Implement Neural Network on Grammatical Facial Expression Recognition

Yangyang Xu

The Australian National University Acton ACT 2601,
u6325688@anu.edu.au

Abstract. The facial landmarks generated by Kinect can be used to build a neural network model. By using such model, machine can recognize the grammatical facial expression. When the input data is too much, and many hidden neurons were used, based on the distinctiveness of pruning algorithm, neuron can be reduced, it improves running time of the simple neural network.

Keywords: Neural Network, Kinect, Distinctiveness, Grammatical Facial Expression, Hidden Layer

1 Introduction

The rapid evolution of Technology is for following the needs of people. Either Siri or Google Home makes a good market by providing the portable AI (artificial intelligence) voice service. However, for people who have a deaf or dumb problem, the voice assistant is useless, because they use sign language to communicate. Currently, the grammatical facial expression (GFE), as an efficient form of non-manual sign languages, which can be identified in semantic level by Kinect sensor, a motion sensing device produced by Microsoft. The RGB camera of Kinect captures a frame of 3D face as Figure 1[1] displayed. The Face Tracking SDK of Kinect helps generate the video facial tracking landmarks files with the x, y, and z coordinators. Then these facial landmarks will be used in further research.

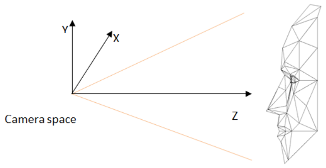

Fig. 1. 3D Face

The facial landmarks data used in this paper is from the "Grammatical Facial Expression Data Set", which is provided by the UCI Machine Learning Repository website. This data repository is completed and well formatted, it does not lose any data. It has plenty of data, 27965 instances, although, these data are only from two people, signer A and signer B. Each signer acted nine types of grammatical facial expressions: Affirmative, Conditional, Doubt Question, Emphasis, Negative, Relative, Topic, Wh Question, and Yes/No Question. Each type of facial data includes two files. One file has data points with timestamps, which represent video frames recorded by Microsoft Kinect Sensor; 100 attributes for each x, y and z. Another file has one digit (0 or 1) at each line to label if the video frame in the previous file representing such type of facial expression [2].

There are several recent papers [3] used this set of data. Freitas et al. are the first researchers, who provided and used the data by training data on Multiple Layer Perceptron (MLP) in 2014. In 2015, Uddin [4] reported his novel results of these data by using Ada-Random Forests algorithm. In the following year, Bhuvan et al. [5]tried Random Forest, MLP Classifier 10, and Bagging on these data, but “Assertion” and “Focus” datasets are not included from this year. In 2017, Walawalkar reported the results running by Convolutional Neural Network (CNN). At the same year, Freitas et al. [6] improved recognition accuracy based on their first paper.

As the qualities of the repository and frequency of use mentioned before, this data set is chosen in this paper. Since the labels are provided for each type of facial expressions, the supervised binary classification is adopted to train a neural network. Artificial neural network (ANN) is inspired by the neural network of biological brain and applied to pattern classification. The MLP and CNN mentioned above belong to it. The single hidden-neuron layer, a simplest ANN feed-forward network, it represents a net allows to learn patterns by transferring data from input layer, to one hidden layer with several artificial neurons, finally to output layer [7]. A hidden neuron layer has an activation function for building non-linear model, such as sigmoid and softmax. These activation functions can use the total weight of inputs of a hidden neuron to calculate an output. For a supervised learning, it means the ANN can classify data by using labelled data to train a model, output layer has number of output vectors, which is same as the number of distinct labels (e.g. binary label 0 / 1). After training, the ANN model will give a label for an input, based on the highest value among output vectors [7, 8].

To improve an ANN model, there are three methods are used in this paper. The first is using optimisation function, such as Adam algorithms, it optimises and updates the model hyper-parameters by the first-order gradient-based function [9]. The second is validating model to get improvement indicators and directions. K-folder cross validation or hold-out can be used. Normally 10-folder is the ideal evaluation algorithms, it measures more accuracy performance than hold-out. But hold-out has simpler implementation and avoids using trained data [10]. The last method is pruning network, this method helps reduce redundant hidden neurons to boost the training efficiency [11].

The following content structure will be: Explain the approaches of pre-processing and training data on NN. Demonstrate and discuss the recognition accuracy and setting of hypermeters by comparing the latest paper of Freitas et al. Talk about the improvement by adopting Gedeon’s network pruning training strategy. Finally, Summarise this experiment and state the work for future.

2 Method

1.1 Pre-process Data

In the data resources, the timestamps and headers will not contribute to the process. Thus, all timestamps will be eliminated. As the reference results shown, highest F-score got when the training data and validation data are from the same signer. However, in real life, there will be more signers that should be recognised. Thus, the input should be combined with both signers. The targets and data points are separated into two files. For data points file, the signer B data will append after singer A, then do the same process in the target file.

However, by testing the first frames of signer A and signer B, which are from the "Affirmative" data file via MATLAB, the coordinators of signer A and Signer B are not in the same range. By using the signer A's nose and eye irises as reference points to translate Signer B, different recoding frames and non-static nose point are the issues to translate frames.

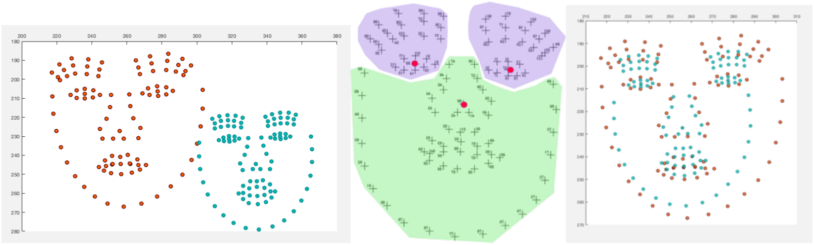

Figure 2. Translate facial parts

Thus, to use better input for NN model, this method will refer to the method from the first paper of Freitas et al.'s [3]. There are 7 angles (A1 – A7) and 11 distances (D1 – D11) will be extracted and calculated by x and y coordinators. To reduce the effect of non-regular face size, the distances of Signer A and Signer B will be normalised by Z-score separately, since the z-score only represent distributions among whole dataset of signer A or Signer B.

A1	**A2**	**A3**	**A4**			
{27,2,10}	{17,10,2}	{48,89,54}	{89,48,51}			
A5	**A6**	**A7**				
{89,54,51}	{51,48,57}	{51,54,57}				
D1	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**
{17,27}	{17,2}	{2,89}	{89,39}	{39,57}	{51,57}	{48,54}
D8	**D9**	**D10**	**D11**			
{44,57}	{44,89}	{89,10}	{10,27}			

Table 1. Specified coordinators for angles and distances

Assume distance D has two nodes, (x_1, y_1) and (x_2, y_2), then Euclidean D:

$$D = \sqrt{|x_1 - x_2|^2 + |y_1 - y_2|^2} \tag{1}$$

Assume the 1D matrix M, which includes the SignerA's D1 (distance) of all frames, N is the amount of frame, then the z-score for each distance D is:

$$Z - Score = \frac{D - \bar{M}}{\sqrt[2]{\frac{\sum(D-\bar{M})^2}{N}}} \tag{2}$$

Assume the angle A is combined by 2 distance elements, D1 and D2, where $D1 \cdot D2$ is a dot product, $\|D_i\|$ is the norm of D_i :

$$\cos A = \frac{D1 \cdot D2}{\|D1\|\|D2\|} \tag{3}$$

The input vector v for neural network will be 1D, it has 118 attributes: 11 distances, 7 angles, all Z coordinators (100), each input vector represents one frame of recording:

$$v = \{D_1, \ldots D_{11}, A_1 \ldots A_7, Z_1 \ldots . Z_{100}\} \tag{4}$$

1.2 Neural Network

The experiment code was built by Pytorch neural network module, since Pytorch provides many build-in functions and it is compatible with python matrix (NumPy) and Tensor. In this paper, the basic model used 1 input layer, 1 hidden layer and 1 output layer. This Model is used to train all categories of facial expressions. Model adopts 15 hidden neurons, 2 output neurons. The learning rate and epochs are 0.0005 and 1200. Since this problem will only output 0 or 1, a straightforward activation, sigmoid, is used in hidden layer. Adam as a popular optimiser, this paper used it to help optimise hyperparameters. The loss function is the Cross Entropy which is used to tell the differences between a vector and output label.

The Loss-Epoch graph can evaluate the model. The evaluation method is based on holdout. There are 80% of original data will be used as a training set, the 20% of original data will be used as a testing set and validating set. Both sets should be formed by randomly picking data entries. The loss values of training and testing sets can be obtained at the end of each epoch running. When the testing loss line is above training loss line, it means the model is overfitting; then we can decrease the epochs until both loss lines have the same decreasing trend.

The accuracy of the testing is calculated by the confusion table generated by python code. There are P positive instances and N negative instances. TN means the prediction and actual output are positive; FN means the prediction is positive, but actual value is negative; FP means the prediction is negative, but actual value is positive; TP mean both prediction and actual values are negative [12]. Learning rate can be improved by knowing the accuracy.

	Predicted Positive	Predicted Negative

Actual Positive	TN	FP
Actual Negative	FN	TP

Table 2. Confusion Matrix Table

$$Accuracy = \frac{TN+TP}{P+N} \tag{5}$$

Once the basic model made in this experiment is corrected by this evaluation. Next experiment will improve this model.

1.3 Optimize Hidden Layer

The pruning process based on distinctiveness is similar to Gedeon and Harris's. This experiment hopes use the pruning process to make a model have more accurate and quicker performance. Firstly, use basic model training and testing one type of 9 facial expressions. Then there are two weights matrix can be retrieved, one is the weight between input layer and hidden layer, another one is the weight between hidden layer and output layer. Because current weight matrix of output is processed by the activation function in the hidden layer, then it will be extracted to determine which hidden neurons can be removed. In this experiment, since the output is binary and there are 15 hidden neurons, the matrix size is 2×15 and each neuron unit has 2 elements. Next, compare the angles between each neuron unit and remove the unit that has angle is below 15 degrees or above 165 degrees. For example, vector $v1$ and vector $v2$ have an angle of 12 degrees, which is below 15 degrees, either $v1$ or $v2$ needs be removed. Then, get the new weight matrix from the left neuron units. Finally, run the basic model with new weight matrix and the number of remainder neurons.

To compare the angle conveniently, combination concept will be used to generate the neuron unit pairs, such as $(v1, v2), (v2, v3), (v3, v4)\ldots$, etc. When there are 15 hidden units, the number of total pairs: $\frac{15!}{2(15-2)!} = 105$. The angle calculation is same as the formula 3.

2 Results and Discussion

Table 2 displays the accuracies from the basic neuron network, basic neuron network with pruning and results which are from latest paper. There are 4 expressions have better accuracy than papers', however, after pruning, only Wh question has higher accuracy than paper's. The model with pruning method has lower accuracy than basic one. Although the differences between each model are not significant.

GFE	Basic Model	Basic Model with Pruning	Model from paper[13]
Affirmative	0.8979↑	0.8446	0.8773
Conditional	0.9470	0.9237	0.9534
Doubt Question	0.9411	0.8965	0.9700
Emphasis	0.9338	0.8902	-
Negative	0.8939	0.8731	0.9582

Relative	0.9634	0.9620	0.9759
Topic	0.9665↑	0.9521	0.9544
Wh Question	0.9320↑	0.9010↑	0.8988
Yes/No Question	0.9343↑	0.9065	0.9412

Table 3. Summary of experiments accuracies

In the basic model, the evaluation approach helps the model avoid overfitting. In the beginning, the basic model has 5000 epochs, after testing with different multiple epochs, most of the training and testing line has been improved. The learning rate was 0.001, according to the accuracy, the learning rate is changed to 0.0005.

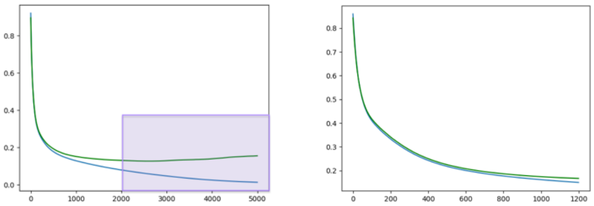

Figure 3. Overfitting Example (Loss-Epoch graph, green: testing; blue: training)

Among the categories of facial expressions, there is an interesting result. Only “Empathsis” has the fluctuates on loss values. Others have smooth loss lines.

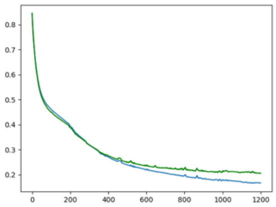

Figure 4. Empathsis Example (Loss-Epoch graph, green: testing; blue: training)

After using the distinctiveness to prune the network, most of the cases, 80% of neurons will be removed, when validate angle range is from 15 degrees to 165 degrees. But even increase the valid angle ranges from 5 degrees to 175 degrees, the accuracy is not improved. Most of time, fewer neurons make testing loss line close to training loss line. But, in the model with pruning, the loss line has higher values than the model. Therefore, in this problem, pruning hidden units can only improve overfitting model, let computers calculate fewer weights, while it cannot increase the model accuracy.

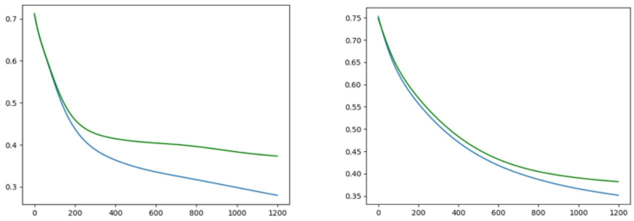

Figure 5. Model with pruning method (Loss-Epoch graph, green: testing; blue: training)

3 Conclusion and Future Work

This study aimed at recognizing the grammatical facial expression by ANN. In the end, the testing accuracies are not too bad to recognize a facial expression based on its own repository. Pre-process is a heavy but important work because it can help ANN model train better by pointing the direction manually, although, we need to analyse the data and find the appropriate way to clean data. The attributes were 300, after preprocessing, the attributes decreased to 118. In this study, the pruning process is not helpful on accuracy, less hidden layer may be the issue. But it may be a good strategy when a model presumes avoiding overfitting.

Considering the performances of all ANN models, the input data needs to contain more information, which helps a model classify more significant differences to boost the accuracy. In the further, because the repository is about image and time series, the deep learning method, such as convolutional and Long-term short memory can be applied on this data repository. Also, people can use the machine to communicate with others by grammatical facial expressions.

4 References

[1] Microsoft, "Figure 1. Camera Space," 2018, https://msdn.microsoft.com/en-us/library/jj130970.aspx.

[2] "UCI Machine Learning Repository: Grammatical Facial Expressions Data Set," https://archive.ics.uci.edu/ml/datasets/Grammatical+Facial+Expressions.

[3] F. d. A. Freitas *et al.*, *Grammatical Facial Expressions Recognition with Machine Learning*, 2014.

[4] M. T. Uddin, "An Ada-Random Forests based grammatical facial expressions recognition approach." pp. 1-6.

[5] M. S. Bhuvan *et al.*, "Detection and analysis model for grammatical facial expressions in sign language." pp. 155-160.

[6] F. Freitas *et al.*, *Grammatical facial expression recognition in sign language discourse: a study at the syntax level*, 2017.

[7] A. K. Jain, M. Jianchang, and K. M. Mohiuddin, “Artificial neural networks: a tutorial,” *Computer,* vol. 29, no. 3, pp. 31-44, 1996.

[8] M. W. Gardner, and S. R. Dorling, “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences,” *Atmospheric Environment,* vol. 32, no. 14, pp. 2627-2636, 1998/08/01/, 1998.

[9] D. P. Kingma, and J. Ba, "Adam: A Method for Stochastic Optimization," *ArXiv e-prints*, https://ui.adsabs.harvard.edu/#abs/2014arXiv1412.6980K, [December 01, 2014, 2014].

[10] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-Validation," *Encyclopedia of Database Systems*, L. Liu and M. T. ÖZsu, eds., pp. 532-538, Boston, MA: Springer US, 2009.

[11] T. Gedeon, and D. Harris, “Network Reduction Techniques,” *Proceedings International Conference on Neural Networks Methodologies and Applications,* vol. 1, pp. 119-126, 1991.

[12] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recogn. Lett.,* vol. 27, no. 8, pp. 861-874, 2006.

[13] F. Freitas *et al.*, “Grammatical facial expression recognition in sign language discourse: a study at the syntax level,” 2017.