Distinctiveness Pruning on 'fight or flight' Response Prediction LSTM Network

Guanjie Huang

Research School of Computer Science, Australian National University
Canberra, Australia
Guanjie.Huang@anu.edu.au

Abstract. Hidden units' distinctiveness analysis is an effective Neural Network (NN) pruning technique. This algorithm using the vector angle to determine if a unit is distinct enough in the hidden layer and decide to prune it or not. In this paper, we try to apply distinctiveness pruning on a fight or flight' response prediction NN with Long Short-Term Memory (LSTM) hidden layer and evaluate the pruning performance. Also, we try to determine pruning and fine-tuning effect on test accuracy of the trained model.

Keywords: Neural Network, LSTM, Distinctiveness pruning, Fine-tuning, Response predict

1 Introduction

"Fight or flight" reaction is the primary brain decision when facing danger situation [1]. Using technology to predict this decision helps in analysing the psychological activities of suspects during interrogation or predicting the behaviour of criminals to protect police officers. Research shows this decision connects to facial blood flow changes and can be determined by collecting facial temperature. By analysing the thermal images, researchers focusing on periorbital, forehead, perinasal, cheek and chin, five face areas' temperature changes. They built a time sequence dataset with sample frequency equal to 10 Hz and up to 20 seconds. To make the data clearer and easier to learn, one solution is to build effective connection models between every two areas [2] and use a modified version of the multivariate Granger Causality (GC) method to quantify the connection. By selecting most discriminative features and applied processed dataset to traditional machine learning (ML) algorithms, the team get over 87% test accuracy on predicting the fight or flight reaction [2]. Also, in previous work, using selected features on basic NN can get over 72% test accuracy [3].

Feed-forward networks have been used for solving lots of problems nowadays. However, training a NN with back-propagation needs a high cost in time and computation resources. Some old ideas believing include more hidden units than required helps training networks within reasonable time scales [4]. However, with this method, many hidden neurons perform no real function. In other words, they are duplicate with the functional units and should be removed to save the cost both in memory and computation time. So, determine suitable hyperparameter is important but difficult to achieve. T.D. Gedeon's research [5] summarises and compares various network reduction technologies. Distinctiveness analysis is one of the methods and can apply automatically when training a NN. It calculates the angle between different hidden units' output activation vector, and determine neurons need to be prune.

Fine-tuning (FN) is a common method wild used in Transfer Learning (TL). When facing a brand-new machine learning task, using a pre-trained model with the FN can make the model quickly converged to save training costs and improve accuracy [6]. FN helps in keeping pre-trained models' feature extract ability and re-training the prediction layer. So, it also can be used in re-training pruned models.

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture [7]. It was proposed by Sepp Hochreiter and Jürgen Schmidhuber in 1997. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. Due to it can memorize values of indefinite length of time, it is suitable for processing and predicting important events with very long intervals and delays in time series such as handwriting recognition, speech recognition, language translate and time series problems.

To achieve the same goal of predicting the "Fight or flight" reaction, and directly apply time sequence dataset, we use a simple LSTM network as another approach and apply the distinctiveness pruning and FN on it. We try to compare prediction performance between these two methods and evaluate the effectiveness of distinctiveness pruning method. We set four different models in the experiment. The models differ in three factors, using applied the pruning method or not, two different vector angle calculation algorithms and with or without FN.

2 Method

Our main objective is to determine the LSTM network's performance on prediction "fight or flight" reaction through the thermal dataset and evaluate the distinctiveness pruning technique in different situations. Figure 1 shows an overview of the experiment structure through the whole process and present in this paper. Where the Cosine Distinctiveness Pruning (CDP) method and ArcTan Distinctiveness Pruning (ATDP) method will introduce later in this chapter. Many methods are chosen to keep similar experiment environment with GC approach tests [3].

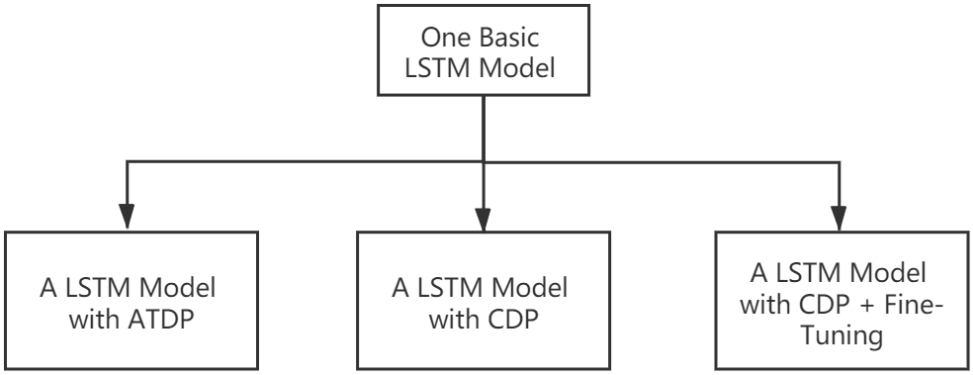

Fig. 1. Experimental structure with Four models.

2.1 Network Structure and Hyperparameters

2.1.1 Network Structure

To achieve using time series data directly, we took a 'many to one' LSTM network as the basement, also added a sigmoid activation function before output. LSTM network has many forms, but in this experiment environment, we only care about the final output of the time sequence. Therefore, we set this 'many to one' structure and only took the output of the last LSTM layer. Also, consider the pruning situation, scale down the output is required. The basement network structure is shown in Figure 2.

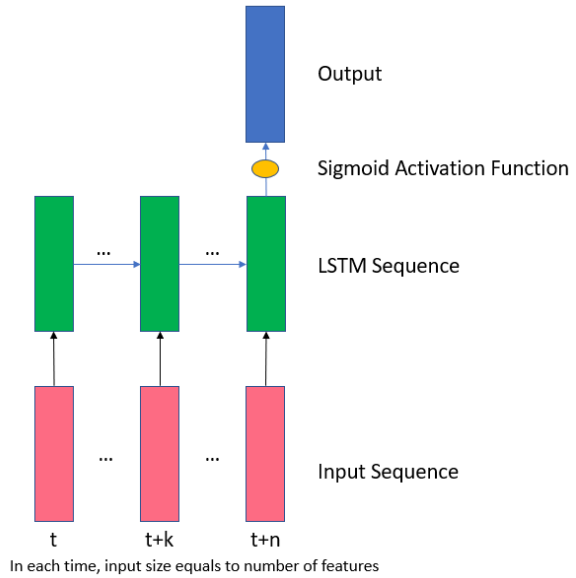

Fig. 2. Basic experiment model structure

2.1.2 Hyperparameters

Input Size

According to Derakhshan's research, the time series thermal dataset contains the maximum and minimum temperature of five facial parts. Therefore, for each person and in each time, there are ten features that can be used as input.

Output Size

The network aims to discriminate deceptive and truthful subjects in 'fight or flight' response, it can be represented by zero and one. Therefore, just setting one output neuron and keep consistent experiment environment with basic NN tests.

Hidden Layer Size and Training Epochs

Determine suitable hidden layer size and training epochs is difficult but necessary. Try to make the model coverage and prevent from overfitting, we used a grid search to determine suitable parameters. In each test, the result was validated through k-fold cross-validation. The results are shown in Table 1.

Table 1. Prediction accuracy among different model types.

Layer size	Epochs=50	epochs=100	epochs=150	epochs=200	epochs=250
5	36.67%	43.33%	46.67%	43.33%	43.33%
10	56.67%	46.67%	53.33%	46.67%	50.00%
15	33.33%	56.67%	50.00%	50.00%	53.33%
20	46.67%	66.67%	50.00%	50.00%	50.00%
25	50.00%	46.67%	33.33%	40.00%	53.33%
30	53.33%	50.00%	53.33%	53.33%	46.67%

It is clear to find, with layer size equals to 20 and training epochs equals 100, the basic LSTM network can get the best performance.

Loss Function

To keep consistence with basic NN tests, we choose Mean Square Error (MSE) as the loss function, its formula is shown in equation (1).

$$MSE = \frac{\sum_{i=1}^{n}(y_i - y_i^p)^2}{n}. \quad (1)$$

The MSE loss function can well reflect the average distance between the predicted value and the true value.

Other Hyperparameters

We also choose Adam optimizer [8] for faster gradient descent to reduce the number of training cycles. This method based on adaptive estimates of lower-order moments and computationally efficient. Considering take Adam optimizer, we try to set the learning rate to a small number 0.01. This avoids models have high sensitivity to the last input training data and avoid overfitting.

2.2 Distinctiveness Pruning and Fine-tuning

2.2.1 Distinctiveness Pruning

Distinctiveness pruning technique has the following steps [5]:

1. After several epochs of training, get the hidden layer's activation output
2. Normalize the output by minus 0.5
3. Build vectors for each hidden unit with the dimension as the number of training set length
4. Calculate the vector angle between each two vectors
5. If two vectors' angle smaller is than 15 degrees, they are considered sufficiently similar, and one of them should be removed. The removed neuron's weight should be added into the kept one
6. If two vectors' angle is greater than 165 degrees, these two neurons are effectively complementary. They can be removed simultaneously

Some notes need to mention for several steps. In step 2, Considering all the outputs of the sigmoid activation function are from 0 to 1, by minus 0.5, the vector angle becomes from 0 to 180 degrees. In step 4, there are two different angle calculation algorithms. They would be discussed later in the following part. In step 5, added all the weights connect with the prune neuron to the corresponding weights of kept one.

Besides the main steps in the distinctiveness reduction method, choosing the pruning time is also essential. From Gedeon's research [5], networks using this pruning method does not require more training and can use it immediately. From this, the pruning process executes after a model training enough. This algorithm removes complementary pairs of vectors and which can be seen as a zero-output pair, also for the similar pair, this method adds the pruned units' weight to the kept one. The network will not experience more training. The added weight maintains the distinct units' function

in the whole network. However, from previous experiments on basic NN [3], the distinctive pruning brings around 10% accuracy decrease due to the unbalance pruning pairs. So, set FN control group is necessary to find out the reason for this phenomenon.

2.2.2 Two Different Angle Calculation Methods

There are two different vector angle calculation algorithms. The first one is the pure a mathematical calculation method. The formula is in equation (2)

$$angle(i,j) = cos^{-1}\left(\frac{i \cdot j}{|i| \cdot |j|}\right). \quad (2)$$

In subsequent sections, we call the pruning method using this angle calculation as Cosine Distinctiveness Prune (CDP)

The second one comes from Gedeon's work [9], his team use another different calculation method to calculate the multi-dimensional vector angles. The formula is in equation (3)

$$angle(i,j) = tan^{-1}(\sqrt{\frac{i^2 * j^2}{(i*j)^2}} - 1). \quad (3)$$

Where i, j are the activation vectors minus 0.5, consider we already normalize the vector in this way, we can directly use our vectors in this algorithm.

In subsequent sections, we call the pruning method using this angle calculation as ArcTan Distinctiveness Prune (ATDP). With these two methods, we build two different version of prune method.

2.2.3 Fine-tuning Method

The introduction of FN is to eliminate the impact of pruning imbalance on test accuracy. Imagine that the small-scale re-training of the pruned network can make the prediction of the network more accurate, which is also consistent with the idea of FN in TL. Therefore, in the experiment, when the total number of training is fixed, doing early pruning, and the remaining training times are used for FN. We also used grid search for determining best FN epochs.

2.3 Data Process Method

2.3.1 Feature Select and Batch Input

The data set contains the face temperature data of 31 subjects who answered eight different questions. The sixth question is directly related to the 'fight or flight' response and is given a deception or fact label. Other questions are irrelevant questions and default to truthful answers. Since the tags of related questions are sufficiently balanced, and to avoid the adverse effects caused by the imbalance of the data set, only the data of the sixth question is used for training and testing in the experiment.

Besides, the construction of the input matrix is also critical. The LSTM layer allows all the time series data to be input at once, so an input matrix of 31 * 149 * 10 was constructed in the experiment, where 31 corresponds to the number of candidates, 149 corresponds to the time series, and 10 corresponds to the input features.

2.3.2 Input Normalization

Due to the dataset is facial temperature captured by a thermal camera, the data are all around 36 degrees. To highlight data features and facilitate the model learning, we stitched together the data of all candidates according to facial parts and applied z-score normalization. The z-score follows the algorithm shown in equation (4).

$$Z = \frac{x-\mu}{\sigma}. \quad (4)$$

Where x is the original data, μ is the mean of data, and σ is the standard deviation.

2.4 Result Evaluation Method

The training of NN is full of randomness, which means the results of each experiment may be different. For a network trained with a small data set under completely random conditions, the obtained test results will vary in size. To make the experimental more precise, we used a controlled variable method, using random seeds to ensure that the randomly shuffled data set is the same in different models. The k-fold cross-validation method is also adopted to ensure the reliability of the results.

K-fold is a simple cross-validation method. The idea is split a randomly shuffled dataset into k groups. Then for each group, a model takes a loop that treats this group as a test set and left n-1 groups as a training set. When all group have

been seen as a test set, the validation process ends. In this way, it is easy to evaluate the model's general performance by using the average performance of each training loop.

In our test process and present in this paper, we divided the whole dataset into five groups with the considerations of the length of the dataset. In this time, the dataset only contains 31 people's data. To avoid using too small test set and keep the accuracy data reliable, we keep the data set as around six labels per group.

3 Results and Discussion

With all method describe above, we build four types of models, with the difference in if applied pruning, which pruning algorithm was used and if applied FN. We evaluate the test result in two dimensions. First, focus on approach performance, we try to compare the test accuracy between GC method and LSTM method. The GC method results are provided by Derakhshan's research [2] and basic NN tests [3].

Second, we focus on how distinctive pruning affect the accuracy and discuss what factor would influence the prune results. Besides, try to verify the theory that FN would eliminate the adverse effects of pruning.

3.1 LSTM Network Performance Evaluation

For the test results comparison, GC method's traditional machine learning result and basic NN results are shown as GC ML and GC NN. Both results are whole dataset input results (not the selected feature test results) [2]. The LSTM method's results are shown as LSTM Base, LSTM CDP, and LSTM ATDP. The whole test accuracy results are in Table 2.

Table 2. Prediction accuracy among different model types.

Predict method	Test accuracy (%)
GC ML	58.9
GC NN	65.7
LSTM Base	66.7
LSTM CDP	60
LSTM ATDP	66.7

From table 1, it is easy to find the LSTM method performs slightly better than the GC approach, and NN approaches are better than traditional ML approaches.

It is easy to explain this, the NN can be regarded as a feature extractor to a certain extent. According to NNs learning mechanism, in several training cycles, the NN adjusts the connection weight and bias of the units through backpropagation. One of the results is that useful feature effects are amplified, which is why the NN performs well in the environment of complex features.

3.2 Pruning and Fine-tuning Effects

3.2.1 Pruning Effects

Through the experiment of prediction accuracy, we also calculate the average pruning ratio in different models. The results are in Table 3.

Table 3. Pruning ratio among different model types.

Prune environment	Pruning ratio (%)
LSTM CDP	22
LSTM ATDP	0

It is interesting to find the ATDP method did not prune any unit in all tests. The result is consistent with previous GC method NN ATDP tests. The reason is this ATDP is sensitive of vector length, which means it is not suitable for long vector angle calculation [3]. So, LSTM ATDP got the same test accuracy with LSTM Base. We also ignore this method in FN part.

The CDP works well in pruning experiments. It achieves 22% pruning ratio on LSTM network. However, back to the test accuracy, since the ATDP method does not effectively prune, we will ignore the results of this method. In table 2, the test accuracy of the network after pruning has declined around 6%. We believe that this phenomenon is reasonable. In the process of pruning, we have removed two types of units, one of which is to remove similar units. We achieve this by

adding the removed units' weight to the kept units. Another is the simultaneous removal of complementary units. We did not optimize the two removal methods quantitatively.

To be more specific, suppose we have three units, No. 1, No.2 and No.3. No.1 and No.2 are similar, and No.2, No.3 are complementary. We can infer that No. 1 and No. 3 are complementary to a certain extent but did not meet the screening criteria. The algorithm first prunes the complementary pairs and then prunes the similar pairs. From the results, No. 1 will be retained, and the weight is the sum of No. 1 and No. 2. With this imbalanced remove method, we only kept the weight of one part of the complementary pairs. This caused weight errors in the network and was reflected in the test accuracy rate.

3.2.2 Fine-tuning Effects

FN is introduced to eliminate the adverse effects of pruning. Using the grid search to find where is the best pruning time, the results are shown in Table 4.

Table 4. Pruning time's effects in test accuracy and pruning ratio.

Pruning epochs / total epochs	Test accuracy (%)	Pruning ratio (%)
0.6	53.33	24
0.7	53.33	24
0.8	56.67	25
0.9	60	26
0.95	66.7	21
1	60	22

In table 4, we define the pruning time as the percentage of total training epochs, when it equals to one, which means pruning applied after training finished. Also, when the number is 0.8, which means the pruning applied when 80% of training epochs are completed. When finished pruning, the re-training process will run for 20% of total epochs.

All experiments are in the same situation with LSTM CDP, hidden layer size equals to 20, and the total epoch number equals to 100.

The results show when doing early pruning, the CDP can get higher pruning ratio. This phenomenon is reasonable, because when applying early pruning, the network may not have converged. Therefore, the functional units may be pruned mistakenly. The test accuracy trend has proved this idea. When doing more early pruning and FN, the test accuracy becomes lower.

Besides, when applied CDP on 95% of training epochs, the negative impact on imbalanced remove method has been eliminated. The result shows it achieved the same test accuracy with LSTM Base as 66.7%. It proofs the theory of FN can increase the prediction accuracy of the pruned network and decrease the pruning loss.

4 Conclusion and Future Work

Overall, A new approach of "fight or flight" prediction is achieved by LSTM network. We find the LSTM networks are more reliable than traditional ML algorithms and basic NN in screening valuable features. In contrast, it is complicated to obtain an optimal network model. Besides, distinctiveness pruning method can effectively prune the LSTM network, and FN was proofed to maintain the network functionality. However, two things need to explore in future research. One is to adjust the ATDP method to eliminate the negative impact of increasing data dimensions. The other is to optimize the distinctiveness pruning algorithm with FN and automatically balance the remove units and select the best pruning time.

References

1. Jacobs, G.: The Physiology of Mind–Body Interactions: The Stress Response and the Relaxation Response. The Journal of Alternative and Complementary Medicine. 7, 83-92 (2001).
2. Derakhshan, A., Mikaeili, M., Nasrabadi, A., Gedeon, T.: Network physiology of ‘fight or flight’ response in facial superficial bloodvessels. Physiological Measurement. 40, 014002 (2019).
3. Huang, G.: Distinctiveness Pruning on ‘fight or flight’ Response Prediction Network. 3rd ANU Annual Bio-inspired Computing Student Conference. (2020).
4. Gedeon, T.D.: Indicators of hidden neuron functionality: the weight matrix versus neuron behaviour. In: Proceedings 1995 Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems. pp. 26-29. IEEE (1995).
5. Gedeon, T., Harris, D.: Network reduction techniques. In: Proceedings International Conference on Neural Networks Methodologies and Applications. vol. 1, pp. 119-126 (1991)
6. Tajbakhsh, N., Shin, J., Gurudu, S., Hurst, R., Kendall, C., Gotway, M., Liang, J.: Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?. IEEE Transactions on Medical Imaging. 35, 1299-1312 (2016).
7. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation. 9, 1735-1780 (1997).
8. Kingma, D., Ba, J.: Adam: A Method for Stochastic Optimization, https://arxiv.org/abs/1412.6980.
9. Gedeon, T.D.: Data mining of inputs: analysing magnitude and functional measures. International Journal of Neural Systems 8(02), 209-218 (1997)