Effect of classifying network physiology of 'fight or flight' response in facial superficial blood vessels using neural network with deep learning and the influence of pruning

Naisheng Liang[1]

[1] Research School of Computer Science, Australian National University

Email: u6356745@anu.edu.au

Keyword: Granger causality, neural network performance, LSTM model, pruning network, weight distinctiveness

Abstract: *Objective*: We construct a two layers neural network with a deep learning approach to identify the dynamic pattern of blood flow changes in the cutaneous superficial blood vessels of the face for 'fight or flight' responses through facial thermal imaging and improve the performance of the neural network by pruning the hidden neurons. *Approach*: To achieve these goals, the original dataset was collected from 41 subjects in a mock crime scenario. The device collected the thermal information form five areas on the face which are periorbital, forehead, paranasal, cheek and chin. Then extracting the causality features by using effective connectivity approach and the graph analysis. Finally we get the features which were modified version of the multivariate Granger causality (GC) method among each pair of facial regions of interests. Thus we can set up a two layers neural network and use the dataset to feed it then see how it performs. *Main results*: Validation was performed to adjust the hyperparameters, and the results demonstrate that the neural network can give a better performance than most methods but the weight vectors are all similar as the angles between them do not converge to 90 degrees. The average result is greater than 60% accuracy rate in discriminating between deceptive and truthful subjects.

1 Introduction

1.1 Assumptions

In this paper we will generally assume a normal neural network which is a feed-forward network of two layers of neurons. Each neuron connects all the neurons in the next layer and thus every hidden neuron has two weight vectors which storages the weights between input neurons or output neurons and the hidden neurons. The network and deep learning model are implemented by using Pytorch and trained by the thermal dataset (Derakhshan and Mikaeili and Motie and Gedeon, 2018) . Moreover, the network also uses back-propagation to measure the error.

1.2 Description of dataset

The dataset we used in this paper to train the simple feedforward neural network(FNN) is constituted by 20 features and 1 label which represent the grange causality of each pair of five different facial areas and target value 1 or 0 namely the participant is in the truthful or deceptive group. To get GC of five facial areas, we need to collect some information of these facial areas.

In the original experiment, the participants will experience a mock crime scenario: they are divided into two groups, people in deceptive group would tell a lie and people in truthful will tell a truth as usual. Then using the device which is a thermal camera to record the change of temperature of face of each participant. Therefore the initial dataset is much larger than we used and now we can calculate the GC of facial areas but here the original paper improve the GC to extend GC which can be called eGC.

To derivate this, we need to figure out the concept of granger causality which is method to estimate causality between two variables in a time series and this was first introduced by Granger (1969) in terms of vector auto regressive (VAR) modelling of multivariate processes. And there is a problem that the conflict of time-lagged and zero-lagged effects in the VAR model and both time lagged and instant effects can be utilized as the the extended version of GC based on a extended version of VAR model and this can figure out the zero-lag effects in linear regressions (Porta and Faes 2016).

Now we can create the present dataset based on the initial dataset by using the formula of calculating eGC and the now we can get 20 features about 20 pairs of different facial areas. After preprocessing the dataset a little, we can investigate the relationship between the reaction of human and the thermal change on their face.

2 Method

2.1 Dataset information and data pre-processing

As we also implement a deep learning approach to investigate this problem which is Long Short-Term Memory(LSTM) model, we use another updated dataset that is more suitable for the LSTM. The reason why we adopt this updated dataset is that it has 31 sheets which storage the maximum and minimum value of temperatures of five different face regions in the chronological order which is 10Hz for up to 20 seconds for each participant and a sheet is the labels of them indicating that the participant tells the truth or deceives. Based on these data in times sequence, we can use LSTM model to determine the condition of each participant by using prediction in the final time stamp.

Given the objective of this network is to predict whether a participant tells the truth or deception using the thermal changes of different regions on their face, the input features are thermal related data: for training two layers FNN, the input features are the GC of each pair of different face areas and for training LSTM model, the input features are the timeseries temperature changes of participants' faces when they are required to answer prepared questions. The target output of both models is a binary class: '1' indicates that the answer of this participant is truthful; '0' indicates that the answer is deceitful.

The data in the FNN's training dataset is all in the range of (0,1) and also neat enough so we do not preprocess it further but here we can select four features from 20 features as input since if we want to make a relationship among all five face areas then at least we only need the GCs of four pairs of different face region and other GCs are redundant. Hence, we can select the four most important features according to the results of experiments from the original paper.

Based on our observation of the data in the LSTM's training dataset, the data are all around 33 and some data has 199 timestamps , however, other data has 179 or only 149 timestamps. Additionally, the dataset records the thermal changes when the participants answer 8 questions but people in the deceptive group are asked to tell a lie for the question 6 which means that the temperature of these people's face will change only when they answer the question 6 and the data also shows this phenomenon obviously. Therefore, we apply the standard z-score normalisation to the data to increase the accuracy and efficiency of the classification model.(Han, J., Kamber, M., & Pei, J. (2011)) Then we use 0 which is the mean value of each column of data to pad out the column whose length is smaller than 199 and we also consider another method at the beginning which clips the length of all columns to 149 since the change of temperature is basically between 70 and 115 timestamp but this may case some information loss and thus we prefer to use the previous method. Eventually, we only focus on the thermal data of question 6 and the dimension of final input data is 199x31x10 which means 199 time steps, number of batch and 10 features.

2.2 Feedforward neural network model design and performance measurements

In this experiment, we construct a two-layer fully connected neural network. One input layer has 20 neurons which corresponds to the 20 GCs of each pair of face regions. One hidden layer has set number of units. One output layer has two output neurons which corresponds to the binary classification labels.

Since the dataset we using is not very large, we apply the k-fold cross validation to it. The dataset is split by participants into 10 folds to construct 10-fold cross validation. Also this dataset is reasonably class balanced (52% of participants are labelled as '1' and the rest 48% of participants are labelled as '0'), therefore we use the mean accuracy of each fold to evaluate how accurate the model is in predicting the class label for a new data pattern(Montana, D. J., & Davis, L. (1989)). Then we run 10 times for each fold to eliminate the effect of randomness of this model. In each run, we calculate the mean accuracy of 10 runs of each fold and the test accuracy of this FNN model is the mean accuracy of 10 folds.

2.3 Neuron pruning technique design and implementation

To describe the technique of pruning we implemented in this paper, firstly there are some basic concepts need to be figured out. Hidden neurons have some properties such as *relevance* (Mozer and Smolenski, 1989, Segee and Carter, 1991), *sensitivity* (Karnin,1990), *badness* (Hagiwara, 1990), and *distinctiveness* (Gedeon and Harris, 1991a) and distinctiveness is the most useful properties to achieve the goal of pruning network. Here is the brief description of distinctiveness: the activation vectors of the hidden neurons output corresponding to the pattern presentation set determines the distinctiveness of this hidden neuron. In practice, we can construct a vector which has the same dimension as the number of patterns in the training dataset for every hidden neuron and the each component in this vector corresponds to the output activation value of the specific neuron. Therefore the vector can show the functionality of particular hidden neuron in the input space.

In this FNN model, we consider that vectors for identical or clone neurons will be the same even though the relative size of their final outputs will be recorded. Based on the previous experience (Gedeon), the separations of angles up to 15° or larger than 165° are considered as a similar or clone neuron of other neurons and one of these neurons should be removed for the general pruning network. And the weight vector of this neuron which has been removed would be added to the weight vector of the remaining neuron. Since the average effect of these hidden neurons will not change significantly in the situation of low angular separations and there still remains enough mappings from the weights to pattern space whose error measure will also not be worse subsequently.

To implement the distinctiveness angle measure between two weight vectors of hidden layer, given two vectors v_a and v_b with the same length, then we calculate the angle between them which is(Steinbach, Michael, George Karypis, and Vipin Kumar. (2000):

$$\varphi(v_a, v_b) = cos^{-1}\frac{v_a * v_b}{||v_a|| * ||v_b||} \quad (1)$$

If the angle $\varphi(v_a, v_b)$ is less than 15° or larger than 165°, then v_a can be replaced with v_b. Hence we do not need to further train after removing the replaceable hidden neurons and the compression rate is defined as the ratio between the number of original neurons and the number of unreplaceable neurons.

2.4 LSTM model design and implementation

Based on our observation of the updated dataset in section 1.2, the LSTM model which is one of Recurrent Neural Network(RNN) models is chosen to perform the binary classification on this dataset as this model is good at handle time series data. To implement this LSTM model, there are not any configurations need to be designed particularly or hyperparameters need to be adjusted as it is similar to the FNN model but it has a layer of cells which can remember the previous inputs and make a prediction based on all inputs from the first timestamp until the present timestamp.

This dataset also only has 31 samples which is same as dataset we use in FNN model, hence we apply same method to calculate the test accuracy of this model which is 10-fold cross validation. The main steps of implementation are same as we did in section 2.2 and we use the output of hidden layer in the final time stamp to make a prediction which is reasonable as this output consider all the condition of all time stamps.

To explore the relationship between LSTM model and the effectiveness of the neuron pruning technique, we compared the mean of test accuracy in this Section with the results in Section 2.3 and Section 2.4.

2.5 Local optimal number of hidden neurons experiment

In this experiment, the aim of us is to find the local optimal number of hidden neurons which can maximize the genral performance by producing the maximum test accuracy.

In Section 2.2, we construct a FNN to perform a binary classification model. Based on that, we use the sigmoid function as the activation function and cross entropy error as the loss function since the label is not encoded by one-hot. For the optimizer, we choose Stachastic Gradient Descent algorithm as this algorithm can give a better performance of convergence and a reasonable loss curve as well as the model is simple namely easy to train and the dataset is also small enough, hence the time of convergence will not be very long.

In Section 2.4, we implement a LSTM model. The difference from this model and model above is that we select the Adam optimization algorithm as the optimizer. This LSTM is much more complicated than FNN and it will need more time to find the solution so we need a effective and computationally efficient algorithm. Adam is a suitable and the performance of it is also stable

For both models, we calculate the mean train and test accuracy for each testified number of hidden neurons from 2 to 20 and we also plot the average test accuracies corresponding to the number of hidden neurons from which we can obtain the theoretical local optimal number of hidden neurons. In table as below, we can see the structure of each experiment clearly.

Table 1: specification of models

	Inputs	Hidden	Outputs	Loss function	Optimizer
FNN	31x20/31x4	9	2	Cross-Entropy	SGD
FNN+pruning	31x20	Changing	2	Cross-Entropy	SGD
LSTM	31x199x10	7	2	Cross-Entropy	Adam

3 Result and Discussion

3.1 Prediction performance of feedforward neural network

As shown in Figure 1, there are two lines in the plot: the blue one represents the test accuracy of FNN trained by all GCs and the red one represents the FNN trained by 4 more important GCs based on the result of previous experiment. The majority of number settings resulted in above 65% of test accuracy and the highest test accuracy of blue line peaks at 78% and the red line peaks at 82% when there are 9 neurons in the hidden layer. Therefore, the local optimal number of hidden neurons should be 9 in FNN.

In a ward, the FNN with 9 hidden neurons has around 82% accurate at most in predicting the participant is truthful or deceptive when given the four specific GC which are Forehead→ periorbital, Chin → perinasal, periorbital → Forehead, Chin → Forehead.

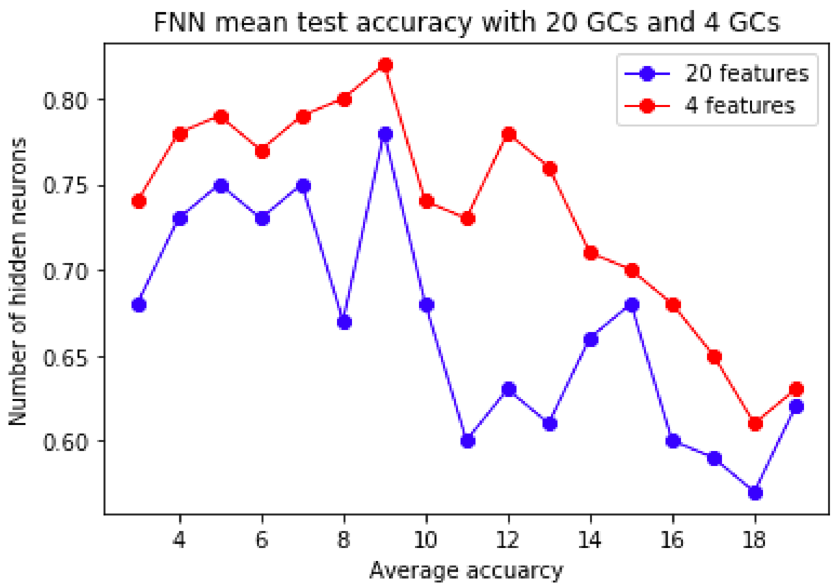

Figure.1. Two dotted lines in different colours are the mean accuracies of FNN with different inputs and number of hidden neurons

3.2 Analysis of neuron pruning technique

Now we can apply the technique to our neural network and investigate how it works for impacting the performance of the neural network.

The dataset has 20 features and 2 target labels which are 1 or 0 thus we should build up a neural network that has 20 input neurons and 2 output neurons corresponding to the dataset. Then splitting the dataset to train dataset and test dataset in order to feed the neural network with the training patterns and we can obtain the loss of training samples and the accuracy of this model based on the testing data.

In this experiment, the initial value of hidden neurons is 19 and the minimum number is 3. Before using the distinctiveness angular measures, we firstly investigate the relationship the number of hidden neurons and the minimal angles between hidden neurons.

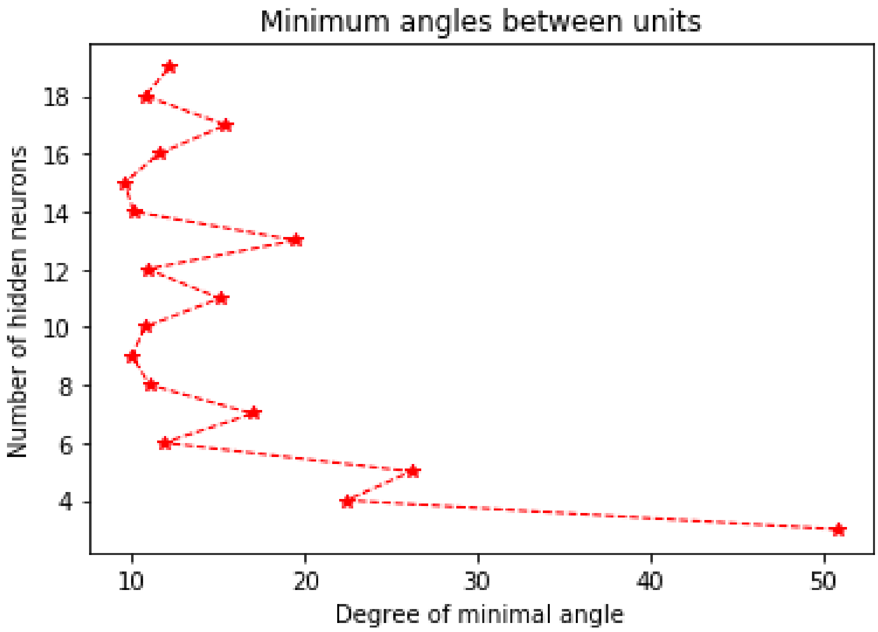

Figure.2. The minimal angle of all pairs of hidden neurons in different number of hidden neurons

As shown in the Figure 2, we can see that when the number of hidden neurons is larger than 5, the values are lower than the standard 15° and values are much higher than this standard if the number is smaller than 5. Therefore we can apply the distinctiveness angular measures to these hidden neurons in the neural network. We gradually shrink the size of the hidden layer.

Another interesting point need to be mentioned is that if we remove the several first neurons which are around 60° then the accuracy will not have a significant reduction in the error measure. This mainly because these neurons are the same as secondary backup neurons for increasing network damage resistance (Gedeon and Harris, 1991b). After we removing these neurons including some neurons as described before, the network still does not require retraining (here we trained for 20000 epochs).

To compare the performance of this FNN model before and after the pruning, we construct Figure 3 which shows that the relationship between the number of hidden neurons and the average accuracy which can be considered as an indicator. The yellow line shows the accuracy of FNN before we applying the pruning technique and the purple line shows that the accuracy after we removing all replaceable hidden neurons.

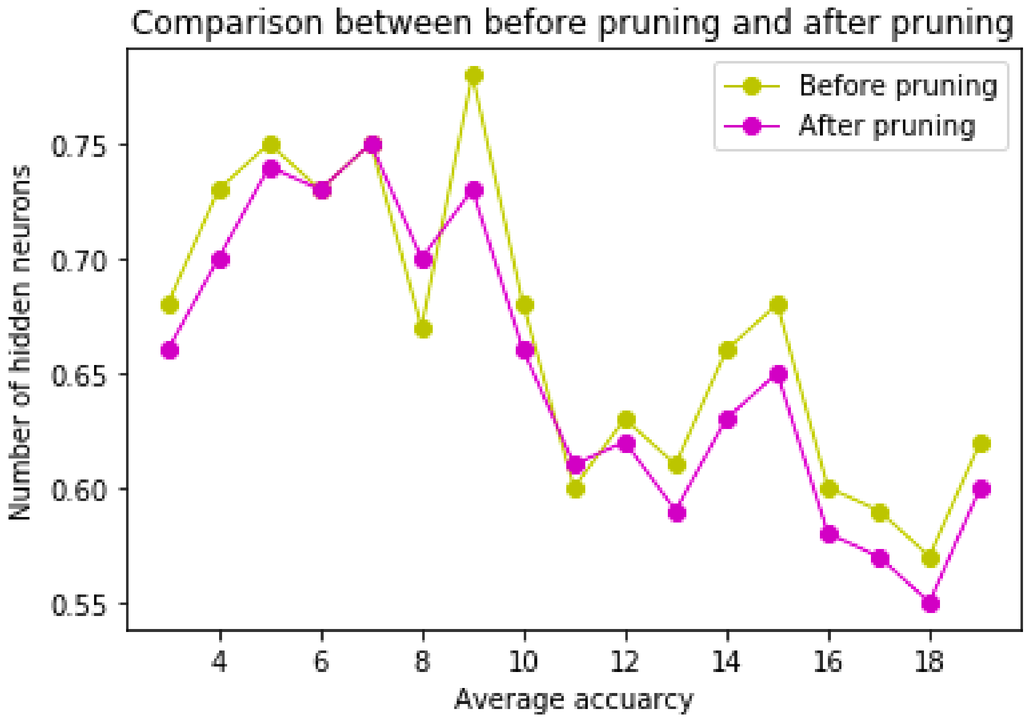

Figure.3. The mean accuracy of FNN with different number of hidden neurons before and after pruning

Based on the result from this figure, the performance decreases slightly at most set number of hidden neurons as some neurons are removed. Generally, the performance gains a reduction with more neurons are removed. Hence, this result is not what we expected as this basically reduces the performance of the neural network we have trained.

3.3 Prediction performance of LSTM model

The approach of investigating how accurate the LSTM model with time series data can achieve is similar to the method we applied in Section 3.1. Based on the result of experiment, we construct the Figure 4 as below. As shown in this figure, the average accuracy of this model is around 60% and the line peaks at 66% and there are 7 hidden neurons in the network. It is clearly that using the thermal change of participants' faces in chronological order when they are answering the question may not be good inputs to predict whether this participant tells a lie or not as the performance of LSTM is worse than FNN we developed with 20 GCs.

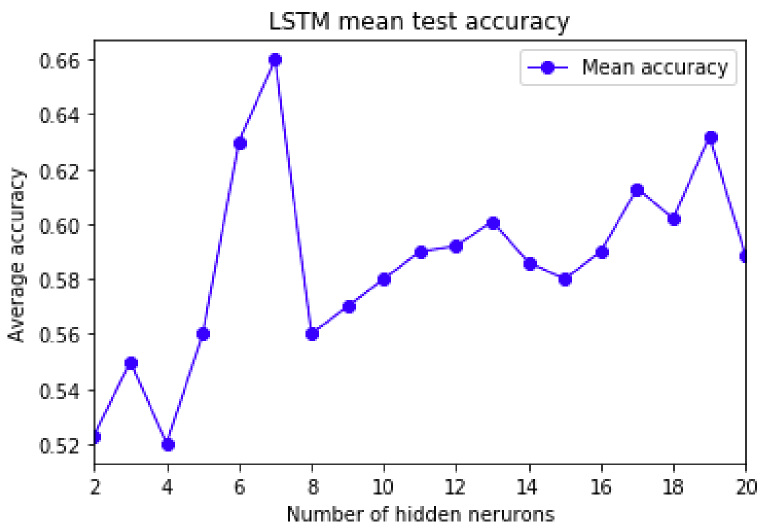

Figure.4. The mean accuracy of LSTM with different number of hidden neurons

After evaluating all methods of predicting this binary classification problem, we construct a table to compare all methods together with their average accuracy which the table 2 as below:

Table 2: Comparison of all methods

Method	Inputs	Hidden	Accuarcy	Time cost
FNN	31x4	9	82	45.33s
FNN+pruning	31x20	Changing	73	42.19s
LSTM	31x199x10	7	66	24.57s

4 Conclusion and Future work

In our work, we develop a FNN prediction model which has at most 82% accuracy then we also apply the neuron pruning technique on the FNN model and implement LSTM with different dataset to construct another prediction model.

The two-layer neural network with 9 hidden neurons can be considered to be acceptable in predicting a human is truthful or deceptive when they are speaking given the particular GC of their five face regions. To further improve the prediction accuracy of trained LSTM model, we should consider some other physiological features which are intuitively related to human's reflection of their faces when they are telling a deception such as the frequency of eye movements.

The distinctiveness angle measure technique can remove some unnecessary hidden neurons without reduce too much of the model generalization ability for our prediction task which is indeed useful. Additionally, if we prune the neurons by thresholding distinctiveness angle, then it may break the global feature constructed with multiple neurons at the same time. Thus we may cluster the hidden neurons and investigate the functionality of these clusters then we can perform the pruning technique to the cluster.

References

Derakhshan, Amin & Mikaeili, Mohammad & Motie Nasrabadi, Ali & Gedeon, Tom. (2018). Network physiology of "fight or flight" response in facial superficial blood vessels. Physiological Measurement. 10.1088/1361-6579/aaf089.

Gedeon, TD and Harris, D, "Creating Robust Networks," IJCNN, pp. 2553-2557, Singapore, 1991b.

Gedeon, TD and Harris, D "Hidden Units in a Plateau," Proceedings 1st International Conference on Intelligent Systems, pp. 391-395, World Scientific, Singapore, 1992b.

Granger C 1969 Investigating causal relations by econometric models and cross-spectral methods Econometrica 424–38

Hagiwara, M, "Novel back propagation algorithm for reduction of hidden units and acceleration of convergence using artificial selection," IJCNN, vol. 1, pp. 625-630, 1990.

Han, J., Kamber, M., & Pei, J. (2011). In Data Mining: Concepts and Techniques (ch.8 Classification: Basic Concpets

Han, J., Kamber, M., & Pei, J. (2011)). The Morgan Kaufmann Series in Data Management Systems Morgan Kaufmann Publishers.

Karnin, ED, "A simple procedure for pruning back-propagation trained neural networks," IEEE Transactions on Neural Networks, vol. 1, pp. 239-242, 1990.

Montana, D. J., & Davis, L. (1989, August). Training Feedforward Neural Networks Using Genetic Algorithms. In IJCAI (Vol. 89, pp. 762-767).

Mozer, MC, Smolenski, P, "Using relevance to reduce network size automatically,", Connection Science, vol. 1, pp. 3-16, 1989.

Porta A and Faes L 2016 Wiener–Granger causality in network physiology with applications to cardiovascular control and neuroscience Proc. IEEE 104.2 282–309

Segee, BE, & Carter, MJ, "Fault Tolerance of Pruned Multilayer Networks," IJCNN, vol.2, pp. 447-452, Seattle, 1991.

Steinbach, Michael, George Karypis, and Vipin Kumar. "A comparison of document clustering techniques." KDD workshop on text mining. Vol. 400. No. 1. (2000

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