Missing Values in Neural Network Classification Systems: A Convolutional Neural Network Approach

Yaoren Zhang

Research School of Computer Science
The Australian National University, Canberra ACT 0200, Australia
U5793258@anu.edu.au

Abstract. *Neural networks have been adopted and applied to many real-world data classification problems. A completed full set of input data are highly preferred in these situations, but in real-world, data sets may have missing values, often encoded as blanks, NaNs or other placeholders. Especially in biological, physiological, or medical data sets, whereas patients and/or experiment subjects are not always under control, and things sometimes are not performing under expectations. These cases are the causes of missing values in data sets. This paper will perform researches based on main methods conducted by Ennett* [1] *with convolutional neural network approach which extends previous work. A major founding is that replacing missing values with means was outperform than replacing with medians or modes. In conclusion, that replacing missing values with means can be a useful method when countering missing values in neural network classification systems especially in biology or medical fields.*

Keywords: Neural Network; Classification System; Missing Value; Convolutional Neural Network; Deep Learning;

1 Introduction

Conducting research on how missing values can influence the performance of a neural network classification system, and in order to fix and improve that, a data set named Horse Colic Data Set from UCI Machine Learning Repository was been chosen. It is a data set with 368 instances and 27 attributes (300 instances for training and 68 instances for testing). However, unlike other medical or biological data sets with very few missing values that can be easily resolved by removing rows/instances with missing values, the Horse Colic Data Set contains an unignorably massive 30% data missing. The missing values are represented with '?', with all other valid values are in numerical.

A problem of classification will be solved in this paper, and for the Horse Colic Data Set, the outcome attribute (column 23) is representing whether a horse will eventually die, live or euthanized. It is similar to many medical diagnose systems which use neural network structure to predict a case/subject by its proprieties will eventually have what outcomes. Missing values can lead the neural network cannot be run, which will output a result with incorrect outputs. Filling these gap between valid attributes needs many considerations depends on what situation and what data set are using.

2 Method

To pre-process the original data set downloaded from the UCI Repository, normally people will just find a way to replace all strings into numeric data that neural network can use. And for the missing values, delete rows with missing values is

also a common practice [1]. But in the horse data set, there are too many missing values, surprisingly there are appearance of missing values in almost every instance. Thus, the method mentioned before are not going to work in this case, since by simply delete all cases with missing values can ended with a near empty data set.

In this specific data set, all the attributes are divided by spaces, firstly a space-to-comma replacement need to be done as the data loader that were using can only distinguish each row with comma. First method for processing the horse data set is to replace each missing value by mean values. In detail, an `Imputer()` pre-processing class under sklearn package will be used. The strategy attribute in `Imputer()` can be set with either "`mean`", "`median`" or "`most_frequent`". And every missing value will be determined by the means, medians or modes of valid values along their corresponding column. This is also a method suggested and examined by Ennett [1].

However, another possible method conducted by researchers which replacing values by normal is not suitable for this case. From the description of the Horse Colic Data Set, some attributes are not filled with continues data, and some of them are discrete data. Most of the attributes are just 0 to 10 or less, calculate the normal of these attributes does not make any progressive changes. Using of normal will not have expected outcomes in terms of output a meaningful numeric value. In the other words, normalize data to a new scale of numbers can leads to an incorrect result.

2.1 Convolutional Neural Network

The implemented convolutional neural network is a three-layer convolutional neural network. In the previous work, a three-layer forward feed neural network were used. Kernels with size of 3 by 3 are used as well as max pooling. Sigmoid function has been chosen for the activation function. Although through the process of testing with different activation functions like ReLU or Softplus etc., sigmoid function outputs best curve (the smoothest) than others. A similar result can be observed by using tanh function. Number of epoch that used by training is set to 6000, due to the relatively small instances number on size. Also, the learning rate is being set to 0.001, in order to obtain a smooth curve on the "loss" axis.

Shared weight topology [2] has been chosen to improve the performance and accuracy of the neural network in the previous work. The auto-associative network will assign different weight between connections of units. In addition, as a new approach, training set reduction method [3] has been tested in this paper. With deep learning approach, the convolutional layers are initiated with `conv2d()` function, with one fully connected layer. Function `max_pool2d()` was been used for pooling. Model regularization strategies like Dropout [4] was not used as the data set is relatively small, which does not require powerful GPU implementations.

In this three-layer convolutional neural network, each layer consists of convolution of the previous output layer. After learning filters as 3 by 3 kernels, max pooling over local neighborhoods are performed in first layer, and the final layer were end with a fully connected network using one full-connected layer.

2.2 Hyper Parameters

Understanding and adjusting hyper parameters in convolutional neural network

is crucial in terms of get better training/ testing accuracies. By searching with brute forces, optimal hyper-parameters are being found and it is resulting an outcome of the highest testing accuracy.

2.3 Test Methodology

The evaluation method of neural network is to print out loss and accuracy in training combined with the final test data set accuracy. It reads the data set in a same way that reads training data, and it will perform a forward pass computation of predicted target by passing input to the neural model. Therefore, the accuracy can be calculated by dividing predicted target set to the test target set. This evaluation method can be visualized by plot a graph which takes epoch times versus loss. Resulting a curve with smoothness character, which stands for the continues reducing of loss is visible.

3 Results and Discussion

By the first method applying to the convolutional neural network, which compute the mean values of missing values' corresponding columns, resulting an accuracy of 69.12%. After 6000 times of epoch, loss is close to 0.8231. Changing the activation function leads to no difference in terms of the curves and testing accuracy. Using sigmoid and ReLU resulting same testing accuracy, which is different from the results in the previous work. In previous work, change activation function from sigmoid to ReLU gives a significant change in the testing accuracy. As seen from **Fig.1** and **Fig.2**.

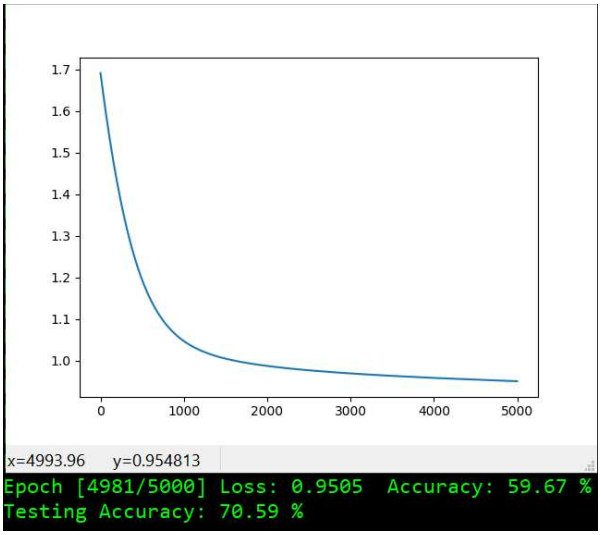

Fig.1 loss versus epoch(Sigmoid)

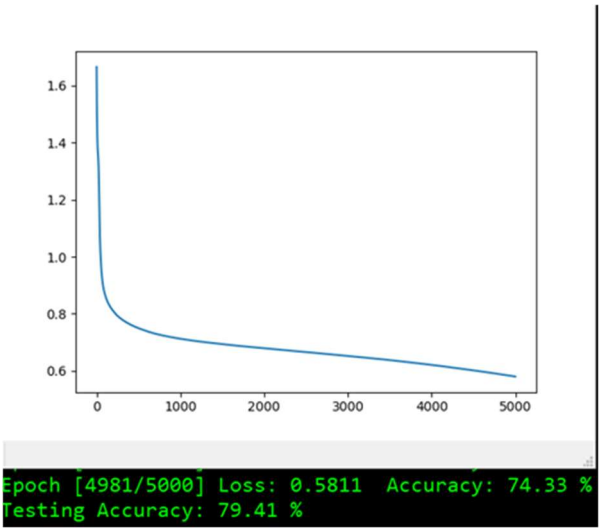

Fig.2 loss versus epoch(ReLU)

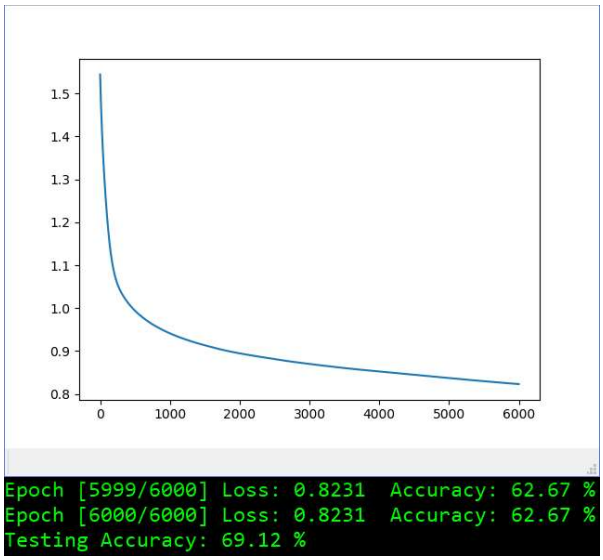

Fig.3 loss versus epoch(CNN)

Determine the best approach for this classification problem can be complicated. In the previous work, using a normal forward-feed neural network with two or three layers can results a testing accuracy near 80%. Achieving a higher testing accuracy is expected but as seen in **Fig.3**, 69.12% made by convolutional network is even worse than the 70.59% made by normal neural network. The results are same for using the median or mode to replacing the missing values.

Training set reduction are also being applied and tested in this convolutional neural network. Due to the small amount of the instances, the testing accuracy remains same for 300, 150, 200 and 100 instances. Comparing to the results made in [3], the classification accuracies are not being improved as expected. The other reason why the results are not getting improved may be the uncleanness of the data set. As in Gedeon's paper [3], the classification accuracy only improved in clean data set.

4 Conclusion and Future Work

Establishing neural network diagnose systems is crucial to help researchers predict and determine what will eventually happened to the subject. Replace missing values by re-examine the subject in medical data may be hard because a particular test is expensive or inconvenient to patients [5]. Although replacing the missing values with means can have the data set completed. But in specific cases, means, medians or modes can bias the data set towards an unwanted non-accurate result. Same as the normal value method. In the Horse Colic Data Set, there are too less instances, and using a convolutional neural network makes worse performances. This is caused by too less feature points for convolutional neural network to get. Using this model on a larger data set can definitely results a significant change.

Network reduction is a superior method in some cases [5]. It uses a classifier to determine and categorize the data set based on the missing values, and it will place the data set into an appropriate classification network. In terms of future works, implementing the network reduction method may be a priority as it is also used by some industry practitioners. With comparison to the network reduction method, another method called value substitution can also put under the future implement scope. By training a new value substitution network [5], it can almost completely cover the whole data set by process the complete pattern of the input networks.

A better approach when handling the missing values in data set is to construct and train a predictive model, to estimate the what possible missing values can be replaced.

Through the research on this topic, getting a deep understand about building and testing a customized convolutional neural network can be consider as one of the major outcomes of this paper. Understanding how medical researchers dealing with missing values in their patient data sets can also be an inspiration to neural network researchers in other areas. Replacing missing values is for improving the accuracy of the whole data set instead of getting the individual case correct.

References

[1] C. M. Ennetta, F. Monique and C. R. Walker, "Influence of Missing Values on Artificial Neural Network Performance," p. 5, 2001.

[2] T. Gedeon, "Stochastic bidirectional training," Sydney NSW 2052 AUSTRALIA.

[3] T. Gedeon, P. Wong and D. Harris, "BALANCING BIAS AND VARIANCE: NETWORK TOPOLOGY AND PATTERN SET REDUCTION TECHNIQUES".

[4] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *ECCV*, Zurich, Switzerland, 2014.

[5] P. Sharp and R. Solly, "Dealing with Missing Values in Neural Network-Based Diagnostic Systems," *Neural Comput & Applic,* vol. 77, p. 3:73, 1995.