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

Yangyang Xu1

1 The Australian National University Acton ACT 2601

u6325688@anu.edu.au

**Abstract.** The abstract should summarize the contents of the paper and should contain at least 70 and at most 150 words. It should be set in 9-point font size and should be inset 1.0 cm from the right and left margins. There should be two blank (10-point) lines before and after the abstract. This document is in the required format.

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

1 Introduction

Grammatical Facial Expression (GFE), a category sign language for people with impaired hearing [1, 2]. Combination of the hardware and software solutions can make machines learn the semantic meaning of such sign language. For the hardware part, Kinect, as known as a motion sensor device produced by the Microsoft, it can capture the facial expressions [3]. 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 [3]. The bio-inspired deep learning algorithms can be implied to these landmarks and recognize some patterns among them.

To classify a facial expression from a time-series-based recording, deep learning can help with. In deep learning field, because Recurrent Neural Network (RNN) can memories data during a sequence of time and be implemented as a multilayer feedforward network, it is a suitable architecture to train time-series based dataset [4]. 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 [5].

In this paper, the dataset used in this experiment, the previous research which used the same facial landmark dataset or used the LSTM on facial expression, the training, testing and validation methods with LSTM architecture, the experiment results and further conclusion of the experiment will be explained in following sections.

2 Data

This data repository used in the following experiment is provided by the UCI website. In this repository [6], 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 listed 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 time period. For the quantity, averagely, there are 2500 instances can be used in training process for each expression, the 300 attributes also make the deep learning get sufficient exploration on features.

3 Related Work

There are few recent papers used the same data repository as this paper. Multiple Layer Perceptron architecture is firstly implemented on this repository by Freitas et al. [2], they did not get very high F-score on “Negative” and “Relative” expressions, most of expressions can be recognized above 0.75 (F-score). In 2016, by extracting the important facial points, Bhuvan et al [7] 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 [1]; their model has very good performance on each facial expression, all of expressions can have over 0.94 of F-score.

There is no paper found that used the LSTM/RNN to recognize the facial expressions of this repository, but there are two papers found used LSTM to train the similar facial landmarks. In Behzed and Mohammad’s research[8], they used the CNN to extract landmarks from videos, then they used LSTM to memories and train the landmarks; their final model can successfully recognize variety facial expressions from 4 data repositories. Alex et al [9] used unidirectional LSTM for 116 facial landmarks, the final expression recognition mean error rate is 18.2 ± 0.6%.

4 Methods

4.1 Data Preprocessing

To help classifier find the patterns, each instance is reconstructed by 11 distances, 7 angles and 100 (whole) of z-coordinators, therefore, an input entry vector has 118 attributes. The distance and angles are specified in Freitas et al’s paper [2]. 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 normalized by Z-score method [1].

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 and . Thus, the Euclidean distance is calculated by [10]:

**(1)**

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

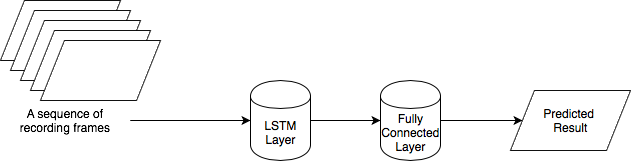
**(2)**

The angle (in cosine) between two distances ( and ) can be calculated by following function, where gets dot product, and are the norms of and respectively:

**(3)**

4.2 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. 1 displayed, there are only one hidden layers, LSTM layer. Fully Connected (FC) layer is used as an output layer.



**Fig. 1** Training Model Architecture

For the input layer, 5 frames of facial expression with 118 attributes will be sent to LSTM layer, since the LSTM has 10 hidden neurons, the input size is .

In the LSTM layer, there are 3 gates and 2 memory cells used for this gradient-based approach of LSTM, which are input gate (), forget gate (), output gate (), new memory cell () and final memory cell () [5]; In the following formulas, the “t” means a timestamp, “” is the sigmoid function, “” is the input, “h” is the output, “b” is the parameter vector, “W” means the parameter matrix [8].

(4)

(5)

(6)

(7)

(8)

(9)

In this experiment, there is () of weight for the input dimensions, () of weight for the output dimensions and (10) for parameter vector.

The fully connected layer which helps map data, the input size is (). As a standard logistic function, sigmoid function can only result in 0 or 1 when the input is not 0. Thus, it can be used to indicate the target facial expression quickly.

After several times of the training, this paper used 100 epochs, 0.01 as initial learning rate and mean square error function for cross entropy. In order to reduce overtraining, 80% random data will be used in the training process. Adam [11] as a first-order-gradient-based optimizer, it can decrease computation time, memory space and be wildly used in many deep learning model.

4.3 Model Testing and Validation

The accuracy is calculated by F-score [12], 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”:

.

4.3 Model Validation

Please check that the lines in line drawings are not interrupted and have a constant width. Grids and details within the figures must be clearly legible and may not be written one on top of the other. Line drawings should have a resolution of at least 800 dpi (preferably 1200 dpi). The lettering in figures should have a height of 2 mm (10-point type). Figures should be numbered and should have a caption which should always be positioned *under* the figures, in contrast to the caption belonging to a table, which should always appear *above* the table. Please center the captions between the margins and set them in 9-point type (Fig. 1 shows an example). The distance between text and figure should be about 8 mm, the distance between figure and caption about 6 mm.

To ensure that the reproduction of your illustrations is of a reasonable quality, we advise against the use of shading. The contrast should be as pronounced as possible.

If screenshots are necessary, please make sure that you are happy with the print quality before you send the files.

5 Results and Discussion

The printing area is 122 mm × 193 mm. The text should be justified to occupy the full line width, so that the right margin is not ragged, with words hyphenated as appropriate. Please fill pages so that the length of the text is no less than 180 mm, if possible.

Use 10-point type for the name(s) of the author(s) and 9-point type for the address(es) and the abstract. For the main text, please use 10-point type and single-line spacing. We recommend the use of Computer Modern Roman or Times. Italic type may be used to emphasize words in running text. Bold type and underlining should be avoided.

6 Conclusion and Future Work

The printing area is 122 mm × 193 mm. The text should be justified to occupy the full line width, so that the right margin is not ragged, with words hyphenated as appropriate. Please fill pages so that the length of the text is no less than 180 mm, if possible.

Use 10-point type for the name(s) of the author(s) and 9-point type for the address(es) and the abstract. For the main text, please use 10-point type and single-line spacing. We recommend the use of Computer Modern Roman or Times. Italic type may be used to emphasize words in running text. Bold type and underlining should be avoided.

References

[1] 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>

[2] 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.

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

[4] 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.

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

[6] 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>.

[7] 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.

[8] 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>

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

[10] 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.

[11] 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>

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

Appendix: Specified Distances and Angles for Data Pre-processing

