Parameter Optimization in Backpropagation Evolutionary Algorithms

Sze Shan Yap

School of Computer Science and Engineering
Australian National University
U6859351@anu.edu.au

Abstract. *When building the perfect neural network, one has to consider the best architecture to suit the network. One of the main difficulties is choosing the correct parameter as tuning the parameter is essential for the good algorithm performance. In this paper, we consider evolving the network with a goal of achieving optimal parameters by applying evolutionary algorithm. With a measure that able to compare the fitness value of each neural network with different parameter, the best candidate is chosen and used to breed the next generation. Generally, this can help to improve the whole performance of neural network as the appropriate set of parameters is used. We conclude by discussing the performance of backpropagation neural network with and without applying evolutionary algorithms.*

Keywords: *Parameter Tuning, Evolutionary Algorithm, Parameter Optimization, Network Performance, Roulette Wheel Selection*

1 Introduction

Genetic Evolutionary Algorithms is the method of optimization which commonly used to generate the high-quality solutions to optimization and search problem by relying on bio-inspired operators such as selection, crossover and mutation [1]. Neural network has helped us to solve many problems. But there are some difficulties could not be solved by neural network itself where the only value could not be learned is hyperparameter. So, we decided to use genetic evolutionary algorithms to learn the best parameter of neural network. We do not have the worry about knowing the correct hyperparameter as they can be learned using the evolutionary algorithms. This can help to improve the whole performance of the neural network. Optimize the neural network by applying genetic algorithms is very popular nowadays because parameter are values required by the neural network to perform properly.

In this paper, we will evolve the network with a goal of achieving the optimal parameter by applying evolutionary algorithm. Generally, we assume the backpropagation network with two layers of neurons. We start with some kind of

initial random values for the variables used in the experiment. Because these values may not be the best ones to use, we should optimize them until we get the best ones through evolution [2]. To select the best candidate, the fitness function is used. The higher the fitness value, the higher quality of set of parameters. After the best candidate is chosen, we will breed the child and bring it to the next generation.

We expected to get the better quality of the offspring than its parents as we are preventing the situation where the bad individual from generating more bad individual [2]. By keeping selecting and mating high-quality individual, there will be high chances to just keep good properties of the individuals and leave out bad ones [2]. Finally, this will end up with the desired optimal and acceptable solution [2]. It is very important to do optimization as we could maximize the efficiency and whole performance of the neural network. All of this measure we will take it as the consideration when applying the evolutionary algorithm.

2 Method

2.1 Dataset Introduction

In this paper, the authors are collecting the data which is the user interaction with the elements of the search engine results pages (SERP) on the desktop monitor by evaluating the search performance [3]. The data is collected by conduct different experiment with different types of people. They investigate the effect of two control types which is horizontal pagination and vertical scrolling [3]. After that, they were compared with which type of scrolling the user prefer. The data comes with 14 columns which include type, target position, subject, time to first click, time to right click, total time on SERP, task completion duration, accuracy, satisfaction, task start, 1st click, right click, task end, scroll and time with wrong web. Since the purpose of the assignment is predict one of the columns of data given, so the target data that I aimed for is "Type". There are 2 main categorical value included which is horizontal and vertical.

2.2 Recap on previous backpropagation neural network

On the previous neural network, we were doing the backpropagation neural network with the initial setup of input neurons = 10, output neurons = 2, hidden neuron = 20, learning rate = 0.01 and number of epoch = 500. For the input neurons, initially it has 14 columns, I have removed 4 of the unnecessary data which is 'Time with wrong Web', 'Subject', 'task Start' and 'Task end'. The reason of remove the column 'Time with wrong Web' is because no matter you scroll horizontally or vertically, there is also a chance of going into the wrong web, so I prefer to remove it. Besides, the reason of remove 'task Start' and 'Task end' would be there is a column called 'Task Completion' in the dataset, the network only need to know how much time they have spent on the search engine result page instead of the start time and end time. By removing the unnecessary data, we are able to lower down the chance of the network getting overfitting.

After that, we split the data into training and testing set and start to train the network. We create Tensor to hold both of the value of input and target. We also performed on the backward pass to adjust the weight and biases of the neuron. The level of the adjustment is determined by the gradient of the cost function. After that, we will perform the testing to test on the performance of the current backpropagation neural network. We also performed the confusion matrix to evaluate the result on both training and testing. We also done the k-fold validation, k = 5 to ensure that accuracy result that we get is reliable.

Furthermore, a few tests that I have tried is I tested on different number of hidden neuron and compare on its average accuracy. For example, I tested on 10 hidden neurons, the average accuracy is 44.107%, I tested on 20 hidden neurons, the average accuracy has been increased to 46.218%. So, I believed that the higher of the hidden neurons, the higher the average accuracy would be. For this method, we did not perform pruning.

We take the current neural network compare with the neural network that have the same set up but the only difference is we perform pruning on the second neural network and we get the result where the whole performance of neural network with pruning is much more better than the neural network without pruning.

2.3 Evolutionary Algorithms Implementation

For the neural network that perform evolutionary algorithm, we use the initial setup where the number of bits in DNA = 10, population size = 10, DNA crossover probability = 0.8, mutation probability = 0.002 and generation size = 5. We start with initialize the population where the number of populations is the number of solution available. Each of the solution is called chromosome and each gene representing the parameter which consist of 0 and 1. The meaning of 0 is represent the parameter can be drop while meaning of 1 represent the parameter remains. We will randomize the value 0 or 1 to the genes in each chromosome.

Also, each of the chromosome has fitness value. In order to choose the best individual, the fitness function is used. The fitness function for the classifier neural network is the accuracy [4]. The result of the fitness function representing the quality of the sets of parameters. Once the selection probabilities have been assigned, sampling method using roulette wheel is required to populate the mating pool [5]. Selection of the best individuals based on their quality is applied to generate which called as mating pool where higher quality individual has higher probability of being selected in the mating pool [2]. The individual in the mating pool is called as parents. Each pair of parents selected from the mating pool will generate one child. For this dataset, we will choose 2 pair of parents and they will generate total of 6 child in order to maintain the population size. The method of selection is called as Rank-Based Selection. Rank-based selection is the selection strategy where the probability

of the chromosome being selected is based on its fitness rank relative to the entire population [5].

The rank for an individual may be scaled linearly using the following formula.

$$Rank(Pos) = 2 - SP + \left(2.(SP-1).\frac{(Pos-1)}{(n-1)} \right)$$

[5]

By just mating the high-quality individual, we are expected to get the better quality of offspring. This is because we prevent the situation whether the bad individual from breed the more bad individual [2]. By keeping the selecting and mating high-quality individuals, there will be high chances to just keep good properties of the individuals and leave out bad ones [2]. After that, we repeat the whole process start from evaluating the fitness value of each individual until it met 5 generation.

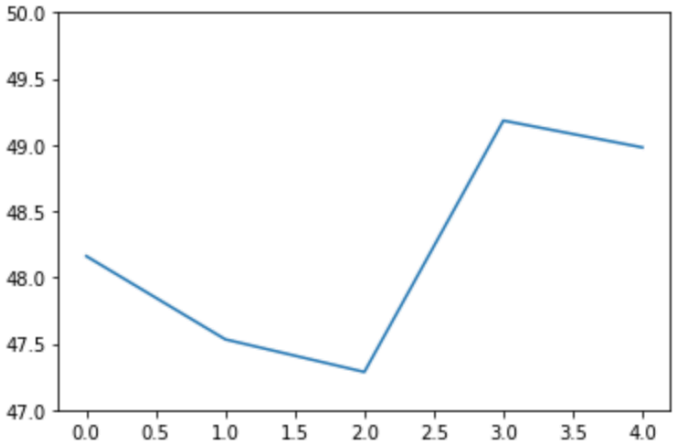
3 Results

The table below shows the average fitness value on each generation.

Generation	**Average Fitness Value**
Generation 0	48.16%
Generation 1	47.53%
Generation 2	47.28%
Generation 3	49.18%
Generation 4	48.98%

Table 1.0

And we come out with the plot graph with the average fitness value above which represent the generation of the network.



Overall, we can see that the fitness value is increase throughout the generation. This means that by using the evolutionary algorithms, the neural network is getting better in using optimal parameter.

Type	**Backpropagation Network**	**Evolutionary Algorithm Network**
Average Accuracy	46.52%	48.22%

Table 2.0

The table above shows the average accuracy between backpropagation network and neural network using evolutionary algorithms. It clearly shows that the whole performance of network with evolutionary algorithms is much better than backpropagation network. The backpropagation come out with the average accuracy with 46.52% while evolutionary algorithm network come out with 48.22%. The reason is we adjusted the parameter on the neural network. The backpropagation network does not used the optimal parameter, but the evolutionary network does. The optimal parameter actually help to improve the whole performance of the neural network.

4 Conclusion and Future Work

By comparing the backpropagation network and evolutionary algorithm network, we can see that the whole performance of evolutionary network is much better than the original backpropagation network. As we have adjusted the parameter of the neural network, the optimal parameter used on the evolutionary network to achieve better performance. The overall accuracy has been increasing a lot in order to achieve the efficient neural network.

The future work could be provided additional dataset for this neural network as the dataset is still not sufficient enough to achieve even better performance. If more useful dataset is provided, the performance of predicting the result might be more better and accurate.

5 References

1. Harvey, M. (2017). Let's evolve a neural network with a genetic algorithm—code included. Retrieved 29 May 2020, from https://blog.coast.ai/lets-evolve-a-neural-network-with-a-genetic-algorithm-code-included-8809bece164

2. Gad, A. (2018). Introduction to Optimization with Genetic Algorithm. Retrieved 31 May 2020, from https://towardsdatascience.com/introduction-to-optimization-with-genetic-algorithm-2f5001d9964b

3. Jaewon Kim, Paul Thomas, Ramesh Sankaranarayana, Tom Gedeon, and Hwan-Jin Yoon. 2016. Pagination versus Scrolling in Mobile Web Search. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM '16). Association for Computing Machinery, New York, NY, USA, 751–760. DOI:https://doi.org/10.1145/2983323.2983720

4. Ahmed Fawzy Gad 'Practical Computer Vision Applications Using Deep Learning with CNNs'. Dec. 2018, Apress, 978-1-4842-4167-7

5. Razali, N. M., & Geraghty, J. (2011, July). Genetic algorithm performance with different selection strategies in solving TSP. In *Proceedings of the world congress on engineering* (Vol. 2, No. 1, pp. 1-6). Hong Kong: International Association of Engineers

6. T. D. Gedeon, "Indicators of hidden neuron functionality: the weight matrix versus neuron behaviour," Proceedings 1995 Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, Dunedin, New Zealand, 1995, pp. 26-29, doi: 10.1109/ANNES.1995.499431.