Implement Long Short-Term Memory Recurrent Neural Network on Grammatical Facial Expression Recognition

Yangyang Xu[1]

[1] The Australian National University Acton ACT 2601
u6325688@anu.edu.au

Abstract. This paper experimented with basic Long Short-Term Memory (LSTM) recurrent neural network on recognizing the Grammatical Facial Expressions Data Set [1]. By using the data-preprocessing, it guarantees the LSTM model can have 90.84% average accuracy among 9 different facial expressions. "Affirmative" and "Wh Question" sign language gain better recognition via LSTM model described in this paper, by comparing to the result got by 1-hidden-layer neural network.

Keywords: LSTM, Recurrent Neural Network, Kinect, Grammatical Facial Expression, Deep Learning

1 Introduction

Grammatical Facial Expression (GFE), a kind of communication sign language which often used by people with impaired hearing [2, 3]. Combination of the hardware and software solutions can make machines learn the semantic meaning of such sign language. For the hardware part, Kinect [4], as known as a motion sensor device produced by the Microsoft, it can capture the facial expressions. For the software part, Kinect can generate facial contour landmarks (x, y and z-axis) for each frame of recording via the Face Tracking SDK [4]. The bio-inspired deep learning algorithms can be applied to these landmarks and recognise some patterns among them.

Deep learning can help with classifying a facial expression from a time-series-based recording. In deep learning field, because Recurrent Neural Network (RNN) can memorise data during a sequence of time and process as a multilayer feedforward network, it is a suitable architecture to train time-series based dataset [5]. In recent work, the extensions of RNN are wildly used. Long Short-Term Memory (LSTM) is an improved extension of RNN, its gradient-based approach (adding 3 gates) can prevent the influence of the gradient that vanished or exploded in further steps [6].

This paper is structured as follows: 1.1 describes the data used in this paper; 1.2 examines related work of same data and LSTM model; The implementation of data pre-processing showed in the 2.1 section; 2.2 defines the LSTM model and describes

training methods; 2.3 briefly describes the testing and validation approaches; Section 3 includes analysis of results and comparison with results yield from Multiple Layer Perceptron. The conclusion of this paper and further work will be in Section 4.

1.1 Data

This data repository used in the following experiment is provided by the UCI website. In this repository [1], there are 9 types of facial signs of 2 signers' (Signer A and Signer B): Affirmative, Double Question, Negative, Wh Question, Conditional, Yes/No Question, Empathies, Relative and Topic. Each facial expression consists of one data points file with timestamps and one file with binary labels (0 or 1). In the data file, the x, y and z coordinators of each landmark are listed as features (columns). The labels are contained in each row for each instance.

The quality and quantity of data are the reasons to choose this repository. For the quality part, this dataset is complete; there is no need to clean data set. These landmarks can be used to represent an expression during a period. For the quantity, averagely, there are 2500 instances can be used in training process for each facial expression, the 300 attributes also make the deep learning get sufficient exploration on features.

1.2 Related Work

There are some recent papers used the same data repository as this paper. Multiple Layer Perceptron (MLP) architecture is firstly implemented on this repository by Freitas et al. [3, 7], they did not get very high F-score on "Negative" and "Relative" expressions, most of facial expressions can be recognised above 0.75 (F-score). In 2016, by extracting the essential facial points, Bhuvan et al. [8] improved the F-score of MLP model to above 0.89. In 2017, a deep learning architecture, Convolutional Neural Network (CNN) is used by Walawalkar and Devesh [2]; their model has excellent performance on each facial expression, all of facial expressions can have over 0.94 F-score.

The preliminary experiment of this paper used two methods; The first one used the simplest neural network, the final average accuracy for 9 grammatical facial expressions is about 93.44%; The second one used the same model as first one, but with distinctiveness pruning strategy [9], the final average accuracy is lower than 93.44%, thereby, this paper adopts the extension(LSTM) assumed in previous work to show if LSTM can have better recognition than 1-layer-hidden neural network (first method).

There is no paper found that used the LSTM/RNN to recognise this repository, but there are two papers found used LSTM to train the similar facial marks. In Behzed and Mohammad's research[10], they used the CNN to extract landmarks from videos, then they used LSTM to memories and train the landmarks; their final model can successfully recognise variety facial expressions from 4 data repositories. Alex et al. [11] used unidirectional LSTM for 116 facial landmarks, the final expression recognition mean error rate is $18.2 \pm 0.6\%$.

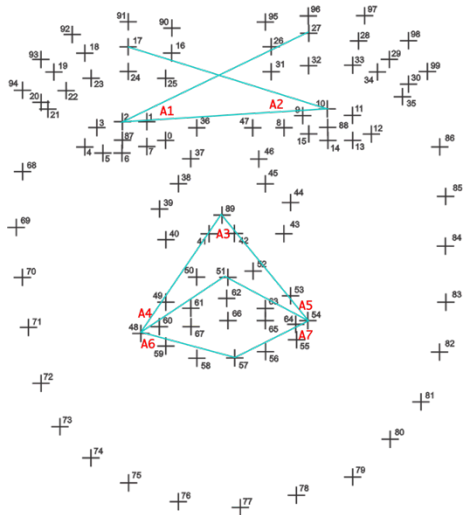
2 Methods

2.1 Data Pre-Processing

Reconstructing instance by 11 distances, 7 angles and 100 (whole) of z-coordinators can help classifier find the patterns; therefore, an input entry vector for deep learning model has 118 attributes with 1 dimension. The distance and angles showed in Table 1 and Figure 1, are specified in Freitas et al.'s paper [3]. Since there are two signers, each signer performed 9 facial expressions, the two data files of both signers should be combined as one data file and normalised by Z-score method [2].

Table. 1 Summary of accuracies from three experiments

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}			



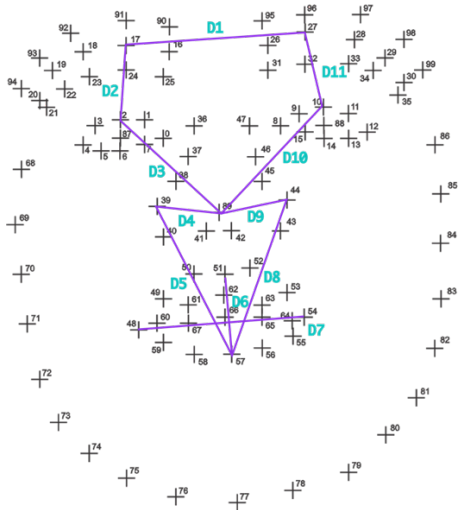


Fig. 1 Top image: 7 angles; Bottom image: 11 distances

Each facial landmark has x, y, and z coordinators, the Euclidean distance between two landmarks uses x and y coordinators of two landmarks, which are (X_1, Y_1) and (X_2, Y_2). Thus, the Euclidean distance (D) is calculated by [12]:

$$D = \sqrt{|X_1 - X_2|^2 + |Y_1 - Y_2|^2} \quad \textbf{(1)}$$

The Signer A and Signer B have their 1D matrixes, M_A and M_B, for one of specified Euclidean distances. Because the Z-score [2] can show the scaled distributions for different signers, which make the values are comparable, each distance instance of M_A and M_B will be calculated independently by the same function, where D is a distance instance, $\bar{M}$ is the mean of the 1D distance matrix, N is the number of instances in the 1D distance matrix:

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

The angle A (in cosine) between two distances (D_1 and D_2) can be calculated by following function, where $D1 \cdot D2$ gets dot product, $\|D_1\|$ and $\|D_2\|$ are the norms of $D1$ and $D2$ respectively:

$$\cos A = \frac{D1 \cdot D2}{\|D1\| \|D2\|} \quad \textbf{(3)}$$

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:

$$v = \{D_1, \dots D_{11}, A_1 \dots A_7, Z_1 \dots . Z_{100}\} \quad \textbf{(4)}$$

2.2 Deep Learning Model for training

The aim of this experiment is using supervised binary classifier with LSTM architecture to classify if a frame of facial landmark belongs to a category of grammatical facial expression. As the Fig. 2 displayed, there is only one hidden layer, LSTM layer. Fully Connected (FC) layer is used as an output layer.

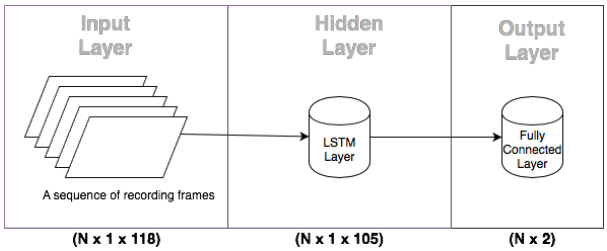


Fig. 2 Training Model Architecture

For the input layer, since each instance vector has a related target label, the total N frames (training data) of 1D input entry vector with 118 attributes which given by the section 2.1, will be sent to LSTM layer. For the hidden layer, there are 3 gates and 2 memory cells used for this gradient-based approach of LSTM, which are input gate (I_t), forget gate (F_t), output gate (O_t), new memory cell ($NewCell_t$) and final memory cell ($FinaCelll_t$) [6]; In the following formulas [10], the "t" means a timestamp, "σ" is the sigmoid function, "x" is the input, "h" is the output, "b" is the parameter vector, "W" means the parameter matrix.

$$I_t = \sigma(W_I[x_t; h_{t-1}] + b_I) \quad \textbf{(5)}$$

$$F_t = \sigma(W_F[x_t; h_{t-1}] + b_F) \quad \textbf{(6)}$$

$$O_t = \sigma(W_O[x_t; h_{t-1}] + b_O) \quad \textbf{(7)}$$

$$NewCell_t = \tanh(W_{NewCell}[x_t; h_{t-1}] + b_{NewCell}) \quad \textbf{(8)}$$

$$FinalCell_t = F_t \times NewCell_{t-1} + I_t \times NewCell_t \quad \textbf{(9)}$$

$$h_t = O_t \times \tanh(FinalCell_t) \quad \textbf{(10)}$$

In this experiment, 105 of hidden neurons were found as the best choice, the output size of LSTM is $(N \times 1 \times 105)$.

The fully connected layer which helps map data, the input size is $(N \times 1 \times 105)$. As a standard logistic function, sigmoid function (σ) can only result in 0 or 1 when the input is not 0. Thus, it can be used in the output gate of LSTM, to indicate the target facial expression conveniently.

To keep the balance between training and testing sets, 80% of data will be used in the training process. However, LSTM needs time-series data sequences; the input data will be in timestamp order. After several times of the experiments in the "Value Range

Tried", as Table 2 displayed. This paper chooses the parameters that make high final accuracy as "Best Value" shown.

Table. 2 Summary of parameters in Training

Hyperparameter	Value Range Tried	Best Value
Number of features(attributes)	-	118
Output Classes	-	2
Number of Hidden Neurons	15 - 200	105
Learning Rate	0.0005-0.03	0.0099
Number of Epoch	5 - 1500	30
Optimizer	Stochastic Gradient Descent (SGD), Adam	Adam
Loss Function	-	Cross-Entropy

Adam [13] as a first-order-gradient-based optimiser, it can achieve fast convergence and perform well in deep learning, thus by using it, less epoch will be needed [14]. Following Cross-Entropy [15] which combined with the SoftMax is used as the loss function to calculate the differences between the output values of the model and the actual target value; then the gradient of cross-entropy will be derived through backpropagation and fed to the Adam optimiser:

$$Loss(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_i^n \exp(x[i])}\right) \quad \textbf{(11)}$$

There are n classes. The x is the output given by the deep learning model, the class is the target output.

2.3 Model Testing and Validation

This paper adopts hold-out validation method [16], which means that validation and testing use the same data set, which is 20% of data. The reason for using hold-out is it can have more straightforward implementation than other validation approaches (e.g. 10-folder cross-validation) and avoid using the duplicated data from the training set [16]. By utilising the hold-out, the loss values of testing/validation and training after each epoch can be stored in the middle of process, thus, these loss values can draw a loss graph to display if a model is overfitting or underfitting. The beginning epoch was set to 1500 after several testing the epoch was corrected to 30.

F1-score can evaluate the accuracy of this LSTM model [17], where "tp" means all prediction and actual values are negative, "fn" means prediction is positive while actual value is negative, "fp" has opposite meaning of "fn":

$$F_{score} = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} \quad \textbf{(12)}$$

$$recall = \frac{tp}{tp+fn} \quad \textbf{(13)}$$

$$precision = \frac{tp}{tp+fp} \quad \textbf{(14)}$$

3 Results

3.1 Analysis of the LSTM model

Except using hold-out, the learning rate and optimiser are helpful in reducing time of training by using fewer epochs.

The final loss graphs of 9 grammatical facial expressions show that the training and testing loss values decrease in the same trend. Because all 9 facial expressions use the same model, the loss changes are varied among them. For "Negative" expression (Fig. 3), it always has significant overfitting after 5 epochs. For other expressions, training line is a little below the testing line at the last epoch, the overfitting is improved after correction of epoch by hold-out [16]. Since the testing data of "Negative" is the 20% continues frames of data, it means after the 5 epochs, the more training cannot benefit accuracy and the model is not good at recognising "Negative" expression. For the "Emphasis", the final accuracy showed in Table 3 is the lowest of all expressions, its loss graph (Fig. 4) has a few of fluctuates, the loss decreasing is followed by each oscillation, it may show the gradient descent of Adam optimiser can benefit to find new optimised values [13].

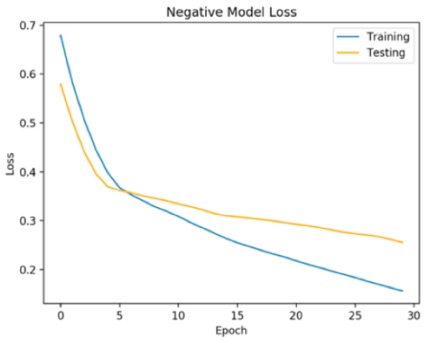


Fig. 3 Loss for "Negative" facial expression

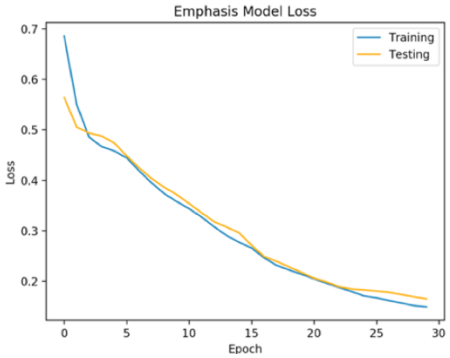


Fig. 4 Loss for "Emphasis" facial expression

The datasets may limit the advantages of LSTM on time-series data; because each frame has a label, however several frames with the same time length and the same label may be a better input to let LSTM find more patterns.

3.2 Comparison of Results

Table. 3 Summary of accuracies (F1-scores) from 2 experiments

Grammatical Facial Expressions	**Result of Freitas et al.'s Paper**	**Result of This Paper**	**Result of Previous Experiment**
Method	**MLP**	**LSTM**	**MLP (1 hidden layer)**
Affirmative	0.8773	0.9022	0.8979
Conditional	0.9534	0.8784	0.9470
Doubt Question	0.9700	0.9416	0.9411
Emphasis	-	0.8696	0.9338
Negative	0.9582	0.8816	0.8939
Relative	0.9759	0.9339	0.9634
Topic	0.9544	0.9246	0.9665
Wh Question	0.8988	0.9211	0.9320
Yes/No Question	0.9412	0.9222	0.9343
Average Accuracy	0.9412	0.9084	0.9344

As the Table 3 showed, there are 2 columns of accuracies. "Accuracy A" has the accuracies from the Freitas et al.'s most recent work [7], "Accuracy B" shows the accuracies produced by LSTM of this paper. "Affirmative" and "Wh Question" have the both highest accuracies compared to values of "Accuracy A". The other expressions cannot have better recognition while using the LSTM of this paper. The average accuracy of applying LSTM is 4% lower than MLP's. In both MLP [7] and LSTM model, the final accuracies of "Doubt Question" are very high among the 9 facial expressions. The results may mean this expression has significant changes of landmark position during the recording by comparing to other neutral frames while training the model. The LSTM has less accuracy than 1-hidden layer neural network, the reason may be that the data set is not enough to train each class of expression [18].

Signer needs to move the head from up to down several times to perform "Affirmative". To perform "Wh Question", signer's forehead needs to fold. Since these two expressions need a sequence of movement to be identified, the higher accuracies in LSTM than in MLP model [7] are reasonable.

The accuracies of “Conditional”, “Emphasis” and “Negative” are below the average accuracy. In the MLP mode [7], the accuracies of “Conditional” and “Negative” are similar and high. It may show the LSTM model hardly recognises these two expressions; memorising a sequence of these types of instances can reduce the model accuracy. Although the “Negative” also involves moving head, it has more changes of other facial landmarks than “Affirmative” [7], handling more changes of landmarks may cause low accuracy.

4 Conclusion and Future Work

This paper demonstrates the ability of this LSTM model to recognise the 9 grammatical facial expressions. By comparing past work [7], It achieves higher accuracy on “Affirmative” and “Wh Question” dataset; it is the first paper used LSTM method on this UCI facial repository [1]. Although the basic LSTM model is less accurate, the final accuracy is enough to identify the category of each testing facial expression. This paper also proves the importance of data pre-processing. Especially when the data are at different scales, Z-score can normalise them to the same scale. By extracting the crucial features (angles and distance), it reduced the working load of training; Original features are 300, but this paper only uses 118 features to get 90.84% average accuracy.

In the next stage, to fit RNN model, the dataset should be reconstructed and combined several frames with the same label. Since the basic LSTM cannot help increase accuracy on all of type of facial expressions, the next work can try other extensions of RNN.

References

[1] F. Freitas, S. Peres, C. Lima, and F. Barbosa. (2014). *UCI Machine Learning Repository: Grammatical Facial Expressions Data Set*. Available: https://archive.ics.uci.edu/ml/datasets/Grammatical+Facial+Expressions.

[2] D. Walawalkar, "Grammatical facial expression recognition using customized deep neural network architecture," *ArXiv e-prints*, vol. 1711, Accessed on: November 1, 2017Available: http://adsabs.harvard.edu/abs/2017arXiv171106303W

[3] F. d. A. Freitas, S. M. Peres, C. A. d. M. Lima, and F. V. Barbosa, *Grammatical Facial Expressions Recognition with Machine Learning* (2014). 2014.

[4] Microsoft. (2018). *Face Tracking*. Available: https://msdn.microsoft.com/en-us/library/jj130970.aspx

[5] A. Rius, I. Ruisánchez, M. P. Callao, and F. X. Rius, "Reliability of analytical systems: use of control charts, time series models and recurrent neural networks (RNN)," *Chemometrics and Intelligent Laboratory Systems,* vol. 40, no. 1, pp. 1-18, 1998/05/01/ 1998.

[6] S. Hochreiter and J. Schmidhuber, *Long Short-term Memory*. 1997, pp. 1735-80.

[7] F. Freitas, S. Peres, C. Lima, and F. Barbosa, *Grammatical facial expression recognition in sign language discourse: a study at the syntax level*. 2017.

[8] M. S. Bhuvan, D. V. Rao, S. Jain, T. S. Ashwin, R. M. R. Guddetti, and S. P. Kulgod, "Detection and analysis model for grammatical facial expressions in sign language," in *2016 IEEE Region 10 Symposium (TENSYMP)*, 2016, pp. 155-160.

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

[10] B. Hasani and M. H. Mahoor, "Facial Expression Recognition Using Enhanced Deep 3D Convolutional Neural Networks," *ArXiv e-prints,* Accessed on: May 01, 2017Available: https://ui.adsabs.harvard.edu/#abs/2017arXiv170507871H

[11] A. Graves, J. Schmidhuber, C. Mayer, M. Wimmer, and B. Radig, "Facial Expression Recognition with Recurrent Neural Networks," 2008.

[12] S. Evan and S. Michael, "Implementation of facial recognition with Microsoft Kinect v2 sensor for patient verification," *Medical Physics,* vol. 44, no. 6, pp. 2391-2399, 2017.

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

[14] A. Sinha, M. Sarkar, A. Mukherjee, and B. Krishnamurthy, "Introspection: Accelerating Neural Network Training By Learning Weight Evolution," *ArXiv e-prints,* Accessed on: April 01, 2017Available: https://ui.adsabs.harvard.edu/#abs/2017arXiv170404959S

[15] Pytorch.org, "torch.nn — PyTorch master documentation."

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

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

[18] M. Wöllmer, A. Metallinou, F. Eyben, B. Schuller, and S. Narayanan, "Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional lstm modeling," in *Proc. INTERSPEECH 2010, Makuhari, Japan*, 2010, pp. 2362-2365.