Grade prediction with Evolutionary Algorithm Optimised Neural network and Network Reduction technique

Mingxuan Dong

College of Engineering and Computer Science
Australian National University
Canberra 2601 AUSTRALIA
E-mail: u6993885@anu.edu.au

Abstract. The determinations of the hyper-parameters for neural networks are empirical and causes much unnecessary time consuming. And Evolutionary Algorithm is a good way to solve this question. With the help of this, we could constraint the number of hyper-parameter combinations in a quite small number. And we implemented the evolutionary algorithm on a fully connected neural network with pre-defined structure to optimise the hyper-parameters. We also tried the network reduction techniques on the network. The result shows that though this technique could improve the training efficiency to some extent, it is not suitable for all the circumstances.

Keywords: evolutionary algorithm, neural network, classification, regression, network reduction

1 Introduction

Neural network has been a hotspot for research for many years. The neural networks abstract the network of neurons in human brain from the perspective of information processing to build up models by different network connection methods. Neural network is also a calculation model, it composed of many small calculation units called neuron. There's a weight between every two connected neurons that is called weight, which is equivalent to the memory of neural networks. The networks would learn these weights from the training process. Also, Neural networks with backpropagation have prevalent for many years. The backpropagation is a concise theory that is also easy to implement (Choi & Gedeon, 1995). With the calculation abilities' increase, especially after the GPU came out, this technique with long history again appeared in our sight. But reducing the amount of the calculation is still an essential issue for this area for efficiency consideration.

The neural network is also a data driven model that with a strong ability to simulate the data. The questions this model solve could mainly be divided into two areas. One is classification and another is regression. Classification, as its name implies, the goal is to classify an entity into one of the established classes. For regression problem, it mainly represents the problem to predict a specific number. For example, the weather forecast, grade prediction are all regression problems.

One of the disadvantages for neural network is its interpretability. Many of the researchers could not explain why their network works with these hyperparameters and don't work for others. A well-structured neural network usually has much hyper-parameters like learning rate, batch size, epochs even the network structure (layer numbers and units for each layer). The parameters are just too much for human to consider. There're only some empirical conclusions for adjusting the hyper-parameters and it usually cause much experience to manage them.

Evolutionary algorithm is another machine learning algorithm inspired by the biological theory *the evolution theory*. The algorithm could use the initialised figures for evolution. With the defined evaluation method and evolution method, the figures could receive better marks on the evaluation method. In this progress, the number we want would evolute to the optimised result.

Also, the network may get into a scenario called overfitting. It means the network learns too much from the training set that is not robust enough to face the unfamiliar data even it fits the training data excellent. And now, we got several practical technologies to deal with this question.

2 Application Domain

The data we used is a set of assessments marks in a Computer Science course (Choi & Gedeon, 1995). The assessments take up 40% of the total mark. And the other 60% of mark comes from the final examination's marks, which is not offered here. So the task is to predict what grade shall a student receive finally. With the help of this prediction, students could know how much effort they should make to achieve their desired grade. The task asks us to classify the final marks into grade, the grades' rule is listed:

- Distinction. The student should be marked of 75 or more to achieve this grade.
- Credit. For students marked from 65 to 74.
- Pass. For students marked from 50 to 64.

- Fail. For students marked less than 50.

3 Method

3.1 Evolutionary Algorithm

The evolutionary algorithm is a machine learning algorithm inspired by *The Evolution Theory* of biology domain. In brief words, the species are evolving in the principle of natural selection. The ones fit better on the natural environment got better chance of reproduction. So, the species in general would fit the environment better in this progress.

The evolutionary algorithm simulates this natural process exactly. There's usually a *fitness* function simulates the evaluation of the environment. The different individuals would receive different marks in this function to demonstrate their adaption to the pre-defined conditions. A *select* method simulates the natural selection for each generation of individuals. The ones with better fitness to the condition would be more likely to be selected to reproduce the children. Another function called *crossover* illustrate the process of combination of 2 chromosomes from 2 parents (usually one individual has many features, and some comes from one parent and some comes from another). The last necessary function is *mutate* which demonstrates the process of individual variation. Usually the possibility for mutate is very low and it could introduce some new feature to the individuals. Sometimes the new feature would lead to a better adaption to the condition but on most circumstances, it doesn't receive good result.

3.2 Pruning Algorithm

Sometimes, after the training of neural networks, there might be some redundancy for the functionality of weights in neurons. The redundancy here means the repetition for neuron's functionality. Though the redundancy doesn't introduce some bad feature to the neural network, it still causes higher calculation resource requirement than the pruned network. In this experiment, we mainly focus on two kinds of redundancies. One is "two neurons work in the same way" and the other is "two neurons hold opposite effect". For the first one, we should add one neuron's weight to another and delete it. For the second one, we should delete both of these two neurons (Gedeon & Harris, 1991). In this way, when we use this model to predict the new data, there's a reduced among of calculation and very close result which saves much calculation resources and time.

For example, there is 64 neurons in one hidden layer, 63 of them would lead the result to a different orientation while the remain one would lead the result to a similar orientation of the other 63 neurons. In this way, we could cut the neuron off to reduce the calculation for prediction. In that example, we could calculate one less neuron than the previous network.

In our experiment, the pruning technology we use is to calculate the angle between each two of the weighs pair in the same network layer. The angle between two weight vectors could be calculated from the formula below

$$\cos\theta = \frac{\boldsymbol{a}\cdot\boldsymbol{b}}{|\vec{a}||\vec{b}|}$$

And after that, we could set on a threshold to define what is the similar neurons. In our experiment, 2 weights with angle less than 25 we regard them as the similar weights and 2 with angle larger than 155 as opposite weights.

4 Experiment Design

4.1 Dataset and pre-processing

The dataset has 16 columns. The first five columns consist of the students' id, program id and other admin information. In this experiment, we think the second column *Crse/Prog* and fifth column *Tutgroup* are useful. The students doing different course and program are different because various programs have various difficulties. Also, the different tutor has different ways to teach the students. The knowledge comprehension of tutor also influences the students. The last 11 columns are marks from different assessments, we certainly should take them into consideration.

The contents of *Crse/Prog* and *Tutgroup* columns are the name for courses, programs or tutorial groups. So, we use One-hot method to deal with the categorized information. One-hot uses a n*1 vector to represent a n categories variable. For each of the category, the corresponding column is marked as 1 as others are 0s. There's total 23 different *Crse/Prog* and 11 different *Tutgroup* consists of 34 input features.

As for the 11 assessment, the 10 of them are from labs, tutorial assignments or mid-term exams etc. We regard them as input features and number of input features become 44. And the last column, final marks is what we should predict in this

experiment. We use the principles to divide them into 4 categories *F, P, C, D* and use one-hot to encode them. The output of the neural network would be a 4-column vector in this way.

Also, there are some row contains unavailable entries and we regard them as zero marks.

4.2 Neural Network Structure

We define a fully connected network with 1 input layer with 44 units, 1 outputs layer with 4 units and 2 hidden layers. The number of units in each hidden layer would be the hyper-parameters learned from the evolutionary process.

4.3 Evolutionary Algorithm

For the evolutionary algorithm, we defined 5 hyper-parameters. We set the population size as 10 which means the number of individuals in each generation. The cross rate 0.7 means the possibility for individuals to share chromosomes is 70%. Mutation rate 0.02 means the individual have 2% possibility to variate their parameters. Generation number is 50 means the algorithm would evolve 50 generations to get the final "specie". At last, is the range for each parameter:

- number of units in hidden layer 1
- number of units in hidden layer 2
- learning rate
- epochs number

These four are the hyper-parameters for neural network to be learnt from the evolutionary process.

4 Result

The **Fig1** shows the result for executing the evolutionary algorithm. Inthe fifth sub image, we can clearly see that the lowest loss among different network structure and different training process happens at the 30^{th} generation. And from other sub images, we could find that the network structure with lower training loss has become stable after the 10^{th} generation. So, we extract the structure information for the best structure in the 30^{th} as our final hyper-parameters for classification neural network, which is also the same as any other structure from the 10^{th} generation.

The exact number of the hyper-parameters are:

- number of units in hidden layer 1 = 28
- number of units in hidden layer 2 = 60
- learning rate 0.0176
- epochs number = 319

After that, we fed these hyper parameters into our classification neural network. **Fig2** shows the loss, training accuracy and testing accuracy for the network. We can see that there's 3 of 328 neurons were pruned.

The comparison of accuracy for previous work was listed in **Table 1**

*(Figure 1, 2 and Table 1 are in the following page)

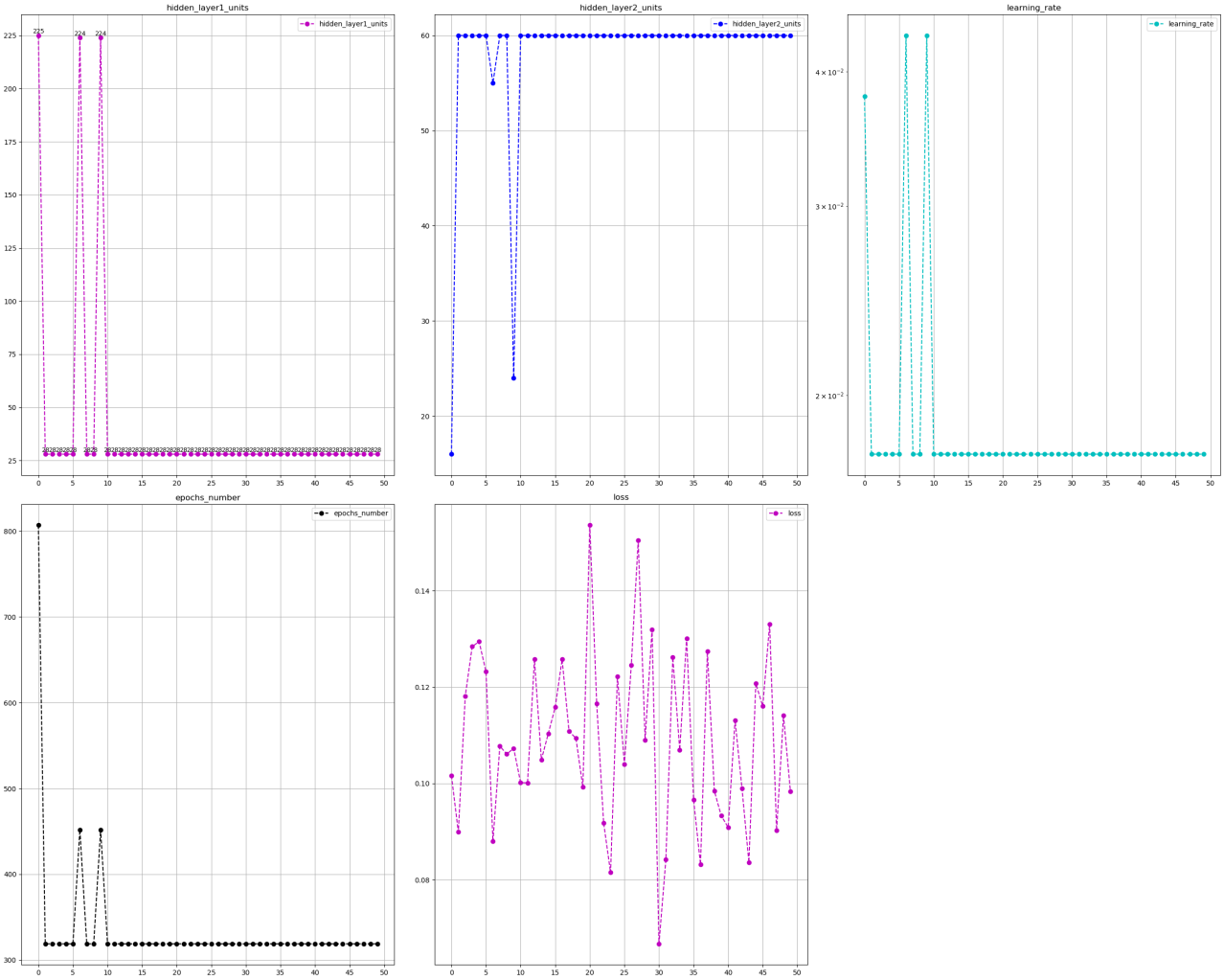


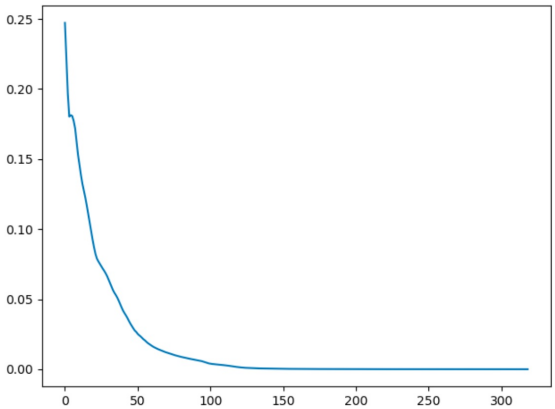
Fig. 1. The best structure for each generation in 50 generations. The horizontal axis represents the generation while the vertical one represent the structure value

Table 1. Experiment Result on test set

Model number	Accuracy
Our Model	67%
Best result*(Choi & Gedeon, 1995)	64.1 %

*The best result from the paper *Comparison of Extracted Rules from Multiple Networks on the same dataset*

Table 1. Comparison of the accuracy between the best result of Tamas' work and Our model



```
Accuracy of the model on the training data : 100 %
test loss: 0.140997
Accuracy of the model on the test data : 67 %
(3, 328)
test loss: 0.162802
```

Fig.2. Training result for the network with algorithm defined hyper-parameters

5 Discussion

After applying on the evolutionary algorithm, we reduced much time initialize and adjusting the parameters for the neural network and we could train the network directly. This proves that the evolutionary algorithm is effective when using it for predicting the hyper-parameters for neural network. But the time for executing the evolutionary algorithm is quite long in comparison of the network training. Because for each of the generation in evolutionary algorithm, 10 networks have been trained to be evaluated.

After that, we applied the hyper-parameters result directly on our classification network, and the result is better than the previous work once it was trained. This appeals that the efficiency of hyper-parameters the evolutionary algorithm sets might be beyond the human experience. We could use this to replace the manual adjusting parameters in the future.

After that, the pruning algorithm pruned less than 5% of the weights (3 of 88). So, we assume that the pruning algorithm might not be useful for a small amount of network. The size of the data set is smaller than 200 entries and there's only 88 neurons in this network. This pruning algorithm should be useful in a larger network.

6 Conclusion and Future Work

In our experiment, we approved that the evolutionary algorithm would be helpful to find the suitable hyper-parameters for neural network. Though it takes a long time to do so, but as the growth of calculation ability, the time to execute this algorithm would be less and less. Also, the evolutionary algorithm could liberate the researchers to do something more meaningful.

As for the network reduction technique. The successful reduction sample proves that this technique would be helpful to reduce the calculation work in practice. Also, we suggest this technique may performs better on a dataset with more entries could be simulated by a larger network.

In the future, we should apply the evolutionary algorithm and pruning algorithm to much more complex neural networks like CNN, RNN and LSTM to prove that the ability of setting he hyper-parameters, also the ability for pruning algorithm of reducing the calculation

References

Choi, E.C.Y. and Gedeon, T.D., 1995. Comparison of extracted rules from multiple networks. In Proceedings of ICNN'95-International Conference on Neural Networks (Vol. 4, pp. 1812-1815). IEEE.

Gedeon, T.D. and Harris, D., 1991. Network reduction techniques. In Proceedings International Conference on Neural Networks Methodologies and Applications (Vol. 1, pp. 119-126).