Predict readers' native language with Neural Network and evaluate strength of correlation of each feature with output

Peilin Li
Research School of Computer Science
Australian National University
Acton ACT 2601 Australia
u6497085@anu.edu.au

Abstract. Neural Network with logistic regression has been wildly used to make predictions for classification problems. Brute Force and Genetic Algorithm are also used in evaluating of the strength of correlation between each feature and the prediction. In this paper, these two methods were used to assess how significantly each feature can affect the prediction and the performance of these two methods were compared. It was figured out from the experiment that both methods could help select features with strong correlation to the output, but they also had shortcomings. Brute Force presented decent reliability but not ideal efficiency because of its high time complexity. Simultaneously, Genetic Algorithm showed much better efficiency and a little bit weaker reliability, but it still a more efficient method than Brute Force.

Keywords: Neural Network, Test strength of correlation, Brute Force, Genetic Algorithm, Native Language Testing

1 Introduction

In the dataset paper, the subjects said that there was no significant difference in eye movement and reading behaviour between first language English (L1) readers and second language English (L2) readers in the reading process [1]. Since readers do not read word by word but predict the meaning of the passage as they read, L1 readers and L2 readers will have different reading strategies and skills because of their different "predicting habit" [2]. And this should lead to some significant differences in their performance in reading, which contradicts the experimenters' conclusions.

And how to determine the strength of correlation between the input characteristics and the predicted results is a big problem in the application of neural network technology. Failure to explain how the predicted values are derived can lead to doubts about the reliability of prediction models when they are used [3]. Brute Force is a very general technique to solve problem which systematically lists all possible candidates for the solution and checks whether each candidate meets the problem statement. Resent research said that Brute Force can also be used in evaluating how each feature affect the prediction in Neural Network [4]. And Genetic Algorithm was considered to be one of the optimal algorithms for Brute Force and it follows the principle of natural genetic operators such like reproduction, crossover and mutation [5]. These two methods will be tested and compared in this report.

So, Neutral Network will be used to analyse the behaviour of the reader in the process of reading to determine whether the reader is L1 reader or L2 reader. In addition, the mentioned methods will be hired to find out the features that have important effect on the result.

2. Method

2.1 Dataset pre-processing

The given data set has 4 forms and the form named ***FOR SPSS*** contains all raw data. Thus, it will be used in the following experiment. The data set has 66 rows of data, and 23 columns [66*23]. In the columns, ***Participant ID*** is obviously irrelevant to the experimental data. And ***Condition [2ed column]*** is the combination of values in ***Text Type*** and ***Condition [4th column]***. So, ***Participant ID, Text Type and Condition [4th column]*** are deleted.

And the categorical data (***Condition [2ed column]*** and ***L1/L2***) are encoded to make the data can be used in the Neural Network. For the same reason, data in ***Time Taken*** are replaced with number of seconds instead of "minute: second".

At the last of pre-processing, columns are renamed to strings without special symbols to facilitate subsequent data analysis and ***L1 or L2*** is replaced at the last column because it's customary to put the dependent variable on the last line.

2.2 Network Design

Because the goal is to predict the class with 19 features which means the Neutral Network only has one output and 19 inputs. So, the model was designed with about 19 hidden neuron and one hidden layer. And the number of hidden neural was decided as 20 with the performance of model with different number of hidden neural (Table 1).

Number of hidden neural	16	17	18	19	20
Mean Test Accuracy	60.73%	61.07%	62.33%	62.26%	63.73%

Table 1 No. of hidden neural vs Model Accuracy

Since the problem to be solved is a binary classification problem, and the results of the experimenters suggested that the difference between the two classes is not obvious and there is experiment prove that sigmoid function has good sensitivity in 0-1 classification problem [6]. Sigmoid function was used as the activation function between every two layers.

And at the beginning, the learning rate was 0.05 and the loss during training has a tremendous shock (Fig. 1 left). And this is because the learning rate is too high, and the lowest point is skipped during the gradient descent. Finally, after setting the learning rate to 0.001, the decline curve became smooth (Fig. 1 right). And the initial model was built whose average accuracy on test set is 63.7%.

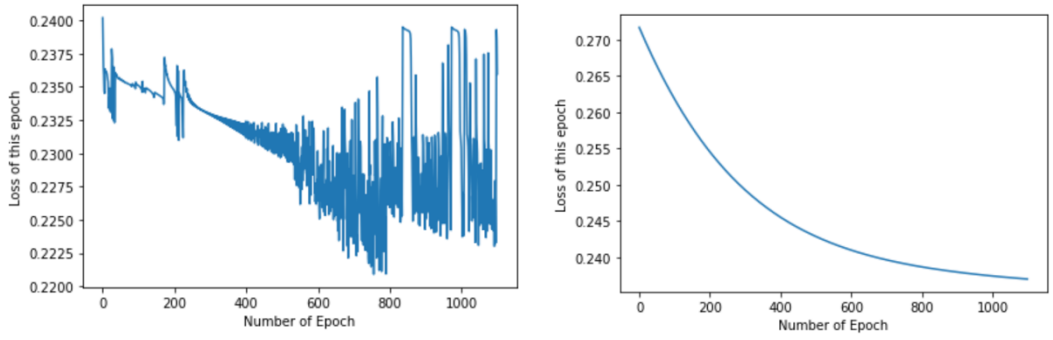

Figure 1 Loss graph of models with learing rate as 0.05 and 0.001

So, after adjusting all the hyperparameters, the final model has hyperparameters as (Table 2):

Input neural	Hidden neural	Output neural	Activation function	Learning Rate	Epoch Number
19	20	1	Sigmoid	0.001	1100

Table 2 Hyperparameters of the model

Because the size of data is small, K-fold was used in the splitting of training and test data. And k was set as 5 to divide training data and testing data by a ratio of 8:2. However, it is not ideal to use K-fold in the subsequent experiments, because the implementation of Brute Force and Genetic Algorithm requires a large number of cycles. If K-fold is used, the time complexity will be too high, and the operation time will be too long. So, in order to simplify the calculation and ensure that the subsequent experiment will not be affected by special training data and test data to the greatest extent, the training data and test data with the closest accuracy to the final result from the 5 trainings were selected as the training data and test data of the subsequent experiment.

2.3 Brute Force

Brute force is a way uses the enumeration method to reduce the number of input features, and compares the accuracy of the new model with the original model to determine the missing of which feature will lead to a significant decline in the prediction accuracy, so as to obtain how significantly can every feature influence the prediction. The elimination of just one input every time produces inconsistent results, so two inputs are eliminated every time [4].

This initial model has 19 inputs and there are 171 ways to remove 2 inputs as there are 171 combinations.

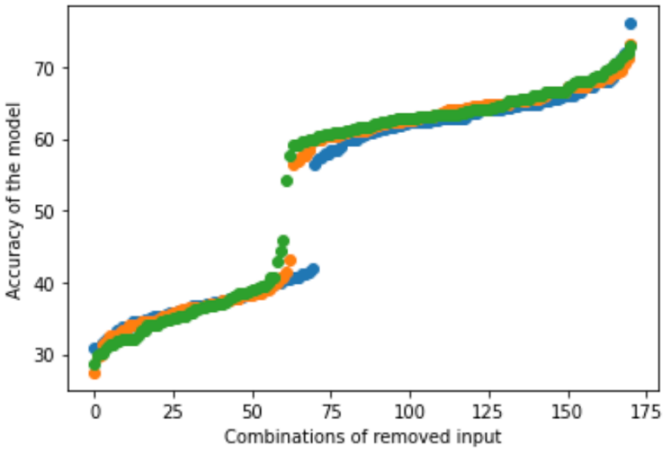

Figure 2 Model Accuracy vs Removed features

The figure above (Fig. 2) shows the accuracy of every situation with different combinations of removed inputs. And three curves with different colour illustrate the situation of three runs. In order to control the experimental variables and to make "combinations of removed inputs" be the only variable, the initial weight of each training was generated from the initial model, and the weight of removed inputs were removed also.

Each curve is divided into two parts. And the difference in accuracy between the two parts is obvious, which indicates that the missing of some inputs can significantly influence the accuracy of prediction, and these inputs are closely relevant with output.

Due to the need for very many times training when use Brute force, the slow fitting speed and requirement of more epochs and lower learning rate of sigmoid function make the cycle time too

long. So, the activation function of first layer of the neural network was changed from sigmoid to ReLU, which can fit faster with less epoch and high learning rate. The model is going to be a little bit less sensitive but it's still acceptable.

Because when the input with strong correlation with output is deleted, the quality of the input becomes very poor due to the lack of important information. The neural network also complies with the "garbage in garbage out" principle [7], so the quality of the neural network also becomes low, and the prediction accuracy is greatly reduced. So, if a feature was deleted in most points of lower part of Figure 1, there must be a strong correlation between it and the output, and if it was deleted in most points of upper part, the correlation between it and output is week.

So, after running the experiment, through the statistics of remaining inputs in situations of the upper and lower parts of the image, the ranking of the correlation strength between each input and output on the result can be obtained (Table 3). And the definition of strength of correlation with output (SoC) here is (1). And in Table 3 the SoC was rescaled to (0,1).

$$SoC = \frac{number\ of\ lower\ points\ without\ this\ feature}{number\ of\ upper\ points\ without\ this\ feature} \quad (1)$$

Strength of correlation with output	Feature
0.94	Reading ratio, Scan ratio, Time Taken
0.81	Skim ratio, F_Out/In
0.69	F_in_T, FD_Out/In, Total Score
0.56	Total fixation dur (s), FD_in_T, FD_out_T
0.50	Condition, number of distractions, Longest reading sequence
0.44	Feel_dised, Use_during_work
0.19	F_out_T, Number of Distractions (from DB)
0.06	ToF

Table 3 Correlation strengh rank

2.4 Genetic Algorithm

The Genetic Algorithm can select good choices of parameters and get ideal combination of them with a principle of natural genetic operators like Figure3.

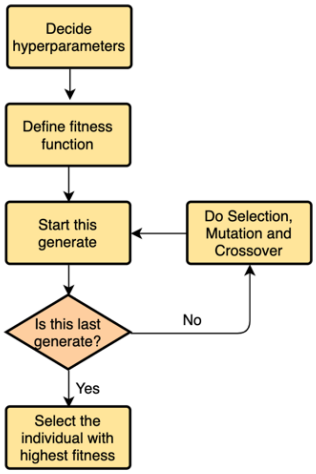

Figure 3 Princeple of Genetic Algorithm

Use features that have weak correlation with output as part of input data can reduce the performance of the model [8]. So, with setting if select each feature as genes of the chromosome and setting the performance of the new model with new features as the fitness value of each individual, after enough generation of evolution, the DNA of most fitted individual should just select the features that have strong correlation with the output. And the hyperparameters in this algorithm and the initial values (from lab 8) of them are (Table 4):

DNA Size	Population Size	Cross Rate	Mutation Rate	Generations No.
10	100	0.8	0.002	100

Table 4 Default hyperparameters of GA model

Firstly, the DNA Size here should be the number of features which is 19. And other hyperparameters should be considered together because each of them may influence other 3 [9]. Considering that DNA Size is quite large, in order to reduce the computing cost, the Population Size and number of generations should be smaller. And it's important to make sure that good DNA can be got after these generations. So, Cross Rate and Mutation Rate may be larger. And several combinations of these 4 parameters and the accuracy of model with best DNA's features are shown in Table 5

Population Size	Cross Rate	Mutation Rate	Generations No.	Accuracy
50	0.8	0.005	40	64.72%
30	0.8	0.02	15	71.32%
10	0.8	0.2	8	62.19%

Table 5 Model Accuracy vs Hyperparameters of the model

So, the hyperparameters of final model was set as (Table 6):

DNA Size	Population Size	Cross Rate	Mutation Rate	Generations No.
19	30	0.8	0.02	15

Table 6 Model hyperparameters

The fitness function is also important in genetic algorithm and in this case the fitness function should make the individual that makes neural network have high performance have high fitness and vice versa. A paper of University College Dublin shows that in genetic programming for classification problem, (2) can be a good fitness function because it can nicely show the accuracy of the prediction [10].

$$Fitness = \frac{TP}{TP+FN} + \frac{TN}{TN+FP} \tag{2}$$

After running this genetic model, the features that best DNA selects are: 'ToF', 'F_in_T', 'F_out_T', 'F_Out/In', 'FD_Out/In', 'Reading ratio', 'skim ratio', 'Time Taken', 'Total Score'. And the accuracy of the new Neural Network model is 71.3% which is higher than the initial one which is 63.7%.

3. Comparation and Discussion

3.1 Compare with Benchmark

In this case, though the problem is a binary classification problem, the predicted value is a continuous number in range of 0~1. The value of input features are also continuous numbers. And Pearson's correlation coefficient (r) is a good measure of strength of correlations between these kinds of values [11]. So, the rank of r between each feature and "L1 or L2" was obtained with Rattle (Table 7).

Feature	Pearson correlation coefficient
skim.ratio	-0.47230831
Time.Taken	0.465336223
Reading.ratio	0.333953659
Longest.reading.sequence	0.31368781
number.of.distractions	-0.273981448
Total.Score	-0.26008695
Number.of.Distractions..from.DB.	0.194814525
FD_in_T	0.187254216
Total.fixation.dur..s.	0.182010577
FD_Out.In	-0.12010151
F_Out.In	-0.12010151
scan.ratio	-0.111072229
F_in_T	0.10427312
ToF	0.097623146
F_out_T	-0.092568229
Use_during_work	-0.088796441
FD_out_T	-0.032463575
Feel_dised	-0.007358115
Condition	-0.003491584

Table 7 Rank of Pearson correlation coefficient

With comparing three different results, it can be seen that the ranks of Brute Force and rank of r are quite similar, and the features which were selected with Genetic Algorithm are also have high value of r. What's more, all results show that there is a strong correlation between the reading behaviour of a reader and whether the reader is L1 or. L2 reader.

3.2 Discussion about Advantages and Disadvantages of two techniques

The advantage of Brute Force is that it has high reliability, is not easily disturbed by other conditions, and can more accurately obtain the relevant degree between each input and output. But its disadvantages are also significant. Brute Force relies on exhaustive methods to reach conclusions and requires a lot of times of model training, which can be time-consuming and computationally expensive if the model's topology is complex or has more types of inputs.

Though Genetic Algorithm has similar problem of high computation cost and time complexity, it can be smarter and more efficiency than Brute Force. Because it's not trying to test every situation with deleting different numbers and kinds of features, it's learning which features should be delete and which are not with much less times of trying than Brute Force. But the reliability of Genetic Algorithm is not high as Brute Force. Although the computational cost of Brute Force is very high, it traverses all the possibilities, so it can't miss the optimal solution. However, the result of Genetic Algorithm can be influenced by the hyperparameters of it and not optimal choice of hyperparameters can lead not ideal result just as other two situations in Table 5.

4. Conclusion and Future Work

Through this experiment, it was proved that there is a strong correlation between readers' reading behaviour and the types of their mother tongue and practiced two methods to determine the strength of correlation between each input and output in the neural network. Among them, Brute Force shows good reliability but high calculating cost, while Genetic Algorithm, although is not as reliable as Brute Force, is good at getting ideal result in with much less computation.

It can be figure out that the small size of data limits the performance of the neural network and the

evaluation of the correlation strength. With more data, the reliability of the neural network and evaluation can be increased. What's more, how to make Genetic Algorithm be more reliable is a really interesting question. I think trying to change hyperparameters automatically (e.g. after running, if the best DNA's performance is not ideal, change hyperparameters and run again until the performance is good enough) is a good way to do that. However, how should the computer know should it increase or decrease the values of hyperparameters and how to determine an optimal fitness function are still good questions which need more research.

References

1. Copeland, L. & Gedeon, T.(. 2015, "Visual Distractions Effects on Reading in Digital Environments: A Comparison of First and Second English Language Readers", Association for Computing Machinery (ACM).
2. Singhal, M., 1998. A comparison of L1 and L2 reading: Cultural differences and schema. The internet TESL journal, 4(10), pp.4-10.
3. Féraud, R. and Clérot, F., 2002. A methodology to explain neural network classification. Neural networks, 15(2), pp.237-246.
4. Gedeon, T.D., 1997. Data mining of inputs: analyzing magnitude and functional measures. International Journal of Neural Systems, 8(02), pp.209-218.
5. Cortez, P., 2014. Modern optimization with R. New York: Springer.
6. de Oliveira, E.J., da Silva, I.C., Pereira, J.L.R. & Carneiro, S. 2005, "Transmission system expansion planning using a sigmoid function to handle integer investment variables", IEEE Transactions on Power Systems, vol. 20, no. 3, pp. 1616-1621.
7. Yale, K., 1997. Preparing the right data diet for training neural networks. IEEE Spectrum, 34(3), pp.64-66.
8. Guyon, I. and Elisseeff, A., 2003. An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), pp.1157-1182.
9. Vose, M.D., 1995. Modeling simple genetic algorithms. Evolutionary computation, 3(4), pp.453-472.
10. Le-Khac, N.A., O'Neill, M., Nicolau, M. and McDermott, J., 2016, March. Improving fitness functions in genetic programming for classification on unbalanced credit card data. In European Conference on the Applications of Evolutionary Computation (pp. 35-45). Springer, Cham.
11. Benesty, J., Chen, J., Huang, Y. and Cohen, I., 2009. Pearson correlation coefficient. In Noise reduction in speech processing (pp. 1-4). Springer, Berlin, Heidelberg.