Minimalizing sperm concentration prediction error:

Optimizing multi-layer perceptron topology with genetic algorithm

Kuangyi Xing
Research School of Computer Science, Australian National University
U5817313@anu.edu.au

Abstract. This paper will optimize the topology of multi-layer perceptron based on the previous model with genetic algorithm. However, since each running time costs a lot, such as an option with 50 max generation size and 31 population size takes 23 hours to get the result, the exact optimal topology and prediction error can not be obtained. However, with the genetic algorithm, some assumptions or guesses in the previous work of Xing (2016) can be confirmed.

Keyword: multilayer perceptron, sperm concentration, genetic algorithm, optimization

1.Introduction

Genetic algorithm is a part of evolutional computing, an optimization process, aim of which "is to improve the ability of an organism to survive in dynamically changing and competitive environments" (Gedeon, 2016). According to Gedeon (2016), the main components of an evolutional algorithm is an encoding of the problem as a chromosome, a function to evaluate the fitness, or survival strength of individuals, initialization of the initial population, selection operators and reproduction operators. In this experiment, a neural network with different size of hidden layer can be considered as an individual. The fitness function is calculating the prediction error of neural network with different topologies with 10^{th} cross validation, built in my previous work (2016). In this paper, the two-layer perceptron and three-layer perceptron will be discussed separately as before. The initial population size and max generation size will be set by Matlab command line. Other parameters such as selection operators and reproduction operators will be set as default.

2. Method

To optimize topology of two-layer perceptron, the prediction error of network will be recognized as fitness function. The hidden layer size will be set to unknown number x. The initial population size and max generation should be set manually. Because with the default initial population size and max

generation, getting a result would probably take four or five days with the computer running constantly and taking up most speed of computer, which is a big cost under my current circumstance. For example, if we want to set the initial population size 10 and max generation 20. The command line will be like ”options=optimoptions('ga','PopulationSize',10,'MaxGenerations',20)”, followed by output command line template “[x, fval,exitflag] = ga(fitnessfcn, nvars, A, b, [], [], LB, UB, nonlcon, IntCon, options)” that minimizes with integer constraints and with the default optimization parameters replaced by values in the options structure.)”. X means best point that genetic algorithm located during its iterations, fval means fitness function evaluated at x and exitflag is an integer giving the reason genetic algorithm stopped iterating. In this paper, the exitflag is always 0, which means that the reason stopping iterating is max number of generation exceeded because big number of generation size will cost a lot. Because the size for hidden layer must be positive integer and the optimization parameters are modified. "ga" means the genetic algorithm. In this experiment, the template will be applied as “[x,fval,exitflag] = ga(@assignment1_2layers,1,[],[],[],[],1,31,[], 1, options). "Assignment1_2layers" is the name of the previous function of calculating prediction error, i.e. the fitness function in this situation. From left to right, "1" means the number of variable is only one, ",1,31" means that the variable will vary between 1 and 31, the last "1" means that the first one variable will be set as integer and options has been set before.
In the three-layer perceptron, the method to set the options is same as two-layer perceptron, while the output template will be applied as "[x,fval,exitflag] =ga(@assignment1_3layers ,2,[],[],[],[],[1 1],[31 31],[],[1 2],options)". The "2" means the number of variable is two Since both the size of first and second hidden layer will be unknown numbers. "[1 1],[31 31]" means that both two variables will vary from 1 to 31. "[1 2]" means that both variables are integers.

3. Result and discussion

The result of each calculation is shown in figure 1 with different initial population size and max number of generation.

Population size	11	21	31	31	11	11	31	5
Max generation	5	5	5	5	10	20	50	50
x	6	20	26	16	14	2	20	31
fval	0.18	0.19	0.19	0.19	0.17	0.18	0.16	0.2
exitflag	0	0	0	0	0	0	0	0

Fig.1

Based on the data in figure 1, a line chart figure 2 can be generated and illustrate the trend more directly.

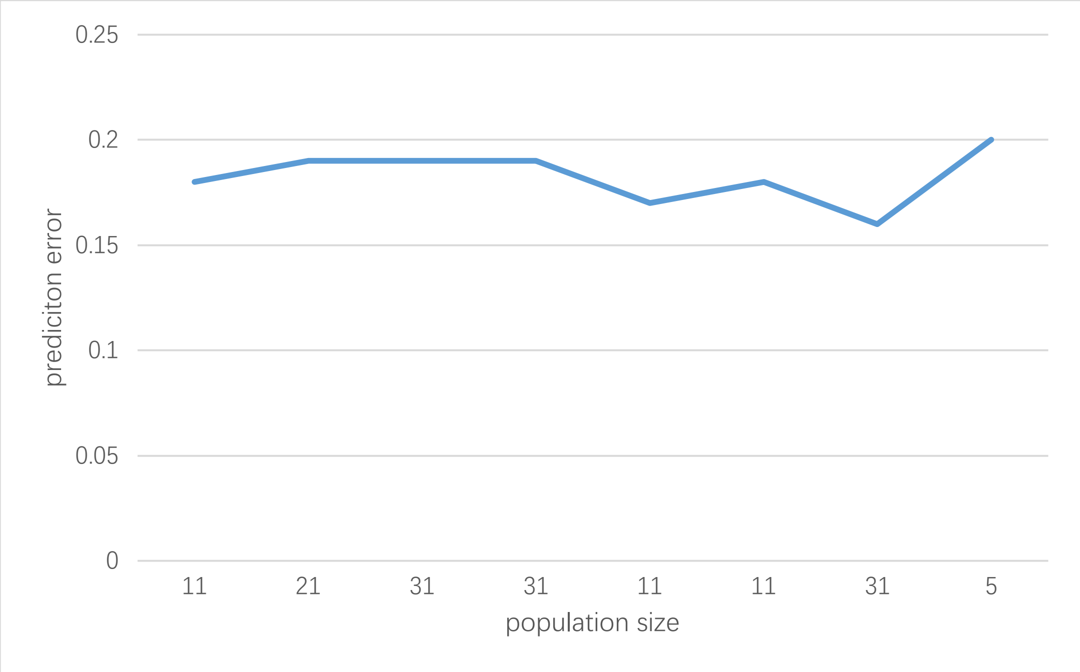

Fig.2. The order of *population size* is the same as figure 1. The “x” and “max generation” element are not shown in this line chart explicitly but they can be referred to according to figure 1 based on the same order of *population size*. The *prediction error* is “fval” in figure 1 namely.

In Gil et al. work (2012), they argued that the optimal solution for hidden layer is 21 neurons with prediction error 0.14. The proximal result in figure 1 should be the penultimate outcome with optimal topology of 20 neurons and minimal prediction error of 0.16. The initial population size and max generation size for the penultimate running are 31 and 50 separately. However, this result ran about 23 hours and the exit reason was still max generation exceeded. Therefore, we can speculate that given enough max generation and enough initial size, the work of Gil et al. (2012) can be confirmed.

In figure 1, the optimal topology varies without any rule. This phenomenon is the similar as my previous work (2016) that “the figure in this paper does not indicate and imply that 21 is the optimal size for hidden layer” In addition, in my previous work (2016), I argued that “the prediction error rate has slight fluctuation between 0.14 to 0.19 and is steady around 0.16”. In this experiment, the rate fluctuates between 0.16 and 0.2 and is steady around 0.18. However, the reason leading to the difference may be that the previous conclusion was only based on five times of running and the too small population size or generation size will constraint calculation. The latter reason can be observed most obviously in the gap between the last and the penultimate result. The last parameters are only five population size but huge generation size and the optimal prediction error is 0.2, much bigger than 0.16 in penultimate result.

The running result for optimal topology of three-layer perceptron is shown in figure 3

Population size	3	5	6	7	8	9	5	6	5	9
Max generation	10	10	10	10	10	10	15	15	20	20
Size of the first hidden layer x(1)	6	28	10	9	8	18	30	10	17	20
Size of the second hidden layer x(2)	12	4	17	13	22	22	1	16	29	22
fval	0.12	0.11	0.16	0.11	0.14	0.12	0.11	0.14	0.11	0.12
Exit flag	0	0	0	0	0	0	0	0	0	0

Fig. 3

According to the data in figure 3, a line chart figure 4 can be generated and illustrate the trend more directly.

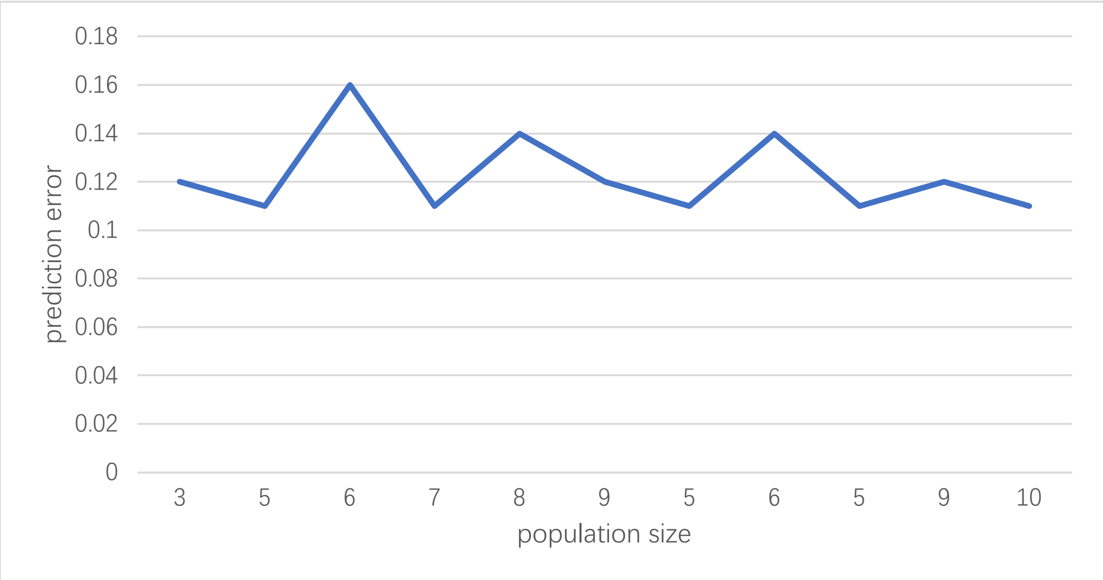

Fig.4. The order of *population size* is the same as figure 3. The “x(1)” and “x(2)” and “max generation” element are not shown in this line chart explicitly but they can be referred to according to figure 3 based on the same order of *population size*. The *prediction error* is “fval” in figure 3 namely.

In three-layer perceptron, the prediction error is reduced, the same as my previous conclusion that “the performance of three-layer perceptron is better than two-layer perceptron generally.” and similar as “the prediction error rate is steady around 0.13 with 0.11 optimal”. However, the optimal topology is still uncertain without enough population and generations. So does the minimal prediction error. The exiting reason are all max generation exceeded.

4. Conclusion and future work

The genetic algorithm fitted much guess in previous work with a more convincing method. In two-layer perceptron, the possibility of the conclusion of Gil et al (2012) that 21 is the optimal size and 0.14 is the minimal prediction error is correct is increased but a more precise result should be provided. The biggest limitation for this experiment is that the adequate generation size and population size can not be obtained. A possible solution for this problem may be degrading the 10^{th} cross validation to the 3^{th} cross validation or other less cost validation method. Though the accuracy may be decreased, the cost is decreased as well. However, from my perspective, the best solution for this problem is to calculate with a super computer.

Reference List

1. Gil, D, Girela, JL, Juan, JD, Gomez-Torres, MJ, & Johnsson, M (2012). 'Predicting seminal quality with artificial intelligence methods', *Expert Systems with Applications*, vol.39, no.12, pp.12564-12573
2. Gedeon, T (2016). *'Introduction to Computational Intelligence'*.
3. Xing, K (2016). '*Sperm concentration prediction with multi-layer perceptron*'. Canberra.