MLP Coursework 4

G77 s1773005 s1607197

Abstract

We explore different type of neural networks, such as Convolutional Neural Networks(CNNs) and Recurrent Neural Networks(RNNs), for 10 and 25 genre recognition tasks using Million Song Dataset. We first explore different number of convolutional layers and feature maps to find the best model structure of CNNs for these tasks. Based on the result, we find that CNNs are not suitable for our inputs which are very small. Then we compare different cells and regularization for the RNNs. Based on the accuracy and speed, we choose GRU cells with 10^{-2} L2 regularization to be the best model structure of our MSD 10 and 25 Genre recognition tasks.

1. Introduction

This report will focus on advanced models of deep neural networks for music genre recognition of the Million Song Dataset. There are two type of genre classification : 10 Genre classification and 25 Genre classification. For each class in each classification, there are 5000 and 2000 labeled examples for training and validation for the two genre classification. The test dataset has 1000 examples per class in the 10 Genre classification and 400 examples per class in the 25 Genre classification. Our previous report focused on building a baseline experiment for this report by deep feedforward neural networks. The best error and accuracy performances on the validation set are displayed in $Table1$ belowed. The structure of the baseline is a 3 layers feedforward neural networks with 200 hidden units. The activation function is Tanh(Kalman & Kwasny, 1992) and optimization is Adam (Kingma & Ba, 2014) with 10^{-3} learning rate. This network was initialized with Glorot(Glorot & Bengio, 2010) initialization and it also has L2 regularization with a coefficient of 10^{-2}.

Measure	MSD 10 Genre	MSD 25 Genre
Error	2.24	2.27
Accuracy	0.44	0.45

Table 1.

In this report we will mostly focus in the improvement that using advance deep neural network: Convolutional Neural Networks(CNNs) and Recurrecnt Neural Networks(RNNs). We fisrt explore the CNNs to see weather it can fix the over-fitting problem of our baseline experiment as they contain less number of parameters than deep feedforward neural networks. Then we explore the RNNs to see weather it can improve the performance if the CNNs are not work well for our tasks as they are good at processing sequential data. To be more specific, several questions can be asked about the architecture of the Network:

1. How does different number of convolutional layers affect the performance?
2. Should we consider the density layers?
3. How does different number of feature maps affect the performance?
4. What kind of cell architecture should we use for our RNNs, LSTM or GRU cell?
5. Which regularization techniques should be used for our RNNs?

2. Methodology

The CNNs and RNNs are advanced neural networks. The CNNs have been introduced because they can learn hierarchical features, showing state-of-the-art performance in speech recognition and music segmentation. Therefore, we choose CNNs to see weather they can better capture the feature of our inputs with less parameters in order to fix the over-fitting problem. The RNNs can process sequential data pretty well. As our inputs contain 120 dimensions segment feature, it seems suitable to use the RNNs. In order to use this family of neural networks in TensorFlow, we will make use of the following methods:

- $tf.nn.conv2d$ - A method that builds the convolutional layer. It captures feature from each 1*1 pixel of inputs by using a window, named kernal, that we can define the size. It also contains a channel parameters, which means the depth of our inputs.
- $tf.nn.max\_pool$ - A method that sweeps a rectangular window over the input tensor, computing a reduction operation for each window. Here we use the maximum as the reduction operation. It also contains padding parameters, which means the stride of the sliding window for each dimension of the input tensor.
- $tf.nn.rnn\_cell.LSTMCell$ - A method that builds a Long Short Term Memory cell. LSTM Cell is a

unit used in Recurrent Neural Networks architectures which can remember important values for longer or shorter durations through gates. The gates control the information flow and let the network 'remember' or 'forget' information.

- *tf.nn.rnn_cell.GRUCell* - A method that builds a Gated Recurrent Unit, which is similar to a LSTM cell without a separate memory cell.

- *tf.contrib.rnn.static_rnn* - Creates a recurrent neural network specified by RNNCell

Furthermore, so that there is some overfitting problems during the training, we use L2 regularization to fit them and *tf.nn.l2loss* is also used.

3. Convolutional Neural Network

3.1. Number of layers

3.1.1. Motivation

In this experiment, we build a deep convolutional neural networks(CNNs) to explore the best structure of the genre recognition task. Several different experiments were performed, namely different number of convolutional layers and number of feature maps. From the baseline we can find that the result is over-fitting. Therefore, we first try shallow convolutional neural network without density layer and small number of feature maps in order to minimize the number of parameters to fix the over-fitting problem.

3.1.2. Description and Results

We use a CNNs with different number of layers. For all networks, the input is assumed to be of size 120*25 and single channel. Softmax functions are used as activation at output nodes because music genre is a multi-label classification task. In this experiment, all the convolutional and fully-connected layers are equipped with identical optimisation techniques and activation functions - Adam optimisation(Kingma & Ba, 2014) with 10^{-3} and Tanh activation function (Kalman & Kwasny, 1992). This is for a correct comparison with our baseline experiments.

In this experiment, we explored the performance of using different number of convolutional layers in MSD dataset. The network consists of 2 convolutional layers with two dimensional convolutional layers (5*5 for all) and max-pooling layers ((4*3)-(5*3)) with the same size stride. Then, we build the network consisting of 3 convolutional layers, which has one dimensional convolutional layers (5*1 for all, i.e., convolution along time-axis) and max-pooling layers ((2*1)-(3*1)-(7*1)) with the same size stride. We try one dimension convolutional layers is because the 120 dimensions of input is only different segment of the same song. The important feature is the 25 dimensions part. Therefore, we only convolution the segment dimension. The number of feature maps for the two different structure networks is 5, 10 and 5, 10, 10 respectively. They are flattened and fed into a fully-connected layer, which acts as the classifier.The results are shown in figures and tables below.

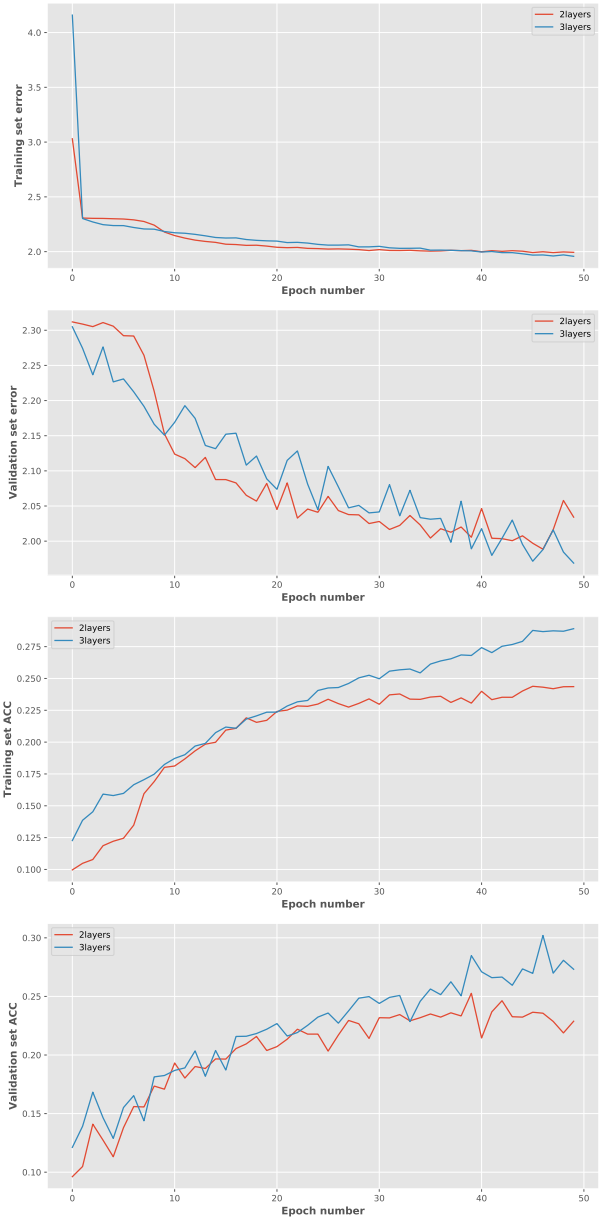

Figure 1. Comparing two and three convolutional layers in error and accuracy in the training and validation set over 50 epochs in the 10 Genre MSD Dataset.

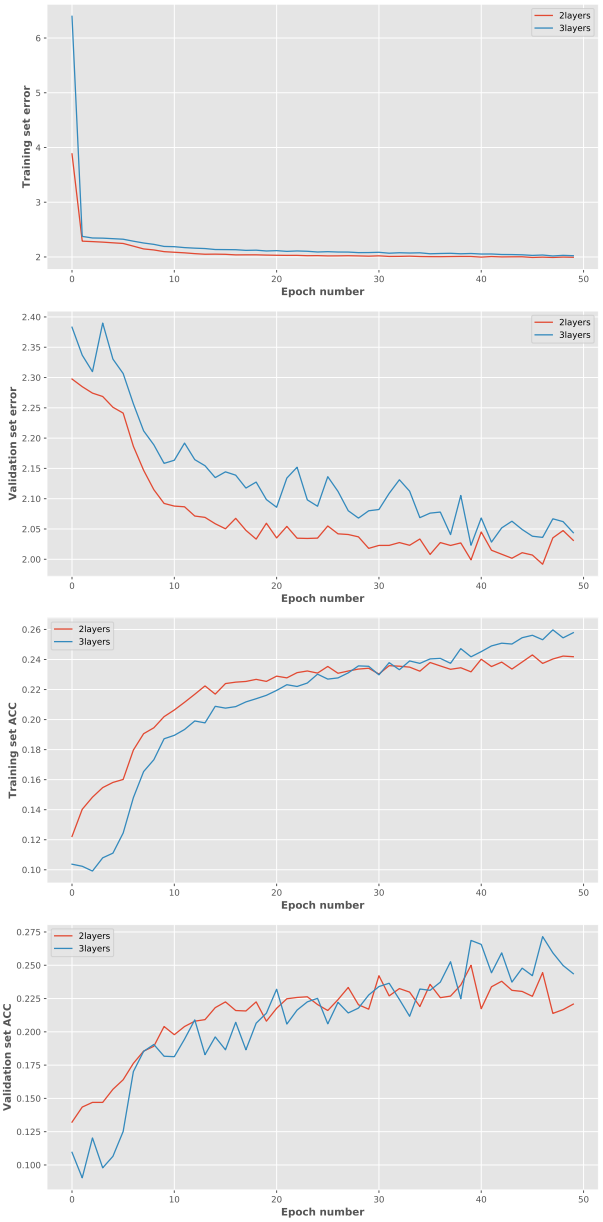

Figure 2. Comparing two and three convolutional layers in error and accuracy in the training and validation set over 50 epochs in the 25 Genre MSD Dataset.

	Measure	MSD 10 Genre	MSD 25 Genre
two	Error	2.03	2.03
	Accuracy	0.23	0.22
three	Error	1.97	2.04
	Accuracy	0.27	0.24

Table 2. Comparing two and three convolutional layers in error and accuracy in the validation set over 50 epochs in the 10 and 25 Genre MSD Dataset.

3.1.3. Discussions

We compared using 2 convolutional layers against 3 convolutional layers in the task at hand. As can be observed in the Figures and Table, they have similar performance on both 10 and 25 genre recognition tasks. Comparing to the baseline built by deep feedforward network, the CNNs has less parameters and no over-fitting, but the error are higher and accuracy are lower for both networks. Choi et al. has found that, as for music genre recognition tasks, they found that the best architecture should contain large number of feature maps to capture the features (Choi et al., 2016) and we also need density layers at the end of conv.layer to better do the classification task(Choi et al., 2017). Therefore, we try the same 3 convolutional layers with 2 density layers and also increase the number of feature maps during the next experiment.

3.2. Number of feature maps

3.2.1. Motivation

From the above, we can find that the number of parameters is too small to represent the feature of our inputs. Therefore, in this experiment, we explored the performance of using large feature maps and density layers in the MSD dataset.

3.2.2. Description and Results

The network consists of 3 convolutional layers that are followed by 2 fully-connected layers. It has the same one dimensional convolutional layers and max-pooling layers ((2*1)-(3*1)-(2*1)) alternate with the same size stride. The two density layers contain 350, 100 hidden units for samll number of feature maps and 500, 100 hidden units for large number of feature maps. The weights of this layer are initialized with Glorot initialization(Glorot & Bengio, 2010) and biases are initialized to 0. The small one has the same number of feature maps as before, and the large one has 30, 50 and 70 feature maps of three convolutional layers respectively in 10 and 25 Genre MSD Dataset. The results are shown in figures and tables below.

3.2.3. Discussions

The result shown in the Figures and Table show that the use of density layers and large feature maps improve performance in the validation set for both datasets comparing to the CNNs before. However, comparing to our baseline before, the accuracy is still lower, especially for the 25 genre recognition task. Comparing our input dimension with the researches using CNNs to do the genre recognition taks before (Choi et al., 2016)choi2017convolutional, we can find that our inputs dimension is much samller. The researches using CNNs have the 96*1366 features and ours only have 120*25 features. Also, the important features to do the genre recognition only have 25 dimensions. Therefore, we can not take advantage of CNN using less parameters to better capture the characteristic of high dimension inputs.

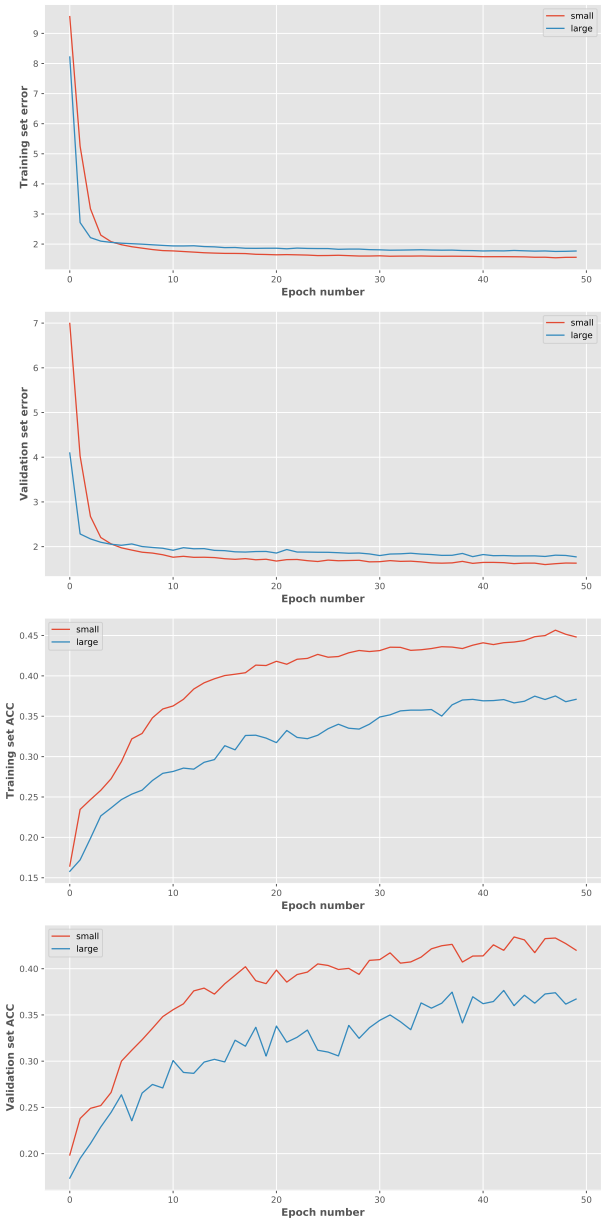

Figure 3. Comparing small and large number of feature maps performance in error and accuracy in the training and validation set over 50 epochs in the 10 Genre MSD Dataset.

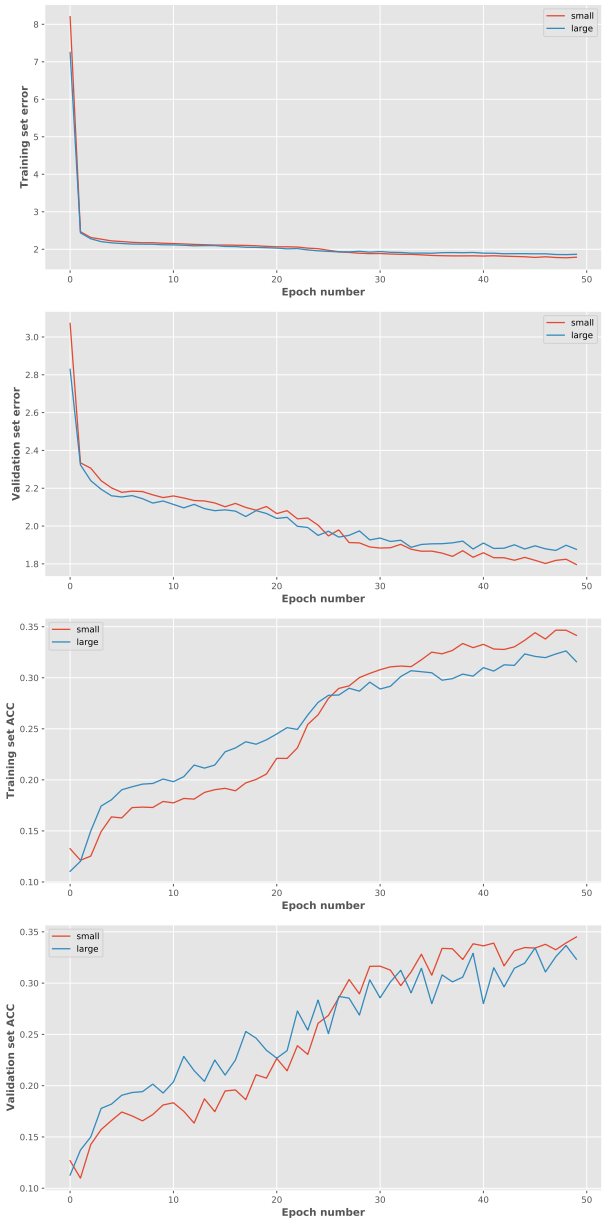

Figure 4. Comparing small and large number of feature maps performance in error and accuracy in the training and validation set over 50 epochs in the 25 Genre MSD Dataset.

	Measure	MSD 10 Genre	MSD 25 Genre
small	Error	1.63	1.80
	Accuracy	0.42	0.35
large	Error	1.77	1.88
	Accuracy	0.37	0.32

Table 3. Comparing small and large number of feature maps performance in error and accuracy in the validation set over 50 epochs in the 10 and 25 Genre MSD Dataset.

4. Recurrent Neural Networks

4.1. LSTM or GRU cell

4.1.1. Motivation

In this experiment, we try Recurrent Neural Networks. They are used to process sequential data, which are built to perform better in datasets that are sequences of values. Unlike feedforward neural networks, RNNs process sequences of inputs by using their internal memory. Therefore, it seems to be the most suitable network architecture for the MSD dataset due to the sequential nature of music. We explore which cell is better for our genre recognition task, the LSTM cell or GRU cell.

4.1.2. Description and Results

In this experiment, we use a fully connected softmax layer with 200 hidden units after the chosen RNN architecture. The weights of this layer are initialized with Glorot initialization (Glorot & Bengio, 2010) and biases are initialized to 0. Also, an Adam optimizer with 10^{-3} learning rate was used in all experiments. A single LSTM or GRU cell was used. The training and validation error and accuracy plots can be seen in Figure and Table below.

4.1.3. Discussion

From the results we can find that, both LSTM and GRU cell have over-fitting problem. The validation error is much higher than training error. However, even with over-fitting, the accuracy of both 10 and 25 Genre recognition are higher than baseline and CNNs, which means that RNNs is suitable to our inputs. Comparing the accuracy of both 10 and 25 Genre recognition, we choose GRU cell to do the further experiment exploring weather L2 regularization with different β value can fit this problem.

4.2. L2 Regularization

4.2.1. Motivation

From the previous experiment, we can find that the Recurrent Neural Networks with a single GRU cell have over-fitting problem. As to our baseline, we use L2 regularization to fix this problem. Here, we try different β value of L2 regularization to find the one that get the highest accuracy and lowest error for both 10 and 25 genre recognition.

4.2.2. Description and Results

The RNNs architecture are same as before with fully connected softmax layer with 200 hidden units. We compare 10^{-2}, 10^{-3} and 10^{-4} β values of L2 regularization. The results are shown in figures and tables below.

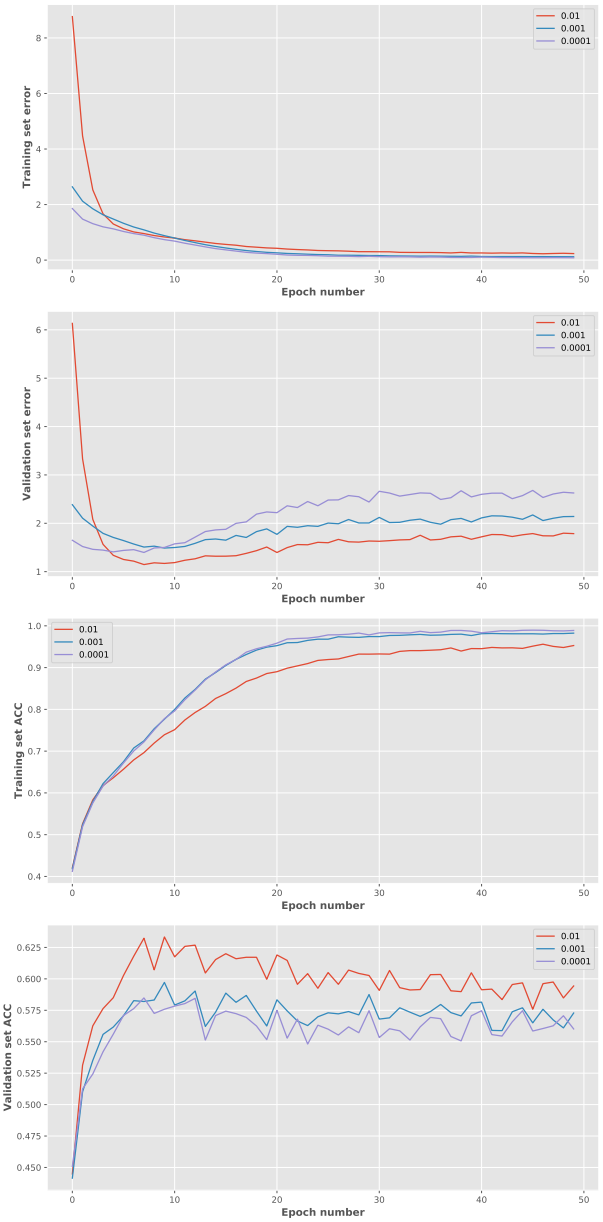

Figure 5. Comparing different β value of L2 regularization with GRU RNNs in error and accuracy in the training and validation set over 50 epochs in the 10 Genre MSD Dataset.

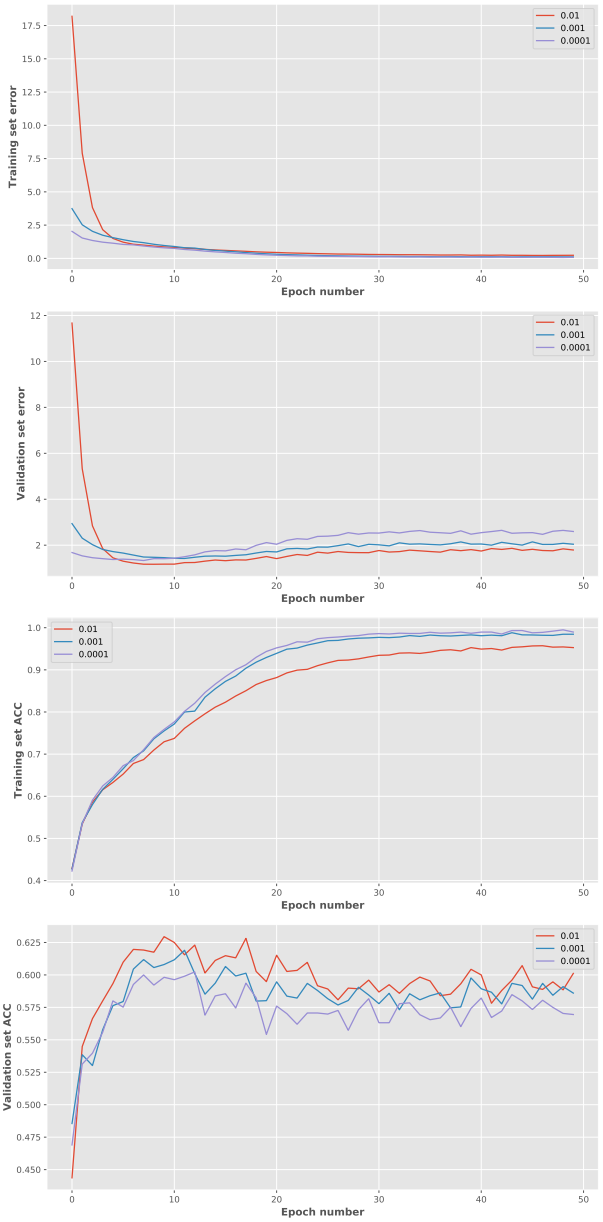

Figure 6. Comparing different β value of L2 regularization with GRU RNNs in error and accuracy in the training and validation set over 50 epochs in the 25 Genre MSD Dataset.

	Measure	MSD 10 Genre	MSD 25 Genre
10^{-2}	Error	1.79	1.79
	Accuracy	0.59	0.60
10^{-3}	Error	2.14	2.04
	Accuracy	0.57	0.59
10^{-4}	Error	2.63	2.60
	Accuracy	0.56	0.57

Table 4. Comparing different β value of L2 regularization with GRU RNNs in error and accuracy in the validation set over 50 epochs in the 10 and 25 Genre MSD Dataset.

4.2.3. Discussion

As to the baseline experiments from previous report, L2 regularization is the best one for our tasks. For both 10 and 25 genre tasks, the 10^{-2} L2 coefficient is the best β value to fix the over-fitting problem. However, from the validation accuracy, we can still find over-fitting problem. Therefore, in the future researches, we can try larger β value for this problem.

5. Test Set Results

The following results are the accuracy scores in the test set of the 10 Genre and 25 Genre datasets. We choose the single layer GRU cell with 10^{-2} coefficient value of L2 regularization and compare it with our baseline model presented in the previous report.

Accuracy	MSD 10 Genre
Baseline	0.44
$GRU - RNN(L2\_1e-2)$	0.60

Table 5. Test set accuracy for the 10 Genre MSD dataset

Accuracy	MSD 25 Genre
Baseline	0.45
$GRU - RNN(L2\_1e-2)$	0.59

Table 6. Test set accuracy for the 25 Genre MSD dataset

6. Related Work

During this part, we compare our advanced model with previous research in the MSD Genre recognition task. Sander et al built a convolutional network to perform genre recognition using chroma features and audio features as inputs and using tags(Bertin-Mahieux et al., 2010) as the targets (Dieleman et al., 2011). More important, they train these features separately, which means they use two convolutional neural networks to train these inputs and combined the results as the inputs of following multilayer perceptron. However, the result of the CCNs is not very good. They only got nearly 30 percent accuracy for the test dataset. After that, MING-JU WU et al found that if we combined the chroma features and audio features, it will increase nearly 14 percent accuracy for music genre recognition(Wu & Jang, 2015). The inputs and results are similar to us, which means that CNNs may not be suitable to these kind of data.

After that, Paulo Chiliguano captured more dimension of chroma and audio features of the Million Song Dataset, which got 130 total dimension of features and 128 dimen-

sion of segments as the inputs. The accuracy was almost 68 percent for the accuracy(Chiliguano & Fazekas, 2016). Also, Siddharth Sigtia et al found that Relu activation function is better than standard Sigmoid units when we use CNNs to do the music genre recognition as neural nets with ReLUs as hidden units can reach the same error level on the training set much faster than sigmoid nets. They found that regularization technique are useful to fix the over-fitting problem(Sigtia & Dixon, 2014). From our experiments, we use the L2 regularization and Tanh activation function for our multilayer percepton model and the results are better than CNNs. This is because the dimension of our inputs feature are only 25. As to CNNs, the number is too small to convolution. Therefore, MLP with Relu or Tanh activation function can be better at this small number of feature dimension, especially when we fix the over-fitting with L2 regularization.

Despite the researches we mention above, we should consider having large dimension of features to do the music genre recognition task. We can also combine the CNNs and RNNs to increase the accuracy (Choi et al., 2017). Choi et al used CNNs to better capture the large dimension of inputs than MLP as they have less number of parameters, and then put the results as the inputs of RNNs model to capture the sequence information of the music. They got nearly 85 percent accuracy of the test dataset.

7. Conclusion

From the previous report, we can conclude that the introduction of a Recurrent Neural Network architecture resulted in a improvement of test set accuracy over the baseline network trained in the previous report. As to our objectives, we find that 3 layers convolutional is better than 2 layers convolutional. Density layers and enough feature maps are necessary for the CNNs to capture the features of the musics. However, as to the low dimension of our inputs feature, CNNs do not perform better than our baseline model based on multilayer perceptron. Therefore, considering the sequence feature of our inputs, we find GRU cell RNNs is better than LSTM cell. Also, the 10^{-2} β value L2 regularization can fix the over-fitting problem of our GRU cell RNNs. Further study can consider using large number of feature dimensions and combing the CNNs and RNNs together to get a high accuracy in the music genre recognition task.

References

Bertin-Mahieux, Thierry, Weiss, Ron J, and Ellis, Daniel PW. Clustering beat-chroma patterns in a large music database. In *ISMIR*, pp. 111–116, 2010.

Chiliguano, Paulo and Fazekas, Gyorgy. Hybrid music recommender using content-based and social information. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 2618–2622. IEEE, 2016.

Choi, Keunwoo, Fazekas, George, and Sandler, Mark. Automatic tagging using deep convolutional neural networks. *arXiv preprint arXiv:1606.00298*, 2016.

Choi, Keunwoo, Fazekas, György, Sandler, Mark, and Cho, Kyunghyun. Convolutional recurrent neural networks for music classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pp. 2392–2396. IEEE, 2017.

Dieleman, Sander, Brakel, Philémon, and Schrauwen, Benjamin. Audio-based music classification with a pretrained convolutional network. In *12th International Society for Music Information Retrieval Conference (ISMIR-2011)*, pp. 669–674. University of Miami, 2011.

Glorot, Xavier and Bengio, Yoshua. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

Kalman, Barry L and Kwasny, Stan C. Why tanh: choosing a sigmoidal function. In *Neural Networks, 1992. IJCNN., International Joint Conference on*, volume 4, pp. 578–581. IEEE, 1992.

Kingma, Diederik P and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Sigtia, Siddharth and Dixon, Simon. Improved music feature learning with deep neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pp. 6959–6963. IEEE, 2014.

Wu, Ming-Ju and Jang, Jyh-Shing R. Combining acoustic and multilevel visual features for music genre classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 12(1):10, 2015.