Music Genre Classification using LSTM and CNN

Qihang Wang,

Research School of Computer Science,
Australian National University
u6821235@anu.edu.au

Abstract. LSTM and CNN are proved to be effective in many sequential data modelling tasks. In this paper, we build deep neural networks with LSTM and CNN to do music genre classification according to the EGG signals generated by human brain activity when listening to music. To explore if there is any relationship between human emotions, brain activities and the sounds they hear. As a result, we find there are long dependencies in the EGG time series data thus the LSTM models perform better in the music genre classification task.

Keywords: LSTM, CNN, EGG, brain activity, sequential data, long dependency, music genre, accuracy.

1 Introduction

Recurrent neural networks (RNNs) like long short-term memory (LSTM) units can effectively model the sequential data [8] and LSTM is more robust when facing with vanishing gradient problem [2]. The convolutional neural networks (CNNs) with one layer of convolution built on top of word2vec [5] have shown good performance when dealing with sentence classification and sentiment analysis problems. What is more, the convolutional networks can be more accurate when getting substantially deeper and built with skip connections [4]. The two network architectures are proved to be effective in natural language processing and natural language modelling. The LSTM has the ability to keep short-term memory for a long time [2] and the CNN can use convolution layers to catch close relationships inside the sequential data.

The dataset is a time series data about music effect from Rahman, J.S. et al. [6]. It records the brain wave of the 24 participants (13 male and 11 female) when listening to three different kinds of music: classical, pop and instrumental. In this paper, we are going to use the EGG signals collect from the F7 channel of the 14 channels which located in the frontal lobe of human brain. The EGG signals are the physiological signal that used to describe brain activities and very effective in detecting the human emotional states [6].

The task of the paper is to build deep neural networks to predict the class of music by looking at the EGG signals from the F7 part of the human brain. By doing this task, we can not only tell that different kinds of music can affect the brain wave in different ways, but also prove the sound we hear can change our emotions. This will bring great value in psychology and medicine, such as the treatment of depression and insomnia because we can use sounds or music to influence the mood of people and make them feel relaxed or excited.

In the present work, three models are trained to finish the music genre classification task: a model with a single LSTM layer, a model with three LSTM layers and a CNN model. And we found that the dependencies inside the EGG signals are complicated and across long distances. The CNN models suffer more from the problem compared to multi-layer LSTM models.

2 Method

First, we import and observe the EGG signals in the data files and then label them as 0 = classical, 1 = instrumental, 2 = pop. After labelling, the dataset will be split into training, testing and validation sets. These sets will be fixed and not change during the rest of the experiment.

Then, we build the LSTM and CNN models and train them multiple times with the fixed training set. And the hyperparameters will be selected base on the models' performance on the validation set. The baseline will be set according to the performance of the single LSTM layer model. And the performance of models will be measured by their average accuracy (average after removing the lowest and highest score) on the fixed testing set. The detailed design of the three models will be given when introducing them individually.

In the end, we compare the performance of the CNN model and multi-layer LSTM model with the single LSTM layer baseline model and try to explain the types of dependencies inside the EGG time-sequential data. And give recommendations on which kind of model should be used to address the music genre classification problem.

2.1 Data Pre-processing

The EGG time series data we are using is separated into 12 excel files and the EGG signals inside files 1 to 4 are recordings of participants listen to classical music, 5 to 8 are instrumental music and 9 to 12 are pop music. From each class of music, we pick two EGG signals and plot them out according to time. From fig.1, it is hard to tell which signal belongs to which kind of music, we can only make an assumption that when listening to pop music, the brain waves of people will not change so dramatically compared to instrumental and classical music. Also, the length of the EGG signals differs from each other, mainly because the length of music varies from each other. The longest EGG signals have 2197 inputs while the shortest ones have 1608 inputs. This could be a problem when doing batch training with LSTMs as all the input patterns should have equal length when put into a tensor. So, we need to define a padding strategy when applying random sampling during the training process. Also, CNN needs the size of its input to be fixed, thus we need to cut the EGG signals into slices of fixed length before feeding them into the CNN model.

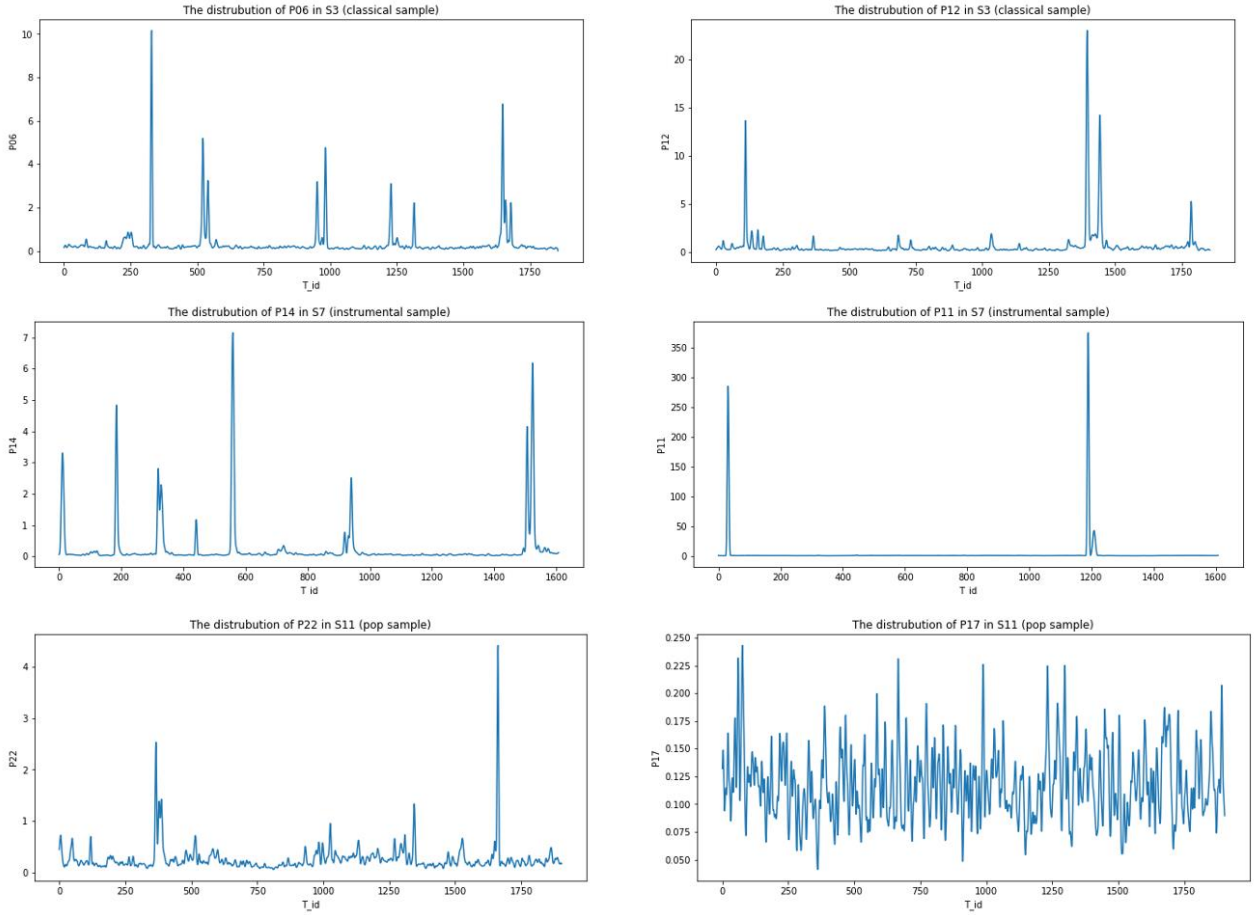

Fig. 1. The sampled signals from the three music genres. The y axis is the EGG signal value while the x axis is the time.

The EGG time series data is split into training, testing and validation sets. 20% of the total data is kept for testing and 20% of the left 80% of data is saved for the validation set. The rest of the data (64% of the data) will be kept for training. We cut down the number of patterns in the validation set for we want to save more patterns for training since deep neural networks are hard to train and tend to rely on a big amount of training data to achieve a good performance.

In the EGG time series data, for every time frame, we have 8 inputs. This means when cutting the data into fixed-size slices for the CNN models, we need to make sure the length of each slice and the gap between each slice can be divided by 8, in order to keep the information in each time frame complete. Also, when padding the EGG signals, we memorize the actual length of each signal, so we can capture the output of the original end of each input signal from the LSTM layers and further reduce the influence of the vanishing gradient problem in LSTM models.

2.2 Single LSTM Layer Baseline Model

The fig.2 gives the architecture of the baseline single LSTM layer model, the input data first go through an LSTM layer, summarize the information of the current EGG signal in a vector with 32 dimensions. The LSTM layer is followed by three linear layers, which transfer the dimension of the output vector from 32 to 32, 32 to 16 and 16 to 32. After each

linear layer, ReLU is applied as the activation function and we use a dropout rate of 10% in the LSTM and linear layers. Finally, the output layer is a linear layer that transfers the dimension of the output vector from 32 to 3.

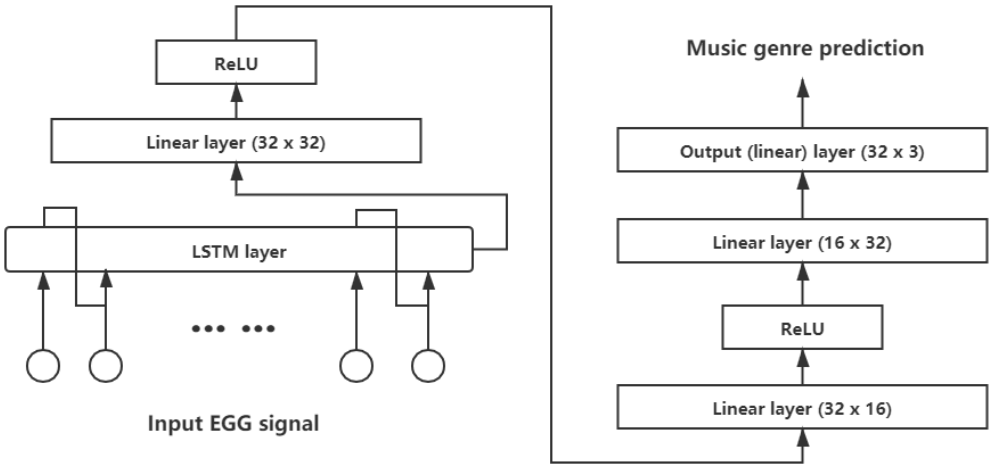

Fig. 2. The architecture of the single LSTM layer baseline model.

As to the training strategy, we train this model on GPU for 300 epochs with a batch size of 16. We choose Adam with a learning rate of 0.001 as the optimizer and use the cross-entropy loss function to calculate the loss of the baseline model. Finally, we test the baseline model on the test set split before.

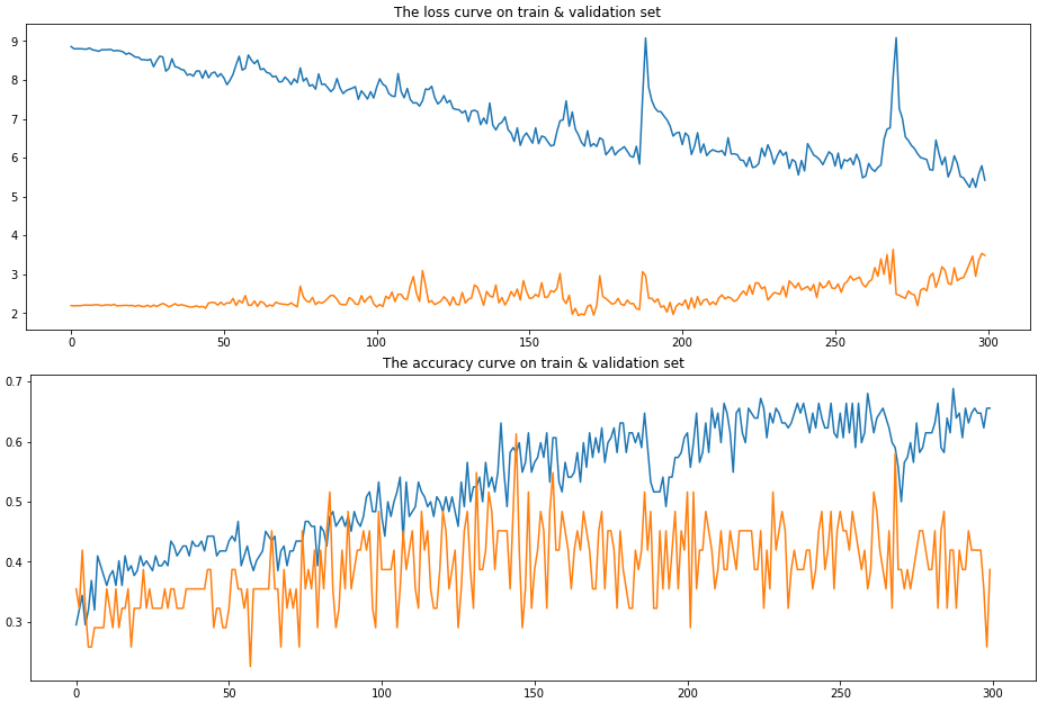

Fig. 3. The training accuracy and loss curve of the baseline model on the training and validation set.

The reasons for choosing these hyperparameters is illustrated in fig.3. After 200 epochs of training, the loss of the baseline model on the training set starts to oscillate and slow its speed of going down. Meanwhile, the loss on the validation set starts to rise faster than before. Also, look at the accuracy in fig.3, the accuracy of the baseline model on the training set stays at around 70% and not goes up. And the accuracy on the validation set stays at 45%. So, it can be inferred that the baseline model fully learns the training set with these hyperparameter settings at 200 to 300 epochs.

2.3 Multi-layer LSTM Model

The fig.4 shows the architecture of the multi-layer LSTM model. We use three LSTM layers to extract a representation vector of 64 dimensions from the EGG signals this time, as it is reported that RNNs perform best when having 3 or 4

layers without residual connections as an encoder of the input sequential data [1]. Similar to the baseline model, the representation vector will go through three linear layers which transfer its dimension from 64 to 64, 64 to 32, 32 to 64, each output of the linear layers will be activated by ReLU function and each LSTM and the linear layer will be trained with a dropout rate of 10%. Again, the final output layer maps the output vector to 3 dimensions with a linear layer.

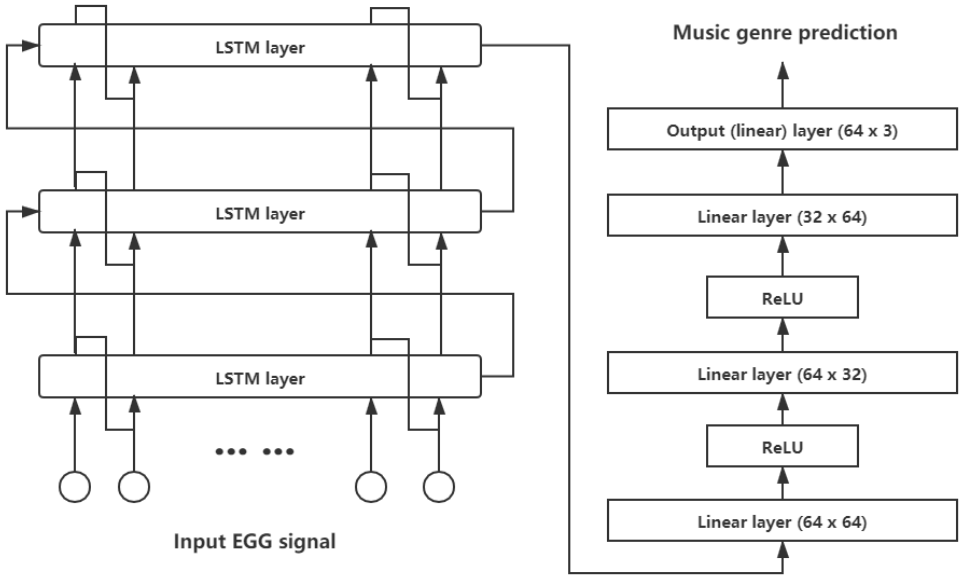

Fig. 4. The architecture of the multi-layer LSTM model.

We train the model with GPU for 600 epochs with a batch size of 16. The optimizer is Adam with a learning rate of 0.001. Again, we use the cross-entropy loss as our loss function. During testing, we test two weights of the multi-layer LSTM model, one is the model weights we get after the whole 600 epochs of training, the other one is the multi-layer LSTM model which has the highest accuracy on the validation set during training. With a higher accuracy on the validation set, we can expect the multi-layer LSTM model has a better generalization ability thus have a better performance on the testing set.

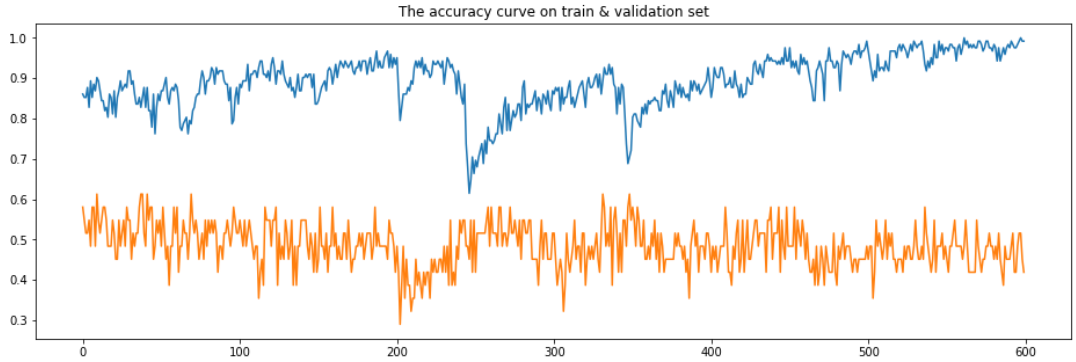

Fig. 5. The training accuracy of the multi-layer LSTM model on the training and validation set.

As to the hyperparameter tuning, we notice that during the 600 epochs of training, the multi-layer LSTM model keep improving its performance on the training set, and can achieve an accuracy above 95% on the training set after 600 epochs. Also, the accuracy on the validation set stays stable at around 50% as shown in fig.5.

2.4 CNN Model

The CNN needs its input size to be fixed. However, the sequential data tend to have different length further process on data is required. We cut the EGG signals into slices with a fixed length of 960 and set the overlap between slices to be 50% (the gap between the heads of neighbour slices is 480). The label of the slice is the same as its source signal.

Because each EGG signal will generate multiple slices and the whole input can just be assigned with one final prediction. We apply the max vote strategy: each slice will be feed into the CNN model independently and receive a prediction as a vote. The music genre that receives the most votes win the competition and the class will be the final prediction of the input EGG signal.

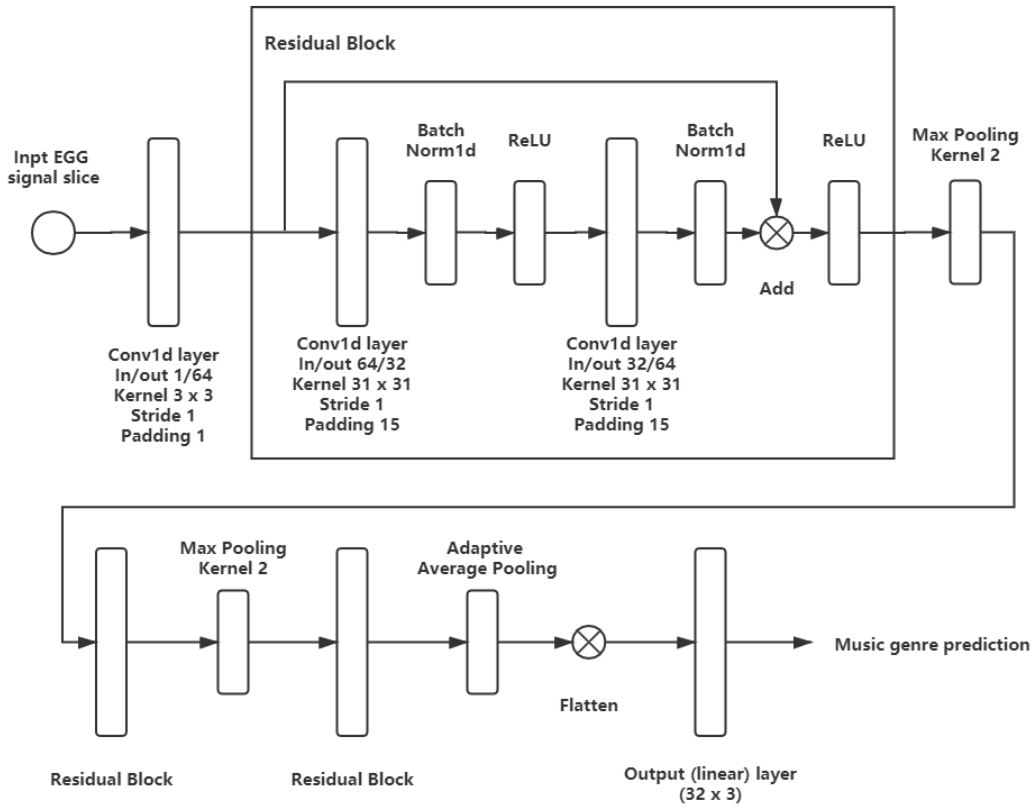

Fig. 6. The architecture of the CNN model.

The architecture of the CNN model is shown in fig.6. The current slice of the input EGG signal will be feed into a one-dimensional convolution layer with 32 kernels, followed by three residual blocks which do not change the channel number and dimension of the input vectors. By using the residual blocks, the features can be reused by different layers of the CNN during forward propagation and can improve the accuracy and coverage speed of the current CNN [3]. The first two residual blocks will be followed by a max pooling layer with a kernel size of 2. The output of the third residual block will be feed into an adaptive average pooling layer and then flatten into a vector of 32 features extract from the current signal slice. The output linear layer will map the vector from 32 to 3.

For each residual block, it is made up of two one dimensional convolution layers. Each convolution layer will be followed by a batch normalization layer and a ReLU activation layer. The batch normalization allows us to use a higher learning rate, reduce the internal covariance shift and act as a regularization function thus in some case can replace other regularization methods like dropout [7]. Finally, the original input of the residual block will be added into the output of the convolution layers and activated by a ReLU function. The shortcut can reduce the influence of the vanishing gradient problem in CNNs [4]. Also, it can accelerate learning and reduce the number of parameters (can use smaller convolution layers) required by the deep CNNs to achieve a reasonable performance [3].

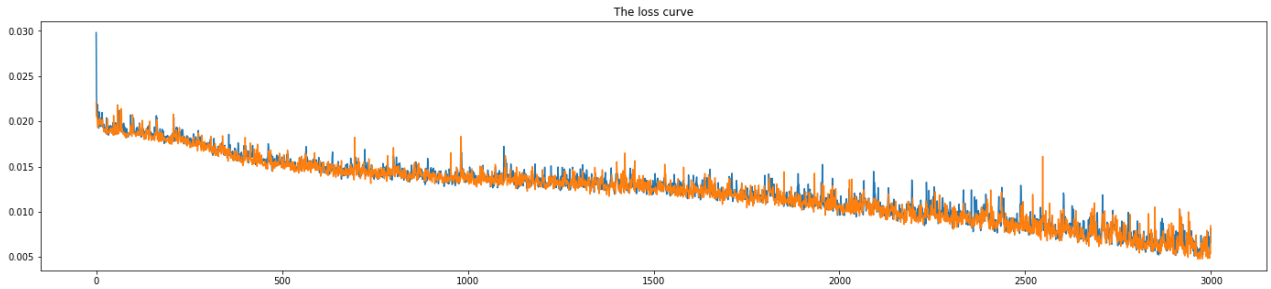

Fig. 7. The loss curve of the CNN model on the training and validation set.

In order to train the CNN model, we set the training epoch to 3000 and batch size to 64. The optimizer we use is Adam with a learning rate of 0.002. Similarly, we use the cross-entropy loss.

In the 3000 epochs, the losses on the training and validation set keep going down and the accuracy of the CNN model on the training set rise up to near 100%. Limited by the computing resources and the training time, we set the number of total epochs to be 3000.

3 Results and Discussion

We train the three models 15 times each. By comparing the accuracy of the multi-layer LSTM models and the CNN model with the baseline model on the same testing set. We determine the final performance of each model on the music genre classification task. Then analyze the characteristics of the EGG signal data and which method suits this task.

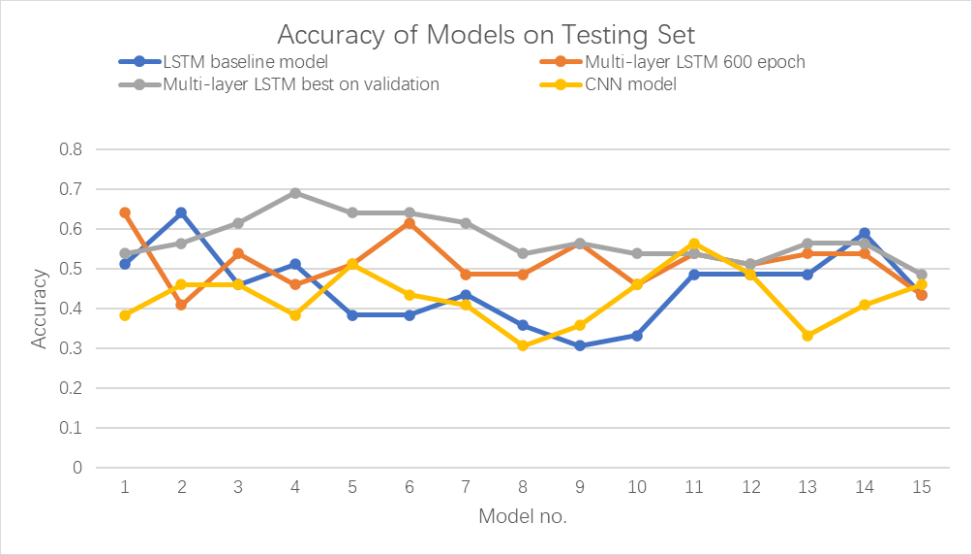

Fig. 8. The performance (accuracy) of the three models on the testing set.

3.1 Baseline Model

The accuracy of the baseline model during 15 times of training is shown in fig.8. It can be seen that its performance on the testing set varies rapidly: the highest accuracy is 64% while the lowest one is 31%. After removing the highest and lowest accuracy, the average accuracy of the baseline model is 45%. The average accuracy of the baseline model on the training set is around 70%.

The results show that adding an LSTM layer to address the music genre classification task is not sufficient. However, it can also be inferred that the LSTM can capture some dependencies in the EGG time series data. The baseline model cannot achieve good generalizability on the testing set. Its average accuracy is only 12% higher than random guessing.

3.2 Compare the Multi-layer LSTM Model with the Baseline Model

Each time we train a multi-layer LSTM model, we keep two weights: the final weights of the model which finishes the 600 epochs of the training process, the weights that achieve the highest accuracy on the validation set during training.

Shown by fig.8, the multi-layer LSTM model that completes the 600 epochs achieve 64% as its best accuracy on the testing set, while the worst to be 41%, better than the baseline model's 31%. After removing the highest and lowest ones, the average accuracy of the multi-layer LSTM model is 52%. The improvement of the multi-layer LSTM model is noticeable. And the model can easily achieve an accuracy above 95% on the training set.

As to the multi-layer LSTM model which performances best on the validation set. Its highest accuracy on the testing set is 69% and the lowest is 49%. The average accuracy of the multi-layer LSTM model is 57%. Compared with the baseline model, the improvement gained by adding LSTM layers is reasonably big.

Comparing the two multi-layer LSTM models with our baseline model, it can be seen that adding LSTM layers is useful in the music genre classification task. By stacking LSTM layers, a better representation of the EGG signals can be extracted and the accuracy of the model can be improved. Also, by selecting the model that performs the best on the validation set, we can expect our model to have better generalizability and reduce the risk of overfitting in some case.

3.3 Compare the CNN Model with the Baseline model

After 15 times of training, the highest and lowest accuracy of the CNN model on the testing set are 56% and 31% as in fig.8. The average accuracy is 43%: 2% lower than our baseline model. While the CNN model tends to get accuracy above 98% on the training set, its generalizability is bad.

By comparing the CNN with the baseline model and the multi-layer LSTM models. We suppose that CNN can model the distribution of the training data and can do some classification based on the information of the sliced signals. However, the dependencies a CNN model can capture in the sequential data is too close and this question makes the CNN model suffers from addressing the music genre classification task.

3.4 Compare the Results with the Original Dataset Paper

Compared with the results from Rahman's paper, by only using the EGG signals from the F7 part of human brain, the max accuracy we can achieve on the test set is 69%, much lower than the paper's highest results (98.6%). With NN and full EGG signal data from all 14 channels of human brain, they managed to achieve the result. They extract 25 features from 14 channels and build NN, KNN and SVM models to do the music genre classification task [6], our multi-layer LSTM model with the EGG signal of F7 channel can only match the performance of their KNN model (73.8%).

4 Conclusion and Future Work

After applying LSTM and CNN networks to deal with music genre classification task. We come to the conclusion that both CNN and LSTM have the ability to model sequential data and the LSTM perform better when dealing with the EGG time series data. Because the length of the dependencies that a CNN can capture depends on the size of kernels of its convolution layers. It can be inferred that the distance of some dependencies in the EGG times series data is too far for a CNN to capture. Thus, the multi-layer LSTM model can better describe the EGG time series data than the CNN model.

This means the dependencies between sequential inputs inside the EGG series data is more complicated than we suppose. There are long-distance dependencies and we have to pay attention to them in order to get better accuracy when predicting music genre.

As to the future work. The strategy that cut the EGG signals into slices and use them to train the CNN proved to be functional. So, we may put both the slices and the whole signal into the multi-layer LSTM model to make the model robust to incomplete inputs and enhance the performance of the model by generating more training data with the slice strategy.

Also, more works should be done in the future to explore the relationship between people's brain wave activity and emotions. So, we can have a better idea of which parts of the brain we should pay more attention to, thus focus on the EGG signals generated from test parts of the brain more.

References

1. Britz, D. *et al.* (2017) 'Massive exploration of neural machine translation architectures', *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pp. 1442–1451. doi: 10.18653/v1/d17-1151.
2. Hochreiter, S. and Schmidhuber, J. (1997) 'Long Short-Term Memory', *Neural Computation*, 9(8), pp. 1735–1780. doi: 10.1162/neco.1997.9.8.1735.
3. Huang, G. *et al.* (2017) 'Densely connected convolutional networks', *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua, pp. 2261–2269. doi: 10.1109/CVPR.2017.243.
4. He, K. *et al.* (2016) 'Deep residual learning for image recognition', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem, pp. 770–778. doi: 10.1109/CVPR.2016.90.
5. Kim, Y. (2014) 'Convolutional neural networks for sentence classification', *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1746–1751. doi: 10.3115/v1/d14-1181.
6. Rahman, J. S. *et al.* (2020) 'Brain Melody Informatics: Analysing Effects of Music on Brainwave Patterns', *Proceedings of the International Joint Conference on Neural Networks*, (October). doi: 10.1109/IJCNN48605.2020.9207392.
7. Ioffe, S. and Szegedy, C. (2015) 'Batch normalization: Accelerating deep network training by reducing internal covariate shift', *32nd International Conference on Machine Learning, ICML 2015*, 1, pp. 448–456.
8. Linzen, T., Dupoux, E. and Goldberg, Y. (2016) 'Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies', *Transactions of the Association for Computational Linguistics*, 4(1990), pp. 521–535. doi: 10.1162/tacl_a_00115.