Pattern Classification using LSTM

Tanmay Negi,
Australian National University,
Acton, ACT, 2601

Abstract: In past few years neural networks have achieved a great success in extracting new features and information from data. Even though new research and studies are constantly being added to this field, neural networks still suffer from low accuracy when it's comes to pattern recognition. There are networks that can produce new patterns when fed some sequence but still they struggle to distinguish two sequence not in the order of their occurrence but in the order of their variations. Here we will use the technique as proposed by A. Ullah et.al [5] in video sequencing for pattern classification on time-series data.

Keywords: pattern recognition, sequencing, features

1. Introduction:

In this paper we will apply Long-Term-Short-Memory (LSTM) model for binary classification. LSTM's are a type of neural network capable of learning order dependencies from a sequential data. A typical LSTM network is comprised of different memory blocks called cells, in contrast to feed-forward-neural networks (FFN), there are two more states that are being transferred to the next cell, the hidden states and the cell state.

This technique was tested on anger dataset [1]. The datasets consist of the variations in pupil diameter of the participants captured at fixed timed-stamps through the duration when a small clip was shown to them. Each video clip consists of some actors/individuals depicting anger emotions, and each clip is labelled as 'Genuine" or "Posed" describing the true nature of expression displayed in the clip.
Lu Chen et.al [1] has shown that using traditional neural networks approach it is possible to for a model trained just on individual's retina scan as mentioned above to get a high classification accuracy while classifying the true nature of emotion portrayed in the clip. The performance of our resulting LSTM model was then tested and compared with the similar standard FFN as tested by Lu Chen, though a small thing to notice that **Kogan [2]** has shown that a FFN trained for categorization such as Lu Chen's, directly using the value of output node as a confidence score is not an optimal strategy, thus we would need to train the network for our control experiment using the probabilities of class membership.

2.0 Method

The experiment was setup in two phases. First, we designed a control setup, a network trained explicitly using probabilities of class membership as suggested by Kogan, which was

then compared with our LSTM model that was designed to exploit and learn pattern of variations in individual's retinal diameter for binary classification of videos.
Though we will measure the individual accuracy of respective model, the main aim of this paper is to analyse the ability LSTM to learn patterns in the data, which on contrast to traditional procedures of predicting next elements of sequence, instead will be used for classification.

2.1 Pre-processing:

The data consists total 20 short videos of varying length, each video is shown to N participants, and we have pupil diameters for left and right eye recorded at unit time-frame while the video is being played. Each video also has a label as "genuine" or "posed" as discussed above, which is encoded as 1 and 0 respectively (the two classes).
As individual's pupil was scanned using a eye-tracker, some of the data-points read zero value, caused due to eye blink, these values were replaced with the mean value of the individual's pupil diameter during the sequence of corresponding video. Then we calculated the mean of left and right eye. Then scaled the data using standard Scaler method **[3].**

2.2 Control model (FFN trained on class probability):

As we mentioned earlier our input features are of varying length (as videos are of varying length), thus to accommodate data for FFN we will just use initial 10 points from the sequence as input features for corresponding video-participant pair.
We designed a simple fully connected feed forward neural network with 10 input neurons corresponding to 10 input features, 5 hidden neurons with sigmoid activation (as it a binary classification task) and two output neurons corresponding to probabilities of two class.
Other hyperparameters include epoch: 600, learning rate: 2e-5 , criterion : CrossEntropyLoss , optimizer : optim.Adam().

To ensure the balance we shuffle the data randomly and then divided between training 60% and testing 40%. Following is the distribution of classes between training and testing.

	Train-set	Test-set
Class zero	51%	49%
Class one	47%	53%

Fig 1

We vary the threshold ϴ from 0.4 to 0.7. The results of train and tests are shown below. All the readings were taken around 90% accuracy.

ϴ	Correct	False +ve	False -ve	Average Accuracy
0.4	104	11	8	93%

0.5	**113**	**10**	**10**	**94%**
0.6	104	10	19	96%
0.7	**104**	**2**	**27**	**93%**

Fig2: Train Phase

ϴ	Correct	False +ve	False -ve	Average Accuracy
0.4	74	7	8	88%
0.5	**66**	**11**	**12**	**92%**
0.6	63	8	18	91%
0.7	**60**	**3**	**26**	**88%**

Fig3: Test Phase

However, Kogan [2] displayed in their work and as demonstrated by L.k. Milne [6], threshold of 0.5 is not sacrosanct in distinguishing between two classes, but in our experiment, we found that at threshold of 0.5 model is quite balanced with differentiating both classes with same error rate, also accuracies were similar across training and testing. This also complies with Lu Chen's et.al work on same anger dataset where they received 95% accuracy.
Thus, we accept the probability model of threshold 0.5 as control model for our experiment with LSTM.

2.3 Pattern classification model:

Our model consists of a single LSTM cell for pattern sequencing followed by a linear layer of two neurons for binary classification.
LSTM layers consists memory blocks which contain memory cells with self-connections storing the temporal state of the network in addition to special multiplicative units called gates to control the flow of information.
As indicated in their work [**4**], Google researchers displayed that increasing the number of these layers have little effect in model's accuracy, but high effect on network's complexity, making training harder. Thus, we kept our LSTM cell single layered, with 2 hidden states.
Other hyperparameters include learning rate: 0.01, criterion: cross-entropy-loss, optimizer: optim.Adam(). For training and testing similar train and test split was used for dividing data

In contrast to our FFN. LSTM can take varying sequences thus we didn't had to trim our initial features as before, but as features were of varying length it was not very optimal to train model in batches thus for each sequence (sequence of pupil diameters corresponding to video), loss backpropagation and optimizer step was implemented separately which increased the training time by large amount.
The average training-testing accuracy of model was around 80%, which was quite lower than our control experiment and Le Chen's demonstration on similar data, but it was quite higher than random prediction of 50%.
However various inconsistencies were observed while model training as displayed in following figures.

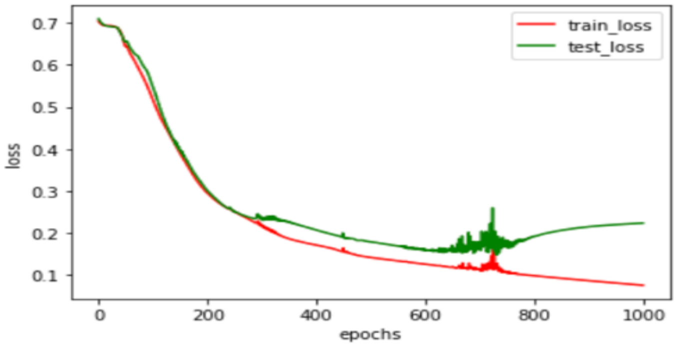

Fig4 iteration 1

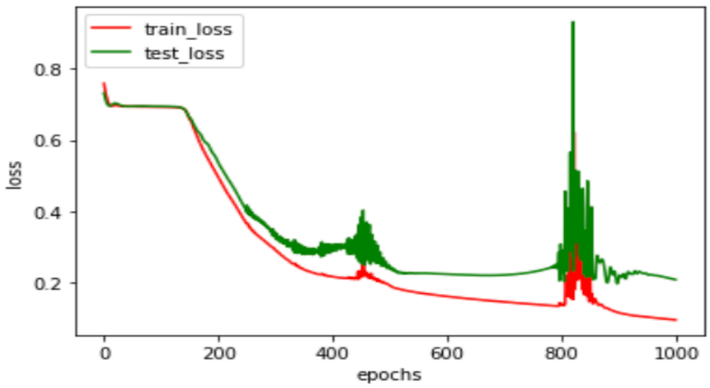

Fig5 iteration 2

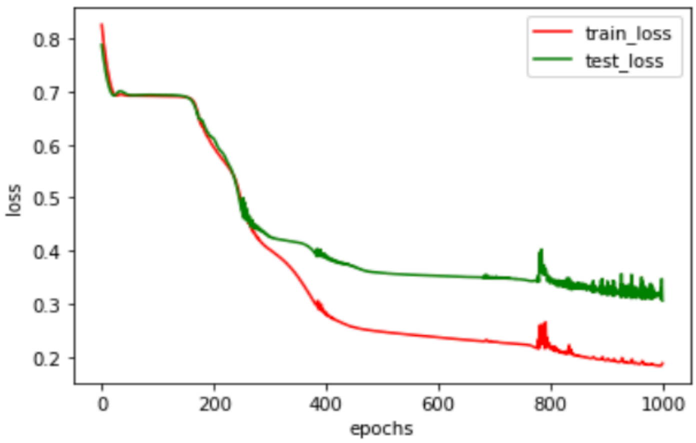

Fig6 iteration 3

As we can see initially both loss graph seems to follow normal downwards behaviour, but they suddenly jump around 800^{th} epoch as seen in all three cases, also there was high significant difference between training and testing loss.

3. Conclusion:

In this experiment we tried to use LSTM for binary classification. Though our control model out perform LSTM model, it is to be noticed that we used reduced features in our control model. Also, our LSTM in spite of displaying some subtle inconsistencies gave an average accuracy of around 80%.

In summary, the result generated implies that LSTM in-contrast to generating new patterns from given sequence, can also learn to differentiate between sequences. The implication is compelling and has a lot of real-world applications, however more research needs to be done.

4. Future work:

In this paper we discussed the implications of using LSTM for distinguishing different sequences of varying length using a simple one layered LSTM model. More research can be done on evaluating this model with more deep layers and hidden neurons which increase the LSTM's performance in deducing more complex features as demonstrated [4].

Furthermore as demonstrated by K**ogan [3],** an as implemented in our control model, we can include the concept of classification loss calculation based on thresholding which might further improve the model's accuracy.

5. References

[1] Chen, L., Gedeon, T., Hossain, M. Z., & Caldwell, S. (2017, November). Are you really angry? detecting emotion veracity as a proposed tool for interaction. In Proceedings of the 29th Australian Conference on Computer-Human Interaction(pp.412-416).ACM.

[2] Kogan, "Neural networks trained for classification cannot be used for scoring," IEEE Trans. on Neural Networks, 1991.

[3] T. Jayalakshmi, A. Santhakumaran, "Statistical Normalization and Back Propagation for Classification," *International Journal of Computer Theory and Engineering* vol. 3, no. 1, pp. 89-93, 2011.

[4] Sak, H., Senior, A. and Google, B. (n.d.). *Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modelling*.

[5] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad and S. W. Baik, "Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features," in IEEE Access, vol. 6, pp. 1155-1166, 2018, doi: 10.1109/ACCESS.2017.2778011.

[6] L.K. Milne, T.D. Gedeon and A.K. Skidmore, "Classifying Dry Sclerophyll Forest from augmented satellite data"