Pruned Genetic Algorithm-Neural Network In Bimodal Stress Recognition

Kunzhi Tang

Research School of Computer Science
Australian National University
Acton ACT 2601 Australia
u6864817@anu.edu.au

Abstract. Stress is a serious problem that everyone will meet in daily life, and stress will significantly affect people's health. Therefore, stress recognition is a very important research. In this paper, we developed a neural network that is applied to classification problems, separates RGB data and thermal data, and finally merges them to identify whether people is stressed. What's more, in this paper we use threshold method to prune unnecessary neural neurons, which can increase our model efficiency. After building and pruning the neural network, genetic algorithm is added to fine tune the last layer. We believe our model will help stress recognition research.

Keywords: Network reduction techniques, Genetic Algorithm, RGB, thermal Super-Pixels, Stress Recognition

1 Introduction

Nowadays, fast-paced lifestyle brings pressure in people's daily life. Pressure can cause people's anxiety, angry or other negative emotions. It can even cause cancer and other severe diseases[3]. Therefore, stress research is very popular in the current research field. For traditional stress recognition, it is necessary to contact people's bodies to obtain physiological signals like heart rate [7] and skin resistance[6]. These methods consume lots of time. However, Sharma et al [5] proposed a method of using fusion of RGB and thermal image for pressure recognition. Irani et al [1] has developed a featured image extracted from super-pixels for bimodal pressure recognition. Sano et al [12] using wearable sensors and mobile phones to recognize stress.

In this paper, we will also use the thermal-RGB compressed video stress data set[1], this data set has the facial data of the subject's RGB and thermal. We use neural network(NN) and genetic algorithm to explore the classification problem based on this data set.

Genetic algorithm is an optimization algorithm inspired by biology[8]. It has been widely used in training machine learning algorithms, dealing with single-objective and multi-objective problems. The traditional neural network (NN) completes tasks by simulating the human brain for learning and classification[9]. Genetic Algorithms (GA) and Neural Networks (NN) are very similar, both are technologies inspired by biology, so this similarity prompted us to try to create a hybrid of the two in this article called GANN (Genetic Algorithm-Neural Network).

In this paper, after establishing a two-layer neural network, we first use threshold method to prune the neural unit, and then fine-tune it after pruning, and then use the GA to update the weights of the last layer of the network to optimize the model.

We compared several models obtained by adjusting the parameter settings, comparing the pruned and fine-tuned model and the unpruned model. The results show that the pruned and fine-tuned model has better speed and almost the same accuracy as the unpruned network.

2 Method

In this paper, we are using a compressed video stress dataset[1]. This data comes from the collection of facial data from RGB cameras and thermal imaging cameras when people watch 20 movies. Each sample contains PCA features from RGB images and thermal images, and each sample has a label "Stressful" or "Calm". we treat the label "Stressful" as 0 and label "Calm" as 1. I randomly divide the data set into a training set and a test set. We think that this data set may cannot converge since the test accuracy is always about 50%. I tried to use breast cancer data in sklearn to validate the model, so there may be no problem with the model.

In this section, we will introduce how we built the GANN network that applied to this classification problem, and how we pruned the NN network using threshold.

2.1 Methods of building Genetic Algorithm-Neural Network

The neural network we defined in this paper contains an input layer, two hidden layers, and an output layer, as shown in the figure 1 below. There is a link with a digital association called weight in each neuron between two adjacent layers[10]. In the hidden layer part of the network, each neuron has two functions. One is a linear function that linear processing of the input, and another function is an activation function. In the model of this article, the activation function is Relu and it can change the value less than zero to zero, and the value greater than zero remains unchanged. The function of the output layer is that the linear layer transforms the output size to be the same as the class size, and then using the softmax function to map the output of the neuron to the (0, 1) to express the predicted probabilities of "Calm" or "Stressful".

For hidden and output layer, batch normalization is used to transform the linear parameter value of the unit between 0 and 1, so after that we could use threshold for pruning.

In the compressive stress data set, the pressure data of people's faces is given in two parts: RGB and thermal components. In this paper, we put RGB data and thermal data separately into the NN model to extract features. Then we remove the last layer of neural network and replace it with Genetic Algorithm and use GA to make fine tune of the last layer. Thus we could get a Genetic Algorithm-Neural Network(GANN).

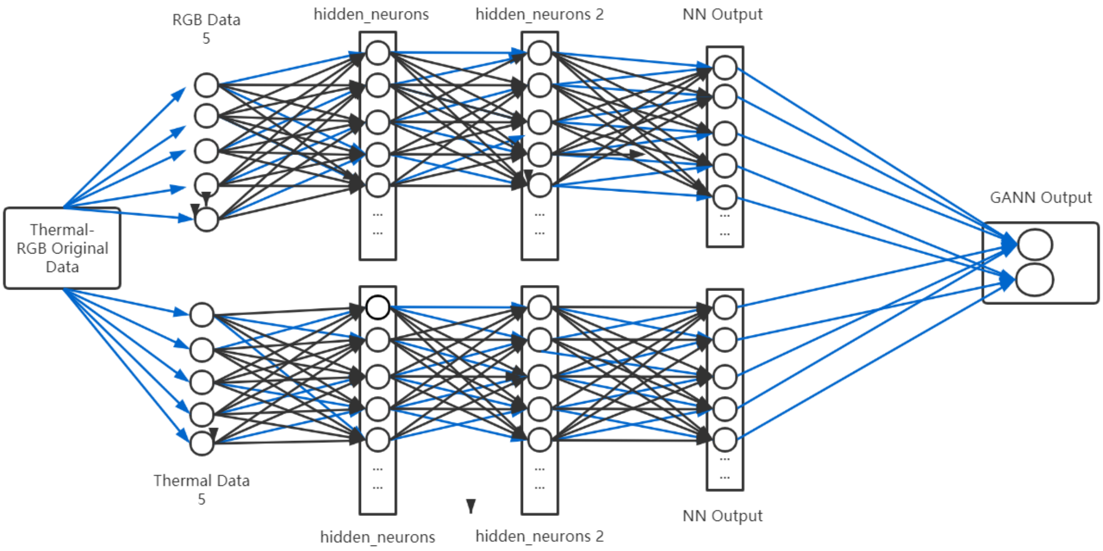

Fig. 1. The GANN Network

2.2 Methods of Pruning Neural Network

In our model, the neurons that need to be pruned are those useless neurons. The weight of its link to the output is very small, or almost zero[2].

We use the threshold method to prune such useless neurons. As each neuron in the neural network has a linear input processing function, as in formula 1:

$$y=ax+b \quad \textbf{(1)}$$

As we believe that neurons' linear classification with parameters less than threshold has little effect on the output[4], so we set the parameters (a and b) of these neural neurons to zero to conduct pruning. As we are using the Relu method, the weight of each unit will be greater than zero, so a value close to zero can be considered useless. We can achieve pruning without changing the structure of neural network.

We compared the value of each neuron's parameter with the threshold. The parameters that are less than or equal to our preset threshold will be marked as "True", otherwise they will be marked as "False". Then we set the parameter of the linear classification marked as "True" to 0 in neurons, so these neurons output weight is 0. Figure 2 showing the pruned GANN network.

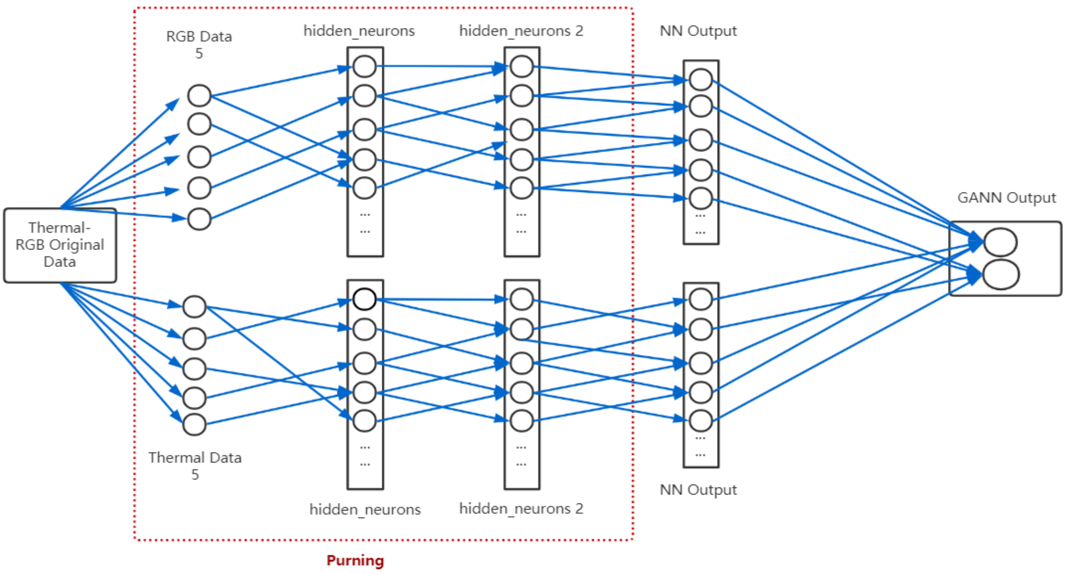

Fig. 2. The Pruned GANN Network

Compared with the figure 1, the black lines represent the linear classification of the neurons being pruned. After pruning, we will fine tune the network. During the fine tune process, we need to set the gradient of the pruned part to zero, and at last save the fine tune and pruned network.

We set up a threshold list to test the prediction accuracy of the NN network after pruning the neural network with different threshold values, and then we can select the most suitable threshold for our network pruning.

Table 1 shows the different threshold values and the number of unit-parameters pruned, also has the prediction accuracy.

Table 1. Comparison of different thresholds

Threshold	Pruned neurons parameter number	pruned Train Accuracy	pruned Test Accuracy	Pruned network with fine tune train accuracy	Pruned network with fine tune test accuracy	Loss
0.001	**106**	**87.09%**	**51.00%**	**91.90%**	**54.00%**	**0.1578**
0.01	973	78.61%	49.00%	91.71%	52.00%	0.1983
0.05	4636	52.02%	54.00%	92.29%	53.00%	0.2001
0.08	6986	49.90%	50.00%	82.65%	51.00%	0.3473
0.1	7973	49.51%	50.00%	77.047%	48.00%	0.4627

We found that the smaller the threshold value is, the higher the accuracy will be, so we choose the value of threshold in GANN to be 0.001.

2.3 Using Genetic Algorithm to Fine tune

We have used neural network for feature extraction before and used threshold to prune and have fine tune the network. After that we will remove the last layer and replace it with GANN and then use GA algorithm to fine tune last layer.

Genetic algorithm is a kind of Evolutionary Algorithm (EA), which is a classic evolutionary algorithm based on randomness. The characteristics of evolutionary algorithms are based on population evolution, survival of the fittest, and random orientation. GA is an iterative calculation process, the main steps include coding, population initialization, selection, genetic manipulation, evaluation and stopping [11].

After creating the initial population, we use the fitness function to evaluate these solutions (children and grandchildren), and then multiply the next generation of feature subsets through crossover and mutation. The higher fitness value means higher probability that the feature subset will be selected to participate in reproduction. In this way, after N generations of reproduction and elimination, the feature subset with the highest fitness function will remain in the population.

$$fitness = 1/|predicted\text{-}Label| \quad (2)$$

The genetic algorithm has three main parameters, the generation number, the number of marriages of the parents and the percent of genetic mutation. Through repeated experiments, we found more generation number give better the fitness and accuracy of the model, so we try to increase the generation as much as possible. The number of marriages of the parents should be less than the solution in population, as we set the number of solutions to 20, which can make the model converge and the training effect is better, so we set the number of marriages of the parents to 10. Finally, we set the mutation percentage of the gene to 10, which is also a general number.

3 Results and Discussion

3.1 Experiment on Pruned and Unpruned GANN Network

As our pruning target is to remove some useless neurons but remain those neurons that have impact on the output. Therefore, the pruned GANN network should have close accuracy to the original network.

We used genetic algorithm to train 3000 generations to see the accuracy difference between the pruned model and the unpruned model. As shown in figure 3, the left side is the pruned network, and the right side is the unpruned network.

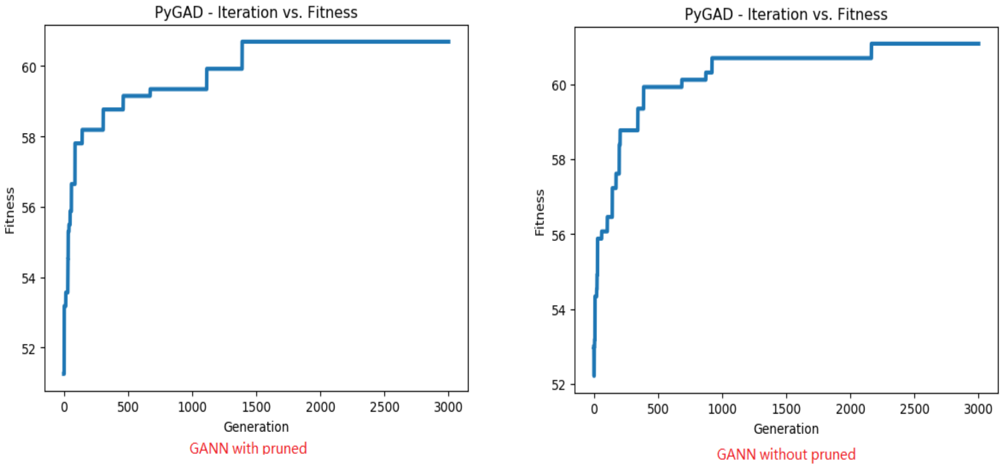

Fig. 3. Comparison of pruned network and unpruned network

The figure shows that the fitness of the two networks is very close, but the GANN network after pruning will reach the fitness threshold earlier, which means that pruning can greatly save the time of training the model and hardly affect the accuracy of the model. The table 2 lists the time and accuracy of the pruned and unpruned networks under the same conditions.

Table 2. Time spent and accuracy between pruned and unpruned

Generation	GANN With Pruned Train Accuracy	GANN With Pruned Spent Time	GANN With Pruned Test Accuracy	GANN Without Pruned Train Accuracy	GANN Without Pruned Spent Time	GANN Without Pruned Test Accuracy
100	56.26%	50s	49%	56.06%	1min 25s	50%
500	57.58%	3min 37s	49%	59.92%	4min 15s	49%
1000	60.30%	7min 12s	50%	60.69%	9min 16s	51%
2000	61.27%	16min 33s	50%	60.70%	20min 21s	52%
3000	61.84%	31min 24s	51%	61.07%	38min 56s	50%

We could find that by using the GANN model to make pressure recognition based on thermal-RGB compressed video stress data set, the prediction results are accurate to a certain degree.

From the table, we can find that as the generation increases, whether it is a pruned model or a non-pruned model, the accuracy is increasing. It shows that if we want to obtain better prediction results, we could increase the generation. However, when we add to 10,000 generations in the pruned GANN, it appeared a situation where fitness no longer increases, just like figure 4.

Continuing to increase the generation, it seems that fitness has a maximum. We think the reason for this is that mutation and crossover are not enough to generate the parameters for the network to converge. This is the limitation of genetic algorithms.

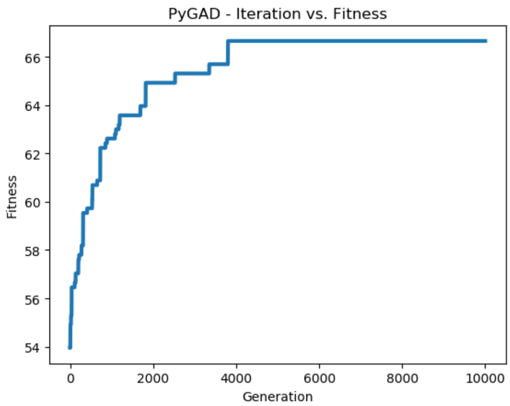

Fig. 4. Pruned GANN With 10000 Generation

3.2 Experiment on Other Parameters of GANN

The genetic algorithm selects the feature subset with the highest fitness by multiplying and eliminating feature subsets to achieve optimizing the weight of the last layer of the network.

In this experiment, I found two main parameters that will affect the output. One is the number of generations. The more generation it has, the more subsets will be generated, and there will be more high-quality feature subsets. Due to GA algorithm training takes a long time, we use 1000 generations when exploring other characteristics of genetic algorithms. The second is the percentage of mutated genes, so we chose 5, 10, 15, 30 of mutated genes percent for comparison.

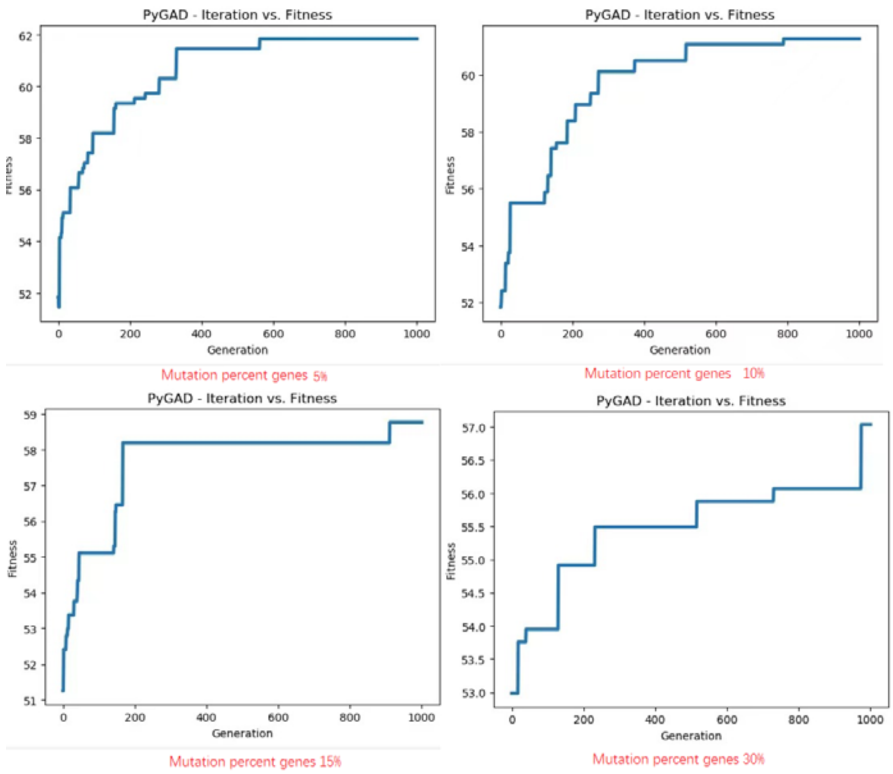

Fig. 5. Different Gene Mutation Percentage

We use a pruned network to experiment, which saves us a lot of time. From figure 5, we could find that the smaller the percentage of mutant genes, the better the predicted results. The different percentage of mutant genes does not seem to have a significant effect on training time.

3.3 Comparison of GANN Network and NN Network

In the experiment, we first established a two-layer neural network (NN), and then after pruning, the output of the last layer of the neural network was turned into the input of GANN, and finally we through GA to fine tune the last layer of the network. We compared the accuracy of the unpruned NN network without GA and the pruned GANN network and pruned NN network without GA. The result is in the below table 3.

Table 3. Comparison of unpruned NN network without GA, pruned GANN network, pruned NN network without GA

NN Epoch/GANN Generation	Unpruned NN Train Accuracy	Unpruned NN Test Accuracy	pruned NN Train Accuracy	pruned NN Test Accuracy	GANN Train Accuracy	GANN Test Accuracy
100	67.82%	55.00%	91.32%	54.00%	55.29%	51.00%
500	85.16%	54.00%	93.44%	56.00%	58.76%	52.00%
1000	88.24%	57.00%	95.37%	51.00%	59.15%	55.00%
1500	92.29%	56.00%	96.12%	49.00%	60.27%	54.00%
2000	89.59%	55.00%	96.97%	52.00%	60.27%	52.00%

From this table we could find that the network after pruning and fine tune performs best, and the GANN performs worst. Moreover, with the increase of generation, the accuracy of GANN has not quickly improved. Compared to GANN, pruned NN network and unpruned NN network have improved very quickly. This shows that compared with GANN, traditional NN may perform better on classification problems.

4 Conclusion and Future Work

Comparing the accuracy of the pruned and unpruned network, we can draw the conclusion that in the GANN network, the pruned network will have better efficiency, and pruning the units with very small weights will not affect the accuracy. As the GA algorithm in this model cannot use GPU for calculations, it takes very long time to train. We also found in experiments that more generations mean better fitness, but the fitness cannot be increased without limitation, and a maximum fitness will appear. What’s more, due to the RGB-thermal compressed video stress dataset, the test accuracy is around 50%, which makes the models we apply are over-fitting. In this paper, we reduced overfitting by conducting fine tune on the pruned model. Although we removed the last layer of the NN network and add GA to update it, the performance of GANN is not as good as the traditional neural network, so maybe gradient is the better way to update the network.

In future work, we could try to use the deep learning method for training, which may have higher accuracy or shorter training time. Regarding to the effect of genetic mutation percentage, we only tested a few commonly used data in this experiment. In the future, if we test some extreme data, like 1% or 99%, there may have some interesting results. If we get a more suitable data set in the future, we think the GANN model will have better performance.

References

1. Irani, R., Nasrollahi, K., Dhall, A., Moeslund, T. B., & Gedeon, T. (2016). Thermal super-pixels for bimodal stress recognition, 2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA) (pp. 1-6). IEEE,2016.
2. Gedeon, TD, Harris, D. (1991). Network Reduction Techniques Proc. Int. Conf. on Neural Networks Methodologies and Applications, AMSE, San Diego, vol. 2, pp. 25-34, 1991
3. Vitetta, L., ANTON, B., CORTIZO, F., Sali, A.: Mind-body medicine: Stress and its impact on overall health and longevity. Annals of the New York Academy of Sciences 1057, 492 { 505 (12 2005). https://doi.org/10.1111/j.1749-6632.2005.tb06153.x
4. Sietsma,J, Dow, RF, "Neural net pruning - why and how," IJCNN, vol. I, pp. 325-333,1988.
5. Sharma, N., Dhall, A., Gedeon, T., Goecke, R.: Modeling stress using thermal facial patterns: A spatio-temporal approach. pp. 387{392 (09 2013). https://doi.org/10.1109/ACII.2013.70
6. Lee, C., Yoo, S. K., Park, Y., Kim, N., Jeong, K., & Lee, B. (2006, January). Using neural network to recognize human emotions from heart rate variability and skin resistance. In 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference (pp. 5523-5525). IEEE.
7. Thayer, J. F., Åhs, F., Fredrikson, M., Sollers III, J. J., & Wager, T. D. (2012). A meta-analysis of heart rate variability and neuroimaging studies: implications for heart rate variability as a marker of stress and health. Neuroscience & Biobehavioral Reviews, 36(2), 747-756.
8. Mirjalili, S. (2019). Genetic algorithm. In Evolutionary algorithms and neural networks (pp. 43-55). Springer, Cham.
9. jarmouni, f. (2020). Train Neural Networks Using a Genetic Algorithm in Python with PyGAD. Medium. Retrieved 29 May 2021, from https://heartbeat.fritz.ai/train-neural-networks-using-a-genetic-algorithm-in-python-with-pygad-862905048429.
10. Wang, S. C. (2003). Artificial neural network. In Interdisciplinary computing in java programming (pp. 81-100). Springer, Boston, MA.
11. Jyothi, V., Kumar, B. R., Rao, P. K., & Reddy, D. R. K. (2011). Image fusion using evolutionary algorithm (GA). Int. J. Comp. Tech. Appl., 2(2), 322-326.
12. Sano, A., & Picard, R. W. (2013, September). Stress recognition using wearable sensors and mobile phones. In 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction (pp. 671-676). IEEE.