for train the prediction array of inputs. The BPNN are used to find the prediction of the value and

location of the missing values in the third stage. Finally, missing values could be replaced by the SQL query statements. In this paper, the EEG brain signal data have been applied in the proposed algorithm and verified the performance of the algorithm on missing data. It has improved the accuracy and completeness of the dataset in the better way.

In future enhancement as, the proposed algorithm will be checking for the datasets of different categories with different parameters.

6. REFERENCES

[1] Wei Wei (2001),Data mining using Neural Networks for Large Credit Card Record Sets, A MS-Thesisof New Jersey Institute of Technology, New York pp: 1-83.

[2] Wei–Sen Chen, Yen-Kuan Du (2009), Using Neural Networks and Data mining Techniques for the financial distress prediction model, Expert System with applications (36), Elsevier, pp : 4075-4086.

[3] Hongjun Lu, Rudy Setiono(1996), Effective Data mining using using Neural Netwoks, IEEE Transactions On Knowledge And Data Engineering, Vol. 8, No. 6, December, pp: 957-961.

[4] Jothikumar R and Sivabalan .R.V, Efficient Data Pre-Processing for Data mining using Neural Networks, International Journal of Scientific Research and Management Studies, Volume 1 Issue 4, pg: 118-123.

[5] Guoquan Jiang, CuijunZhao(2012), The Research of Data Mining Based on Neural Networks, International Conference on Computer Science and Information Technology, *IPCSIT vol. 51 , pp:51-55.*

[6] RudrasisChakraborty and Nikhil R. Pal (2015), Feature Selection Using a Neural Framework With Controlled Redundancy, IEEE Transactions On Neural Networks And Learning Systems, Vol. 26, No. 1, pp: 35-50.

Authors Biography:

Agusthiyar. R. He received his Post Graduate MCA degree from Madurai Kamaraj University, Tamil Nadu, India in the year 2001 and M. Phil degree from Periyar University, Tamil Nadu, India in 2008 respectively. He is currently doing his Ph. D in Data Cleaning in Data mining at Anna University, Chennai. He has 12 years of teaching experience in academics and now he is working as Asst. Professor (Sr. G) in Dept. of Computer Applications. His research interests include Data mining, Artificial Intelligence and Neural Networks.

Prof. Dr. K. Narashiman is the Director for AU TVS Center for Quality Management & IQAC, Anna University Chennai. India. He is a well known Teacher, Trainer, Consultant and Researcher in the area of Quality Management. He has been recognized by the state of Bremen, Germany for implementing Quality Management System. Trained with scholarship in JAPAN, he is successfully propagating Quality Management to industrial and academic community.