Predicting the Depression Level from Physiological Signals with the Cascade-Correlation Network and the Genetic Algorithm

Jiahui Wu
College of Engineering and Computer Science
The Australian National University
Canberra, Australia
u6418750@anu.edu.au

Abstract. In this paper, we tried to predict the depression level from the physiological signals of the observers by using the Cascade-Correlation network and genetic algorithm. With the optimal feature selection generated by the genetic algorithm, the average test accuracy of 46% can be achieved by using cascade correlation network.

Keywords. Cascade-Correlation Networks, Genetic Algorithms, Depression Detection, Pupil Dilation,

1 Introduction

Depression is a one of the most common mental illnesses in the world. World Health Organization (WHO) stated that there are more than 200 million people suffering from depression in worldwide[1]. Even worse, the number of affected people is still rapidly growing each year. Due to lack of effective assessment, some people in low-income countries cannot receive an appropriate treatment for depression. In fact, according to WHO, people from countries of all income levels are often misdiagnosed. As a result, some healthy people may be prescribed antidepressants by mistake.

The current assessment methods for depression (such as questionnaires and interviews) are descriptive and subjective with large variation [1]. These assessment methods for depression are inaccurate, making this global problem difficult to solve. Hence, it is important to develop a low-cost and accurate method to detect depression.

According to Zhu's study [1], physiological signals may be helpful for depression detection. They tried to use the physiological signals from observers to recognize the depression level of the individuals in video. They recorded the Galvanic Skin Response (GSR), Skin Temperature (ST) and Pupillary Dilation (PD) of the observers when they were watch the video of people of different depression level.

Zhu and her co-workers used two types of models [1]. One is the general Neural Networks (NN) model. For the other one, they firstly used Genetic Algorithms (GA) to select features and trained the NN model with the optimal subset of features. The result shows that the NN model can achieve an overall accuracy of 88% when it trained with all the features. And with the help of feature selection, the overall accuracy of the GA+NN model can be improved to 92%.

In this paper, we are trying to investigate the relation between the physiological signals from the observers and the depression level of the individuals in video. We used the Genetic Algorithms to select an optimal subset from the GSR, ST and PD features, and then used the Cascade-Correlation (CasCor) networks to predict the depression level based on the selected features. This paper examines the performance of the GA+CasCor model on the depression detection in this case, and we will also discuss the measurement results and future improvements.

[1] https://www.who.int/news-room/fact-sheets/detail/depression

2 Methodology

2.1 Features from Physiological Signals

Based on Zhu's paper [1], they selected 16 videos of depressed individual from 2014 Audio-Visual Emotion Challenge dataset. And each video is associated with a single depression level.

Zhu and her co-workers collected the maximum, minimum, variance, standard deviation, mean, root mean square, means of the absolute values of the first and second difference from normalized and filtered GSR signals as the features respectively. According to Zhu's paper [1], a Very Low Pass (VLP) Butterworth filter (with a cut-off frequency of 0.08 Hz) was applied to the normalized GSR signal to generate the VLP CSR signal. And for feature extraction, they found the number of the Skin Conductance Response (SCR) occurrences for VLP, LP and normalized GSR signal, the average amplitude of all these occurrences, and the ratio of peak occurrences in VLP to those in LP. Similarly, these 23 features were also extracted from the ST signals [1].

They collected the PD signals of 12 students with EyeTribe eye tracker in 60 Hz, when the students were watching the videos of depressed individual. To extract the features of PD signal, Zhu and her co-workers normalized the data of left and right eyes respectively [1]. For the normalized data, they collected the average size, the maximum, minimum, variance, standard deviation, mean, root mean square, means of the absolute values of the first and second difference [1]. In addition, they applied a VLP Butterworth filter (with a cut-off frequency of 0.08 Hz) to generate the left, right and average VLP PD signal and collected them as the features [1]. For the left, right and average PD signal, the amplitude and number and of peak occurrences VLP and LP PD signals, and the ratio of peak occurrences in VLP to those in LP were also collected as the features [1].

Hence, there are totally 85 features extracted from GER, ST and PD signals by Zhu and her co-workers.

2.2 Cascade-Correlation Networks

The Cascade-Correlation (CasCor) architecture was introduced in 1990 [2]. One goal of this architecture was to deal with slowness of the back-propagation multi-layer networks at that time.

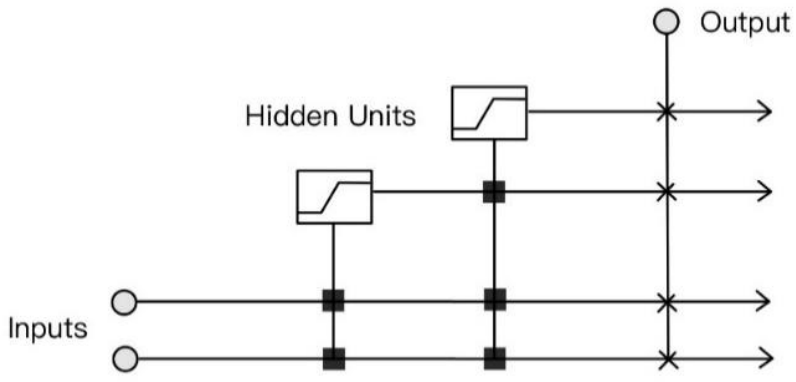

Fig. 1. The Cascade-Correlation architecture. The squares represent the frozen weights in the Cascade-Correlation network, and the cross represent the adjustable weights in the Cascade-Correlation network.

In a CasCor network, only one hidden unit is added into the network in each time. After the hidden unit is added, the

weights of this unit will be frozen, which means that all the weights of this unit will not be changed in subsequent training. Hence, the only adjustable weights in a CasCor network are the weights of the output units (as indicated in figure 1). The original method to select a new hidden neuron is based on the covariance [2]. The learning algorithm of CasCor networks is following:

1. Train the network until the loss reaching a plateau;

2. Take the input of the network and output of all existing hidden neurons as the input of a new hidden neuron;

3. Train the new hidden neuron by maximizing the covariance between its output and the of output error of the original network;

4. Freeze all the weight of the new hidden neuron and add it into the network;

5. Repeat steps 1 to 4 until the performance of model is satisfied.

In this paper, we used a variation of the CasCor networks. We modified step 3 in the learning algorithm above. In our model, the new hidden neuron will be added into the network before any training. Then we tried to minimize the loss by training the new hidden neuron and the output neuron only. Weights of the new hidden neuron will be frozen before another new hidden neuron is included into the network.

To predict the depression level from physiological signals, we set the number of outputs is 1. And the closest integer of the output is the prediction of from depression level (since the possible values of depression level are 0, 1, 2 and 3). That is, we can see this problem as a regression problem. As a result, our CasCor model was trained with the mean square error loss function. And we specified the number of epochs to be 1000 and the learning rate to be 0.003 and choose the Adam optimizer. The activation function for each hidden neuron was sigmoid function.

2.3 Genetic Algorithms

The Genetic Algorithm (GA) is inspired by the idea of biological evolution, which was introduced by John Holland and his collaborators in 1975 [4]. It is a family of randomized search algorithms. To find an optimal solution by GA, we need to firstly encode each possible solution into a unique chromosome. And the set of encoded solutions is called population. Similarly to Charles Darwin's theory of natural selection, there is some mechanism to determine the fitness of each individual in the population, and the selection in GA prefers to the chromosomes with high fitness. A typical genetic algorithm starts with an initial population, and then the algorithm will follow the steps below:

1. Evaluate the fitness of each chromosome in the current population;

2. Select the chromosomes with high fitness;

3. If the fitness of the selected chromosomes is good enough, stop and output the result;

4 Else, produce the population of next generation by crossover and mutation and go back to step 1.

To obtain a better performance, we used GA to select an optimal subset of the 85 features extracted from GER, ST and PD signals. More specifically, we encoded the selected subset into a binary array of length 85. For a given binary array,

the i-th bit is 1 means that the i-th feature is contained in the corresponding subset, and it is 0 means that the feature is not selected by this subset. At the very beginning, there were 100 chromosomes in our population, 5 of them encode the complete selection (i.e. all bits are 1 in these 5 chromosomes), and all the other chromosomes were randomly generated.

To evaluate the fitness of each chromosome, we trained a network with the corresponding feature selection for 20 times in each generation. For less running time, we chose the 2-layer NN model with 48 hidden neurons, while the number of epochs was adjusted to 500 and the learning rate was adjusted to 0.01. The fitness of each chromosome was defined as the average test accuracy of all training sessions. That means that the more times a chromosome appears in the whole evolutionary process, the more numbers of training sessions it has. With higher number of training sessions, the fitness of the chromosome would become more convincing. Again, to reduce the running time, in our GA program if a chromosome appears for more than 16 times, then we will not train the NN model and update its fitness anymore, since the total number of training sessions for such a chromosome is already not less than 320, which we think is enough for training.

Elitism was used in our GA method. We passed the best 50 chromosomes in the current generation to the next generation and only updated the half with lower fitness. The rest 50% in the next generation was produced by uniform crossover between random parents. And the mutation rate for each bit was 0.5%. We allow some newly generated chromosome to have a high fitness by luck, but its fitness will converge to its real value as the number of appearances increases.

The maximal generation in this case was 100. And the final result of our GA method was determined among the best 25 chromosomes in generation 100 by democratic voting. That means for each feature, we counted the number of 1 bits among the best 25 chromosomes. If the number of 1 bits for a feature is larger than 12, then this feature will be included in the final selection. Otherwise, we will drop this feature. Hence, by voting among the chromosomes with high fitness, we obtained the optimal feature selection for our prediction problem.

2.4 Performance Evaluation

In addition to the CasCor model in 2.2, we also use a two-layer Neural Network (NN) to predict the depression level for comparsion (with the same loss function, optimizer and hidden activation function with the model above). That it, there are totally 4 models discussed in this paper. They are CasCor, NN, GA+CasCor and GA+NN.

To quantify the performance of models, we repeated the training 100 times for each choice, and then calculated the average accuracy and average loss of the test set. In this paper, we used average test accuracy and average test loss to evaluate the performance of models on this depression detection problem.

3 Results and Discussion

3.2 Feature selection generated by GA

To find an optimal subset of the features, we applied the generic algorithm and collected the fitness data for each chromosome in each generation.

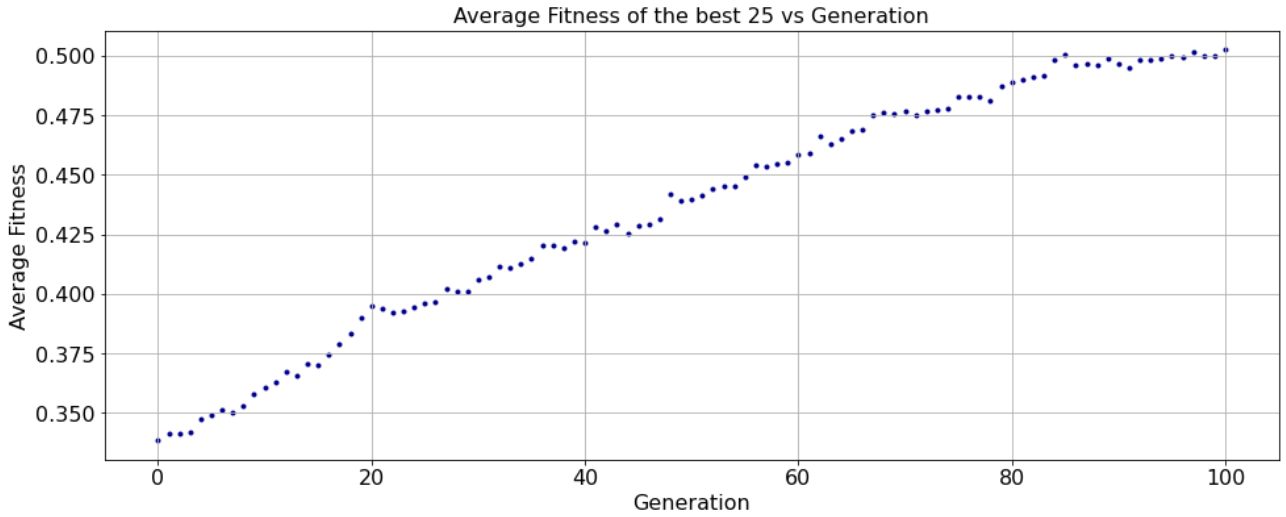

Fig. 2. The change of average fitness of the best 25 chromosomes from generation 0 to generation 100.

As figure 2 indicates, the average fitness of the best 25 chromosomes in each generation almost kept increasing in the whole evolutionary process. It started at around 0.34 at the very beginning, and finally reach around 0.5 between generation 90 and generation 100, which implies that the fitness of the whole population was significantly improved by our generic algorithm.

By the voting among the 25 chromosomes in generation 100, we included the following features in our optimal selection:

Table 1. The feature selection generated by the generic algorithm.

rms_normalised_gsr	max_normalised_skintemp
min_filtered_gsr	mean_normalised_skintemp
max_filtered_gsr	var_normalised_skintemp
rms_filtered_gsr	rms_normalised_skintemp
first_diff_filtered_gsr_abs_mean	min_filtered_skintemp
num_normalised_gsr_peaks	mean_filtered_skintemp
num_filtered_gsr_peaks	rms_filtered_skintemp
ratio_scr_occurrence_vlp_lp	vlp_skintemp_peak_occurrences
mean_normalised_pupil_left	mean_vlp_skintemp_peak_amplitudes
first_diff_normalised_pupil_left_abs_mean	max_normalised_pupil_avg
mean_normalised_pupil_right	mean_normalised_pupil_avg
rms_normalised_pupil_right	first_diff_normalised_pupil_avg_abs_mean
second_diff_normalised_pupil_right_abs_mean	second_diff_normalised_pupil_avg_abs_mean
min_normalised_skintemp	

3.2 Performance measurement

According to feature selection in table 1, we measured the measured the average test accuracy and average test loss of GA+NN model by changing the number of hidden neurons from 1 to 50, and then we compared the performance of the GA+NN model with the performance of the NN model.

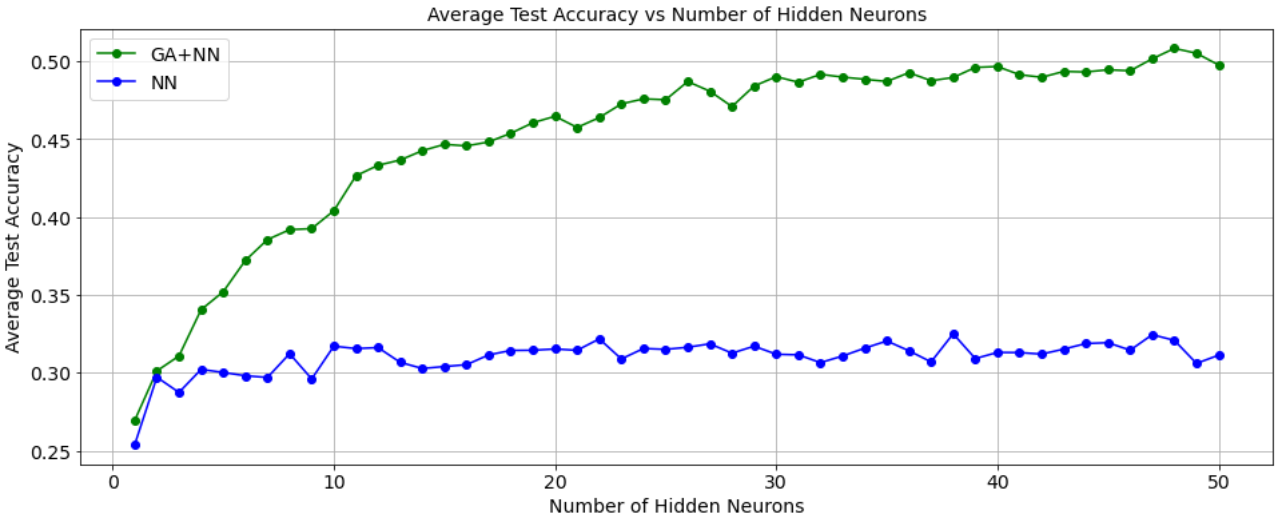

Fig. 3. The change of average test accuracy as the number of hidden neurons increases from 1 to 50.

As shown in figure 3, the average test accuracy of NN converges to around 0.32 as the number of hidden neurons becomes larger. By contrast, the average test accuracy of GA+NN almost doubles as the number of hidden neurons increases from 1 to 50. And when the number of hidden neurons is larger than 30, the average test accuracy of GA+NN becomes stable and finally converges to around 0.5, which is consistent with the result in figure 2 (showing that the average fitness of the best 25 chromosomes in final generations is around 0.5).

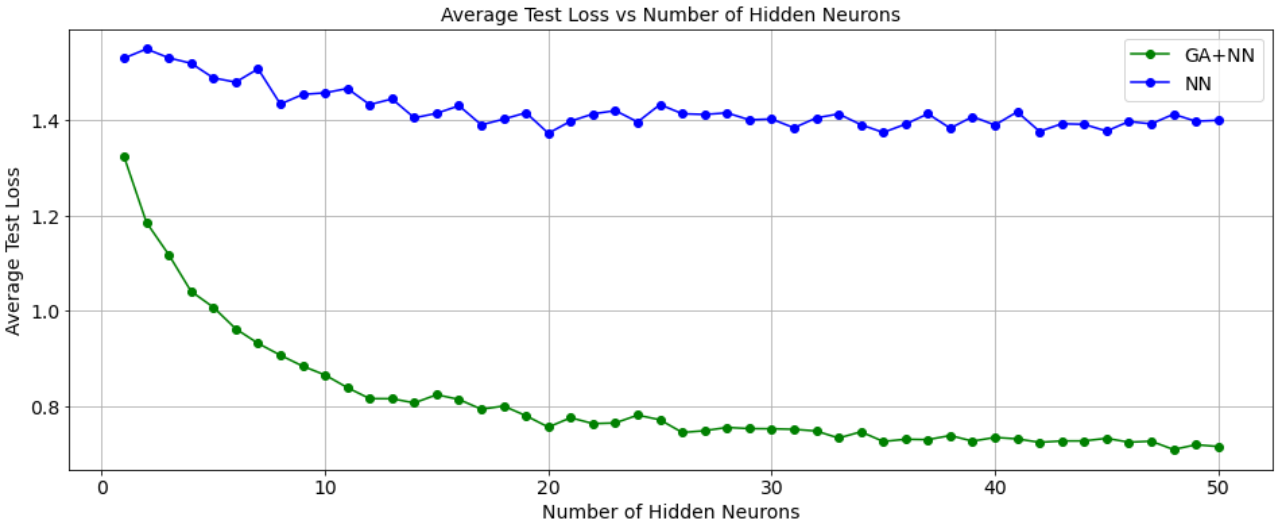

Fig. 4. The change of average test loss as the number of hidden neurons increases from 1 to 50.

According to figure 4, when the number of hidden neurons varies from 1 to 50, the average test loss of GA+NN converges to around 0.7 while the average test loss of NN converges to around 1.4. That is, the average test loss NN model can be reduced by 50% at most by using GA for feature selection.

Provided with the optimal feature selection, we measured the average test accuracy of the GA+CasCor model with varying number of hidden neurons and plotted the average test accuracy of GA+CasCor as a function of the number of hidden neurons. Similarly, we also plotted the average test accuracy of CasCor as a function of the number of hidden neurons.

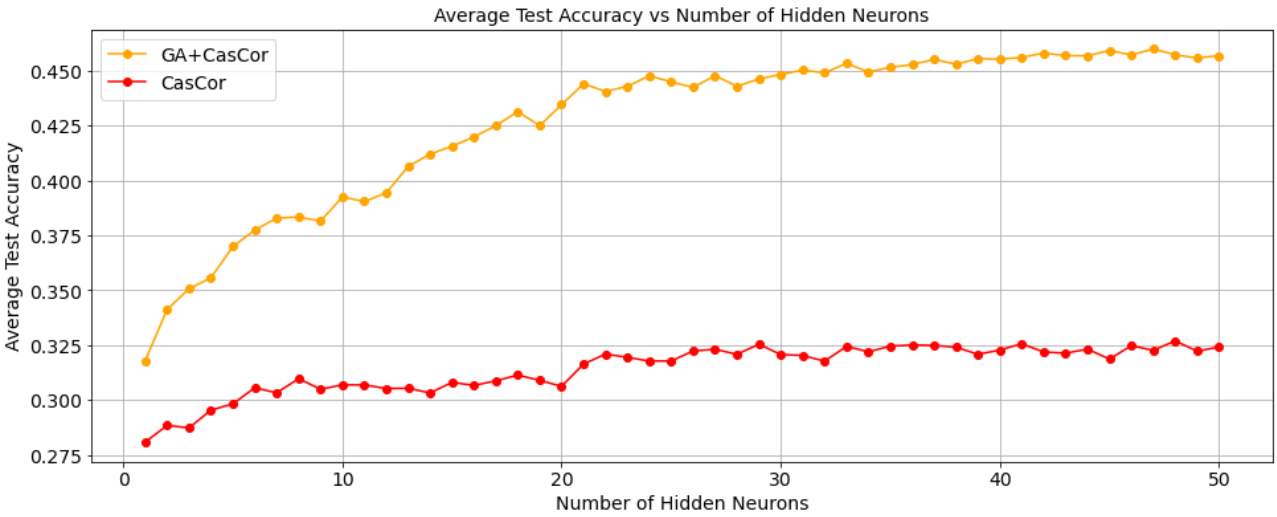

Fig. 5. The change of average test accuracy as the number of hidden neurons increases from 1 to 50.

From figure 5, we can find that the convergence of the average test accuracy of CasCor is around 0.325 while the corresponding convergence of GA+CasCor is around 0.46, which means that the average test accuracy of CasCor can be increased by the optimal feature selection.

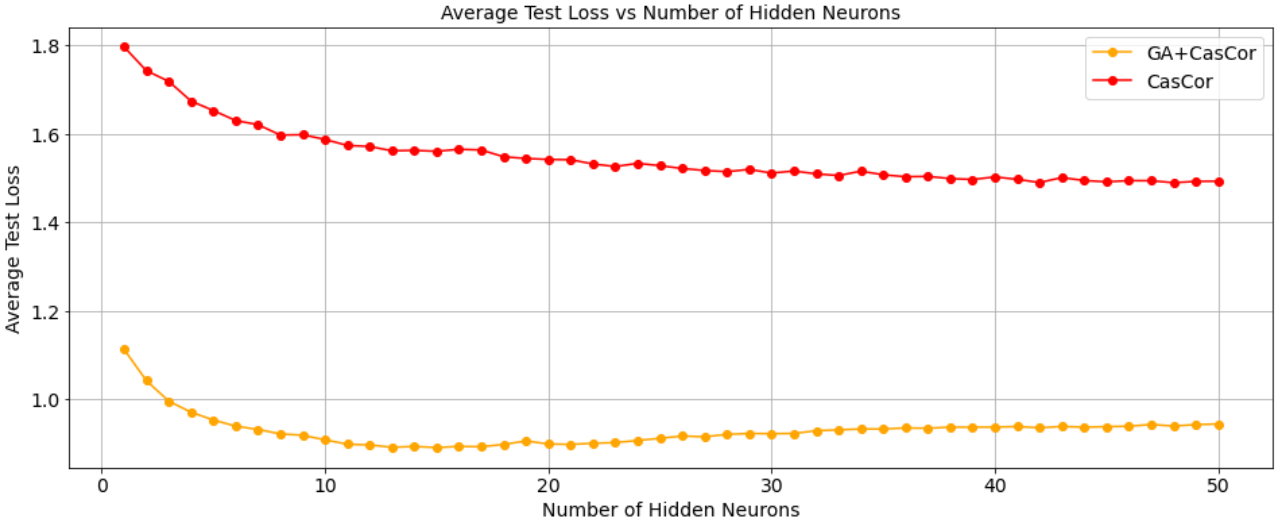

Fig. 6. The change of average test loss as the number of hidden neurons increases from 1 to 50.

A similar conclusion can be found if we measured the average test loss for both CasCor and GA+CasCor models. As figure 6 indicates, the convergence of the average test loss of GA+CasCor (which is around 0.9) is approximately 40% lower than that of CasCor (which is around 1.5). Hence, although the model that was used to calculate the fitness in GA was NN, the result selection of GA can still significantly improve the performance of CasCor.

To compare the performance of NN and CasCor under the optimal feature selection, we showed the variation of the average test accuracy of GA+NN and GA+CasCor together in figure 7 and the variation of the average test loss of GA+NN and GA+CasCor together in figure 8.

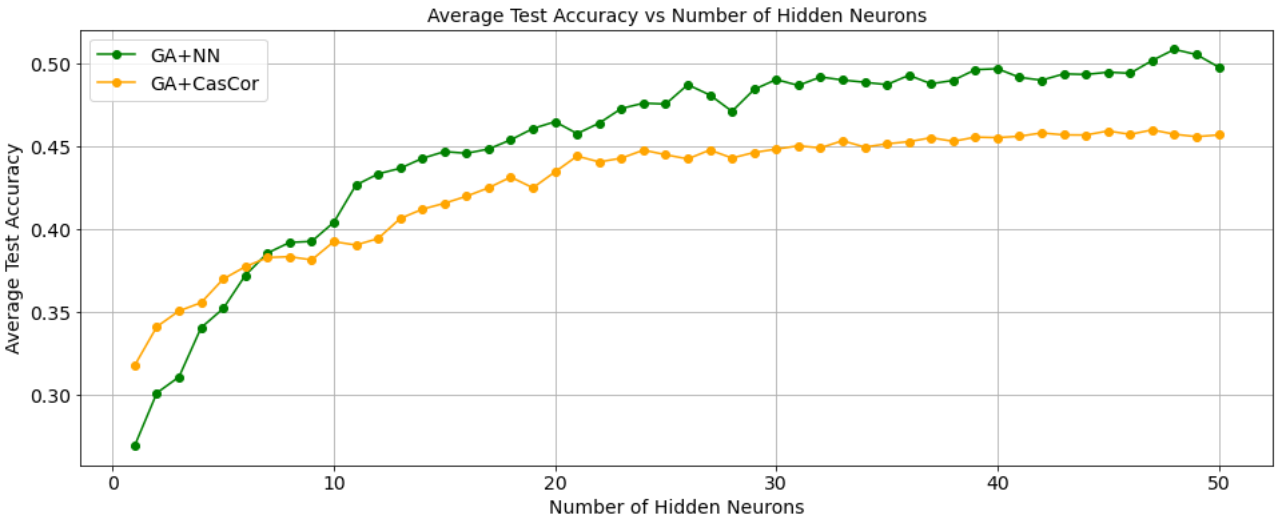

Fig. 7. The change of average test accuracy as the number of hidden neurons increases from 1 to 50.

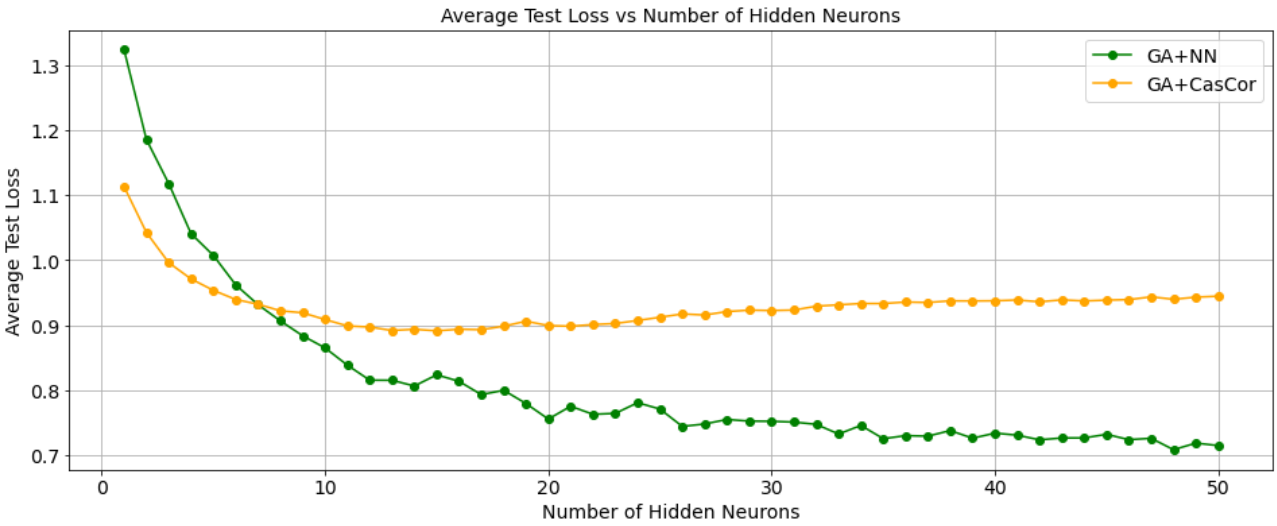

Fig. 8. The change of average test loss as the number of hidden neurons increases from 1 to 50.

By combining the results in figures 7 and 8, we found that when the number of hidden neurons is less than 6, the average test accuracy of GA+CasCor is higher than that of GA+NN and the average test loss of GA+CasCor is lower than that of GA+NN. That is, the performance of GA+CasCor is better than the performance of GA+NN when the number of hidden neurons is less than 6. However, if more hidden neurons are added into the networks, the performance of GA+NN will surpass that of GA+CasCor (with higher average test accuracy and lower average test loss).

The reason (why GA+CasCor is worse than GA+NN when more hidden neurons are included) might be that the fitness we define in our generic algorithm is the average test accuracy of NN. The feature selection we have in table is optimal for NN model, but it might be not the best option for our CasCor model. To solve this problem, we can evaluate the fitness with the average test accuracy of CasCor and find the optimal for CasCor specifically.

5 Conclusion and Future Work

As the number of hidden neurons increases, the average test accuracy of GA+CasCor converges to around 0.46 while GA+NN converges to around 0.50. And the convergence of the average test loss of GA+CasCor (which is

around 0.95) is higher than that of GA+NN (which is around 0.73). The performance of GA+NN is better than that of GA+CasCor when more hidden neurons are added into the network. This result might come from the definition of the fitness in our GA method. Hence, for future study, we can define the fitness as the average test accuracy of CasCor instead.

To further improve the performance of our model on this problem, we can increase the number of videos in the experiment can collect more data from different observers, since there were totally 16 videos for 12 observers to watch in the experiment [1].

Besides, we solved this depression level prediction problem under the assumption that the response patterns of all observers are identical. However, this assumption is not necessarily correct. What if the response patterns of different people to the depressed individuals are different? We can customize the neural network for each observer, which means that we can train a model with the data from each observer separately and find the specified response pattern of each observer. Perhaps we can achieve higher accuracy by combining the predictions from the customized model for each observer.

References

1. Zhu, X., Gedeon, T., Caldwell, S., & Jones, R. (2019). *Detecting emotional reactions to videos of depression.* INES'19: IEEE 23rd International Conference on Intelligent Engineering Systems.
2. Fahlman, S. E., & Lebiere, C. (1990, June). *The Cascade-Correlation Learning Architecture.* School of Computer Science, Carnegie Mellon University. https://proceedings.neurips.cc/paper/1989/file/69adc1e107f7f7d035d7baf04342e1ca-Paper.pdf
3. Treadgold, N. K., & Gedeon, T. D. (1997). *A cascade network algorithm employing Progressive RPROP.* School of Computer Science & Engineering, The University of New South Wales. http://users.cecs.anu.edu.au/~Tom.Gedeon/pdfs/A%20Cascade%20Network%20Algorithm%20Employing%20Progressive%20RPROP.pdf
4. Whitley, D. (1994). *A genetic algorithm tutorial.* Statistics and Computing, 4, 65–85. https://sci2s.ugr.es/sites/default/files/files/linksInterest/Tutorials/Whitley94.pdf