Genuine/Posed Anger Detection by LSTM with Model Compression

Ruiqiao Jiang

School of Computing,
The Australian National University,
Canberra ACT 2601
u6818746@anu.edu.au

Abstract. Emotion is an important part of human society, and the judgment of human emotion has always been an enduring research topic. Anger is a strong emotion and identifying whether a person is really angry or pretending to be angry is a meaningful topic in human-computer interaction. With the rise of deep learning, the use of neural networks to classify emotions has become a trend. But too deep models will take up too much computing resources and consume too much time. To address this problem, we firstly train a one-layer fully connected neural network model and a LSTM model for emotion classification task. Then we apply model compression technique on both models to explore the balance of computing recourses consumption and classification accuracy. Results show that the LSTM has the best classification performance, which achieves 98.7%. The model compression method applied on both models is proved to be helpful to capture the most balanced model between accuracy and model cost.

Keywords: Neural Networks, Anger Dataset, Classification, Compression

1 Introduction

Recently, a lot of work has focused on emotional recognition. The computer analyzes and processes the signals collected from the sensor to get the emotional state of the other person. This behavior is called emotional recognition. From the point of view of physiological psychology, emotion is a complex state of organism, which involves both experience and physiological response, as well as behavior, and its composition includes at least three factors: emotional experience, emotional expression and emotional physiology.

At present, there are two ways for emotion recognition, one is to detect physiological signals such as respiration, heart rhythm and body temperature, and the other is to detect emotional behaviors such as facial expression recognition, speech emotion recognition and posture recognition. In [1], they asked 20 subjects to watch 20 videos about anger and recorded their pupillary responses with time frames. Then the trained a model by these data to classify whether the video is about real anger or not. Results showed that the estimation accuracy is 95%, whether the accuracy of verbal responses is 60%. Because of the great interest in human emotion analysis, we selected this Anger dataset including anger_v1 and anger_v2, which are about human pupil data.

As far as model selection is concerned, we select a lightweight fully connected network [2] on anger_v1 that contains a small number of features in the data set. We also use a LSTM model [10] on anger_v2 that contains a more complex time-series data. However, in the field of deep learning, both the fully connected network model and the LSTM model can contain an extremely large number of parameters, which makes it difficult to operate normally under the limited computing resources. Similarly, it is too time-consuming to get good results in a limited time.

In order to solve this problem, network pruning technology has been put forward. Although the network is deep and wide, there are lesser neurons that really contribute in the neural network model. So, we need to get rid of these extra parameters, and this technique implements an algorithm to determine which neurons are useful and which are redundant.[3] This can significantly reduce the number of parameters of the model and make the calculation more efficient. But it is inevitable that some of the accuracy will be lost. At present, this technology has been applied to many different fields. In [4], they applied the model compression method to the image compression task and achieved good results.

In this paper, both fully connected network and LSTM model are used to classify anger, and the method of model compression is applied on both models to make the model lightweight with limited loss of accuracy.

2 Method

2.1 Dataset

In this paper, two anger datasets are selected in order to compare the performances of different types of models for anger expressions' authenticity detection.

The first selected dataset is named anger_v1. This data set collected the pupil response data of 20 subjects after watching 20 videos where ten of them are genuine anger and ten are posed anger. This brings together a total of 400 pieces of data,

and each data has 8 features: video name, mean, the standard deviation, the change of left pupillary size, the change of right pupillary size, d1 in PCA [5], d2 in PCA and label.
In data processing stage, video name attribute was removed due to the reason that it barely relates the prediction target. After removing the label attribute, six attributes were utilized for this binary classification task. Although the dataset has been pre-processed, after careful inspection, each feature does not strictly meet the standardization. Hence, Batch Normalization technique [6] was employed in PyTorch [7] to standardize the input data to eliminate the data bias of the network.

The second dataset, namely anger_v2, is the captured pupillary distances of both eyes of 20 participants while they were viewing 10 videos of genuine anger and 10 videos of faked anger. This dataset is comprised of 780 data records in total, with each row tagged with the label of 'genuine' or 'fake' of the corresponding video. These pupillary distances were recorded by a special device named an eye tracker at a fixed frequency, and the experimented videos have variational lengths. Hence, the lengths of pupillary distance vectors have diverse distributions. Fig. 1 displays the distribution of time frame lengths of two types of videos.

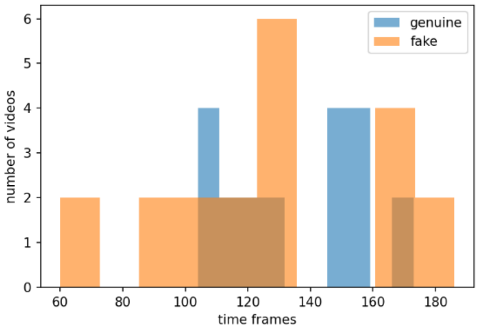

Fig. 1 The distribution of time frame lengths of videos

To demonstrate the data characteristics, Fig.2 is presented to visualize the pupillary distance variations of four subjects from different race. It is obvious in the figure that PD variations patterns are different as the time frame increase between videos with genuine anger expressions and ones without. There patterns along with time frames are critical, which we determine to apply LSTM [12] to capture these patterns and distinguish the differences of two types of PD response vectors.

Before the LSTM training stage, we shuffled the data and spitted the training and test set with the test size ratio equalizing 0.1. Meanwhile in the FCNN training stage, we set the test size ratio as 0.2 for its smaller size of data. Random seeds were fixed in the experiments for reproduction and avoiding deviation of the prediction result.

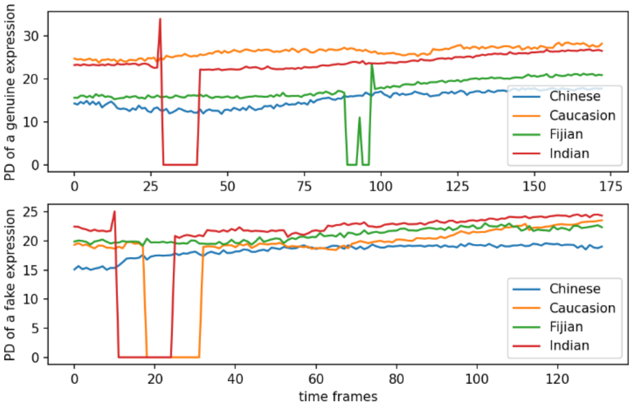

Fig. 2. Pupillary distance changes when subjects were viewing different videos

2.2 Network Architecture

In this section, we applied two neural networks from different architectures based on the characteristics of the corresponding dataset: single layer fully connected neural network and LSTM.

Targeting dataset anger_v1, a fully connected network composed of one hidden layer and one output layer. The reason of the model setting is because the dataset size is relatively small, and features are simple.

The structure of the fully connected network is visualized in Fig. 3. The whole model is divided into two parts, namely Fully Connected layer and Classification layer.

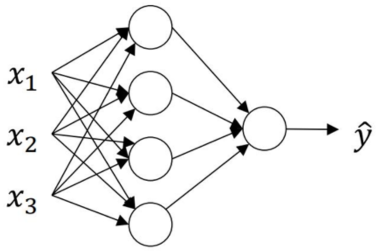

Fig. 3. Fully connected network structure.

Fully Connected layer. This architecture takes a 6 input, and *x* unit hidden layer which results to *x* output. Before input, I used the batch-norm method with dimension 6 to standardize the data. Then use Leaky Relu [8] as the activation function to apply to the output.

Classification layer. This architecture takes a *x* input, and 2-unit hidden layer which results to 2 output. The output here is the binary prediction of the data by the network.

We built a LSTM model with three hidden layers and one classification layer to classify the labels in the second dataset. The reason for this option is mentioned before, the second dataset resembles time series data, at the meantime LSTM is proved to have a good performance when predicting time series data. The overall structure is as Fig.4

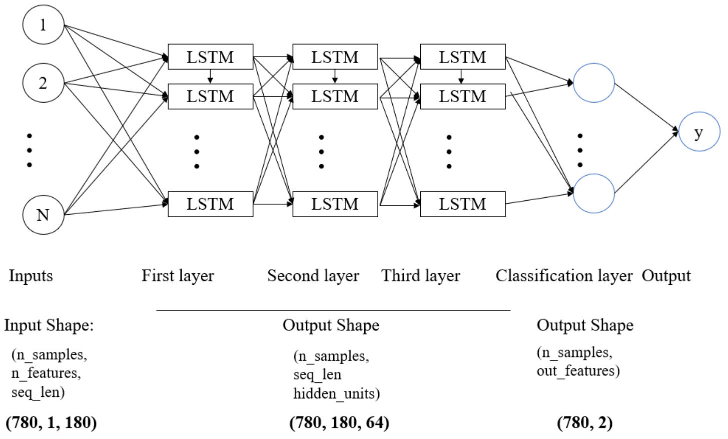

Fig. 4. LSTM structure.

LSTM layer. 3 LSTM layers were stacked in the model with each layer containing 64 hidden units. It is observed that the input data has various lengths, ranging from 61 to 186 time frames. We adopted 180 as the sequence length and padded all input vectors to this length. In addition, dropout [13] technique was employed in each layer to prevent over-fitting.

Classification layer. This module takes the output of the last LSTM layer and produces the probability vectors as the model output. It was designed with Linear layer with Pytorch.

2.3 Loss Function and Optimizer

Cross entropy is used as the objective function of training. It is calculated by formula 1. where *N* is the total number of training set, y_i is the real label and p_i is the output of the network.

$$L = \frac{1}{N}\sum_{i} -\left[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)\right] \qquad (1)$$

Adam optimization algorithm [9] is an extension of random gradient descent algorithm. Recently, it is widely used in deep learning applications, especially in computer vision and natural language processing tasks. Empirical results show that Adam algorithm has excellent performance in practice and has great advantages over other random optimization algorithms. So, in this paper, Adam is chosen as the optimizer.

2.4 Evaluation method

An appropriate verification strategy can reflect the actual performance of the network. Therefore, it is essential and critical to choose an appropriate testing strategy. Because the dataset is balanced, it does not need to consider complex verification strategies, and because it is a simple binary classification task, the simply use of test accuracy can be used as a fair testing method. Specifically, test accuracy is calculated as follows:

$$test\ accuracy = \frac{TP + FN}{num\ of\ test\ samples} \tag{2}$$

Where TP and FN represent numbers of true positive and false negative samples, respectively.

2.5 Model Compression

Model compression methods are proposed by researchers to lower down the resource's consumption during model training without significant drop in accuracy. Current popular model compression methods include parameter pruning, low-rank factorization and weight quantization, etc. Parameter pruning technique is applied in this section to reduce the size and inference time of a trained machine learning model.

T.D. Gedeon [4] promoted an intuitive pruning idea from the perspective of neuron's sensitivity and similarity. Namely, if two neurons produce highly similar output vectors on the training samples, they can be considered redundant neurons with identical functions. Experiences have suggested that redundant neurons normally don't have high sensitivity to the prediction accuracy. Therefore, proper removal towards these redundant neurons will definitely reduce model size and save computation time with limited accuracy loss.

The angle between the two vectors can well reflect the relationship between them, which makes it the similarity index between two neurons. If the angle is less than 15 degrees, it can be considered that the two vectors are almost the same, and if the angle is greater than 165 degrees, it can be considered that the two cancel each other out. In order to verify the effect of the compressed model, we can obtain the output vector of each neuron in the network and calculate the angle between all the vectors.

In this paper, to achieve the balance of model size and prediction accuracy, we conduct units pruning on both models. The angles between all vectors are sorted, and the mean values of the smallest five angles are obtained as the criteria for judging the uniqueness of neurons in the model. If the angle is too small, the model can be further compressed, that is, to reduce the number of neurons in the full connection layer. If the angle is appropriate and the loss of accuracy is small, it can be considered as the most balanced model.

3 Results and Discussion

In this section, a total of two stages of experiments are carried out. In the first stage, we firstly trained the fully connected network on the anger_v1 dataset, then trained the LSTM model for emotion classification on the anger_v2 dataset. Models' performances were compared in the first stage of experiments. In the second stage, we employed the forementioned model compression technique on both models to explore the balance of resource consumption and prediction accuracy.

3.1 Emotional Binary Classification task

Table 1. The results of two classifications on the Anger dataset.

Dataset	Method	Highest Accuracy
Anger_v1	Original [1]	95.00%
Anger_v1	1-layer FCNN	97.50%
Anger_v2	**LSTM**	**98.7%**

Experiment settings. In the anger authenticity classification task, we adopted different models on different datasets to compare the prediction accuracies. When training one-layer fully connected neural network, we randomly split the dataset into a training set with 320 samples and test set with 80 samples. In data loader, batch size is set to 80, so that an epoch will have four updates in the batch gradient direction, and epoch is set to 300. We select the Adam as the optimizer with the learning rate of 6e-3. Differently, LSTM training is more complex than FCNN training for its larger number of parameters. The experimented LSTM model contains three LSTM layers with 64 hidden units in each layer and one classification layer. After several training experiments of LSTM, we finally adopted 8 as the batch size, 180 as the input sequence length and 300 epochs. Regarding the optimizer and learning rate, Adagrad [14-15] is selected to be the adaptive optimizer, and learning rate is set to 4e-3. To prevent over-fitting, we also applied 20% neurons dropout in each LSTM layer.

Fig. 5 and Fig. 6 displays the training/testing loss and accuracy along with the increase of epochs of two models respectively. It can be seen that both the training loss and the test loss decrease with the training process of two models. Meanwhile, LSTM's performance is observed to fluctuate at a larger amplitude compared to the simple FCNN. Furthermore, it can also be seen that the training accuracy and test accuracy have been converged in both graphs, which shows that there is no problem of over-fitting and or under-fitting in the model.

In terms of the highest test accuracy, LSTM model is the champion, which achieves a 98.7% of prediction accuracy. The final result of FCNN model is 97.5%, which has exceeded results of the basic model [1], which is 95%, The results are summarized in Table 1. These results imply that for the anger emotion classification, using LSTM along with original time series alike data has better performance than applying FCNN on a feature-engineered data that ignores the time series characteristics.

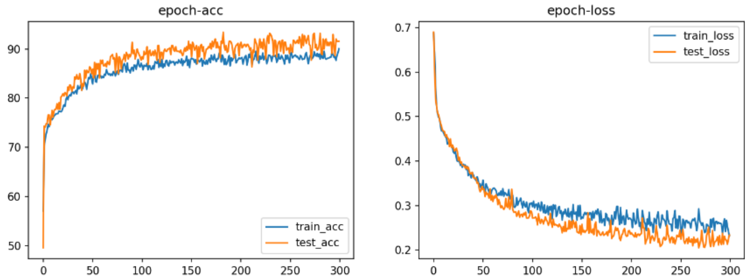

Fig. 5. One-layer FCNN training and evaluation

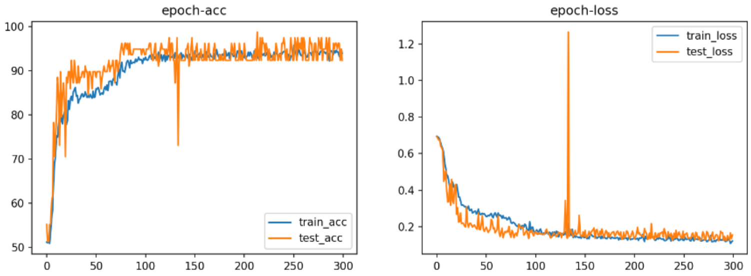

Fig. 6. LSTM training and evaluation

3.2 Model Compression task

In the second stage of experiments, model compression technique is employed on both 300-epoch trained models. To explore the tradeoff process of computation consumption and classification accuracy, we use the average test set accuracy of the last trained epochs for the reason that the emergence of the highest accuracy may be accidental, and angle of output vectors as the neuron's redundancy index. In order to obtain the output vectors of all neurons, all the training data is utilized as stimulus input, so as to extract the output vector from the model for angle calculation.

The model compression results of two models are shown in Fig. 7 and Fig. 8 below.

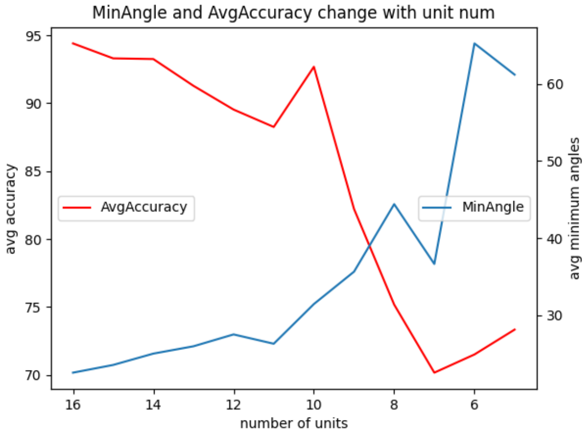

Fig. 7. The average 5 minimum angles and the average accuracy of last 100 epochs changes with numbers of units in FCNN.

For FCNN experiment, with model compression starting from 16 unit, we can see that with the decrease of the number of units, the average accuracy shows a downward trend, while the minimum angle value shows an upward trend. Result reveals that there exists a critical point, after which the accuracy has been greatly reduced, while the angle value has been greatly improved. And this point is the best balance point in model compression. In FCNN experiment, this point is when unit number is 10. At this point, the number of units decreased by 37.50%, but accuracy decreased by only 1.82%.

Model compression experiment with LSTM model is more complicated. There is no apparent upward trend for average 5 minimum angles, but the downward trend in terms of average accuracy is shown in the figure. The most balanced model is observed to be the 56-neuron model, which means 10 neurons can be further reduced to shrink the model size without hurting the prediction accuracy. This may imply us that LSTM has a powerful representation ability with long sequences data.

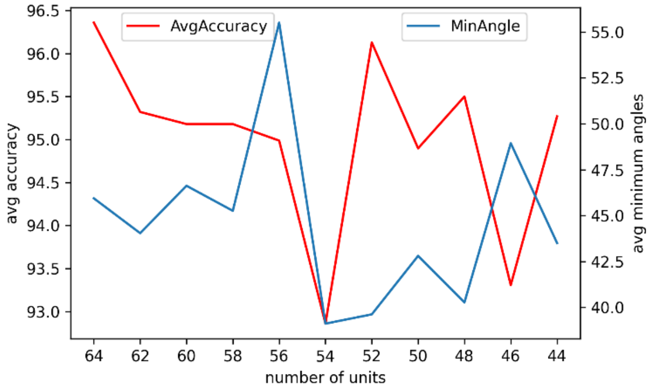

Fig. 8. The average 5 minimum angles and the average accuracy of last 100 epochs changes with numbers of hidden units in each LSTM layer.

4 Conclusion and Future Work

The above experiments show the effectiveness of our both models. The classification accuracy of FCNN on the anger_v1 dataset is up to 97.5%, and the LSTM on anger_v2 achieves 98.7%, which are much better than 95% of the basic model [1]. In the part of model compression, the best pruning critical points of two models are successfully found, which reduces the complexity of the model without too much affecting the accuracy.

There are also some limitations in this work. For example, dataset size is not sufficient for deep learning models, and model compression method on LSTM takes too long even with the GPU computation power. As for future work, we might consider applying data augmentation techniques to enlarge the dataset size for deep learning models with better generalization ability and better prediction accuracy. It is also can be considered to further explore the relationship between the various features in the data set to expand more feature dimensions, so that we can use a more efficient model for modeling and training.

References

1. Chen, Lu, et al. "Are you really angry? Detecting emotion veracity as a proposed tool for interaction." Proceedings of the 29th Australian Conference on Computer-Human Interaction. 2017.
2. J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
3. Gedeon, TD, Harris, D, “Network Reduction Techniques,” Proc. Int. Conf. on Neural Networks Methodologies and Applications, AMSE, San Diego, vol. 2, pp. 25-34, 1991.
4. Gedeon, TD, & Harris, D “Progressive Image Compression,” Proceedings International Joint Conference on Neural Networks, vol. 4, pp. 403-407, Baltimore, 1992.
5. Wold, S., Esbensen, K., Geladi, P., 1987. Principal component analysis. Chemometr. Intell. Lab. Syst. 2, 37–52.
6. S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015.
7. Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In NIPS, 2019.
8. A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In ICML, 2013.
9. D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.
10. Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural networks. 2005 Jul 1;18(5-6):602-10.
11. Gers FA, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM. Neural computation. 2000 Oct 1;12(10):2451-71.
12. Greff K, Srivastava RK, Koutník J, Steunebrink BR, Schmidhuber J. LSTM: A search space odyssey. IEEE transactions on neural networks and learning systems. 2016 Jul 8;28(10):2222-32.
13. Baldi P, Sadowski PJ. Understanding dropout. Advances in neural information processing systems. 2013;26:2814-22.
14. Mukkamala MC, Hein M. Variants of rmsprop and adagrad with logarithmic regret bounds. InInternational Conference on Machine Learning 2017 Jul 17 (pp. 2545-2553). PMLR.
15. Lydia A, Francis S. Adagrad—An optimizer for stochastic gradient descent. Int. J. Inf. Comput. Sci.. 2019 May;6(5).