MLP Coursework 3

G77 s1773005 s1607197

Abstract

We build a baseline model based on deep feedforward neural networks for 10 and 25 genre recognition tasks using Million Song Dataset. We first explore different hidden layers and hidden units to find the best model structure for these tasks. Based on the result, we choose 3 hidden layers with 200 hidden units to be our baseline model structure. Then we compare different optimization methods and activation functions. Based on the accuracy and speed, we choose Tanh activation function with Adam optimization methods to build our baseline experiments. As the deep feedforward neural networks can not capture the temporal relationships among features, we think of using RNN for our further experiments to increase the accuracy.

1. Introduction

This report will focus on the advanced model of deep neural networks for identifying the genre of songs in the Million Songs Dataset. The dataset has two subsets which are 10 Genre dataset and the 25 Genre dataset. In our report, we hope to build a model to predict the category of song accurately. The genre labels of most existing researches are from the Last.fm Dataset. It has a disadvantage which is it has too many genres. It is 200,000. And its songs and tacks are not one to one correspondence. Therefore, all existing studies use different methods to preprocess the data, resulting in the genre labels in different values. Genre labels which are used in existing research are not comparable. That is why we choose to use CD2C tagtraum genre annotations as our genre labels to train our models. It has a significant advantage is that its songs and tacks are one to one correspondence. In our research, we use recurrent neural networks to catch and identify timbre feature, chroma feature, and loudness feature respectively and to predict the song belongs to which category accurately. We choose CD2C and MSD Allmusic Style Dataset as our genre labels. Existing researches mostly use CNN to capture these features for these two kinds of genre classification tasks, but CNN cannot grasp the timing relationship between feature. Therefore we may consider using RNN for our further experiments.

2. Objective

In this section, we will focus on building a baseline experiment based on feed-forward networks in the Million Song Dataset. We will use the accuracy of 10 and 25 genre recognition tast to evaluate our model. And there are some experiments we could do. They are for exploring different hidden layer depths and widths, different optimization methods and different activation functions.

3. Data set

The Million Song Dataset is a freely-available collection of audio features and metadata for a million contemporary popular music tracks. The data for each track contains textual features, numerical descriptors, and various audio features derived using a music analysis platform provided by The Echo Nest. In the version of the data we provide, we include a 25 dimensional vector for each included segment, consisting of the 12 timbre features, 12 chroma features, and loudness at the start of segment concatenated in that order. To allow easier integration in to standard feedforward models, the basic version of the data we get includes features only for a fixed length crop of the central 120 segments of each track (with tracks with less than 120 segments therefore not being included). Therefore, the overall input dimension per track is 120x25=3000. And the input dimensions have been processed by subtracting the per-dimension mean across the training data and dividing by the per-dimension standard deviation across the training data. The Million Song Dataset does not provide any genre tags in its original form, but various external groups have already genre-tagged part of the data by cross-referencing the track IDs against external music tagging databases. Here we are provided two classification task datasets derived from the Million Song Dataset. The 10-genre classification task from CD2C tagtraum genre annotations(Schreiber, 2015) derived from multiple source databases (beaTunes genre dataset, Last.fm dataset, Top-MAGD dataset), with the CD2C variant using only non-ambiguous annotations. The 25-genre classification task used MSD Allmusic Style Dataset derived from the AllMusic.com database(Schindler et al., 2012).

4. Deep Feedforward Neural Network

4.1. Objective

In this experiment, we build a deep feedforward neural network to construct a baseline experiment for our further experiments. Several different experiments to study this deep feedforward neural network were performed, namely different number of hidden layers and number of hidden units.

4.2. Methods

The feedforward neural network is the simplest type of artificial neural network. We use multilayer perceptron to build our baseline. As it is well-understood approach with high speed and not bad performance on high dimension feature of dataset. In order to use this family of neural networks in TensorFlow, we will make use of the following methods:

- $tf.nn.softmax_{c}ross_{e}ntropy_{w}ith_{l}ogits$ - We use cross entropy as our lost function to compute the difference between outputs and targets. Also, since the task at hand is a classification problem, the last output of the network is using softmax as activation function. The softmax can normalized across every dimensions of the output layer so that they sum to one. Then each dimension of the output layer will represent the probability that the segment of the song is recognized as which genre. Therefore, it can classify the data point as one of the possible genres.

- $tf.train.AdamOptimizer$ - The optimizer that implements the Adam algorithm.

- $tf.nn.relu$ - Using Relu activation function to compute rectified linear of layers output.

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.

4.3. Experiments and results

We use a feedforward neural network with different number of layers and hidden units. As the basic Sigmoid activation function can cause a neural network to get âĂIJstuckâĂİ during training, we use ReLU activation functions(Nair & Hinton, 2010) between layers. There is always a fully connected softmax layer with the same number of hidden units of the cell in the feedforward neural network. The weights of this layer are initialized with Glorot initialization (Glorot & Bengio, 2010) and biases are initialized to 0. The lost function is cross entropy. The learning rate are 10^{-3} and first set epoch numbers to 50 in all experiments.

4.3.1. Number of hidden layers

In this experiment, we explored the performance of using different number of hidden layers in MSD dataset. Here we used 2 layers, 3 layers and 4 layers with 100 hidden units in the 10 and 25 Genre MSD Dataset respectively. The results are shown in figures and tables below.

4.3.2. Discussions

From the results we can find that the different layers do not increase the performance of the models in both the 10 and 25 genre recognition tasks. It seems that increase the number of hidden layers do not help the model to capture the feature of segments of songs very well. Even with L2 regularization, we can find that there is overfitting for the accuracy of both 10 and 25 genre recognition. Therefore, we just choose 1 hidden layers based on the similar experiment before (Scaringella et al., 2006).

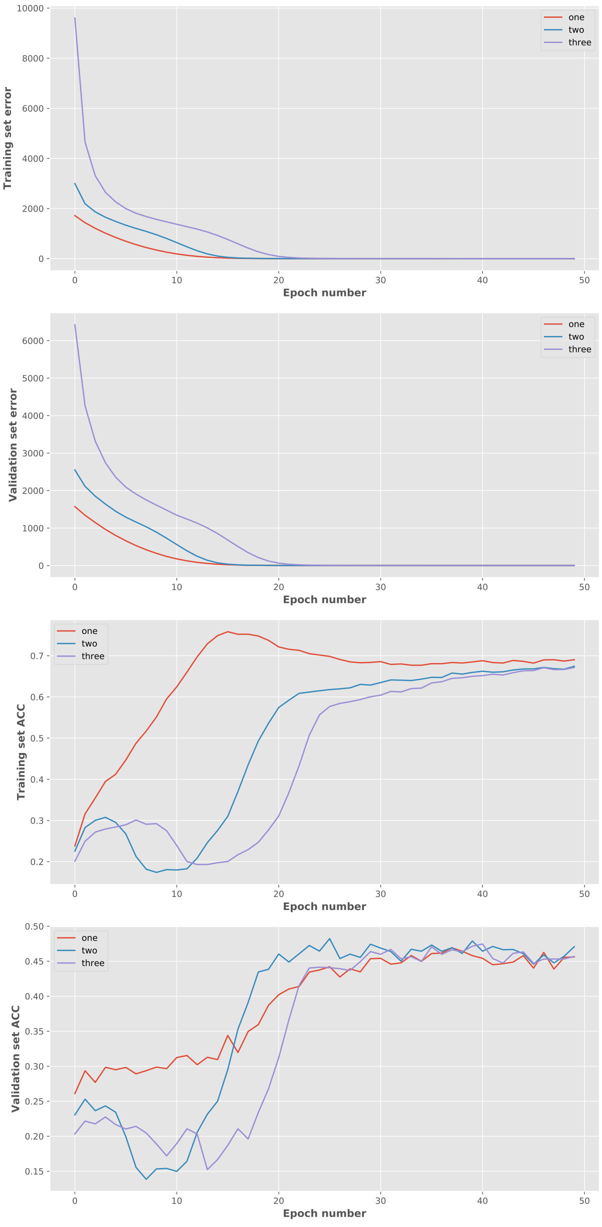

Figure 1. Comparing one, two and three hidden layers performance in error and accuracy in the training and validation set over 50 epochs in the 10 Genre MSD Dataset.

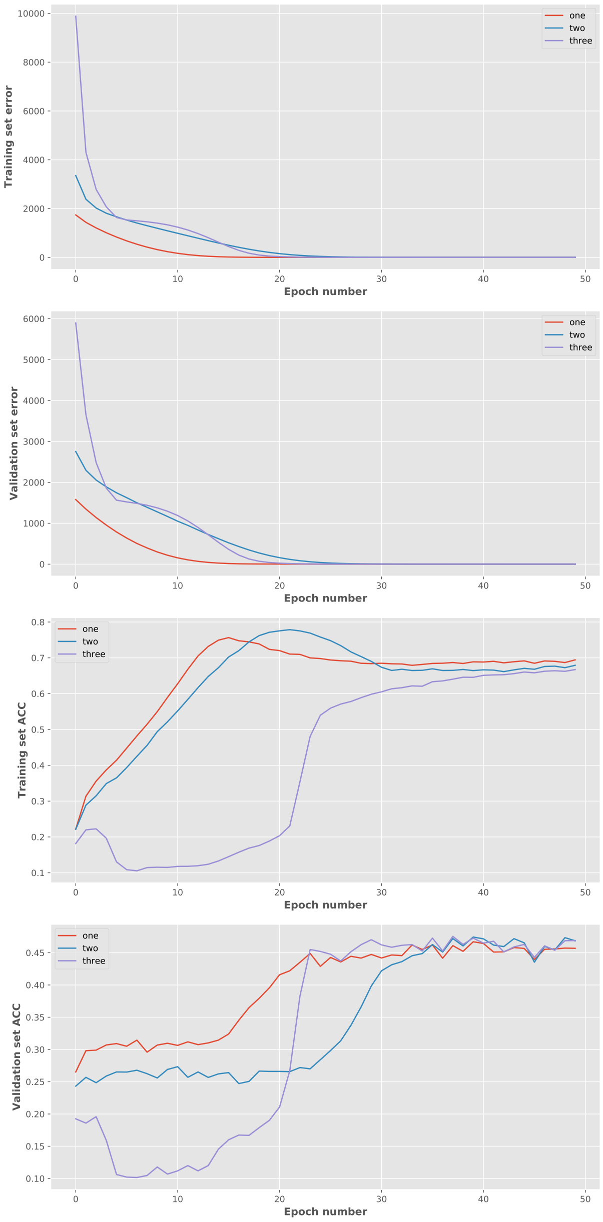

Figure 2. Comparing one, two and three hidden layers performance in error and accuracy in the training and validation set over 50 epochs in the 25 Genre MSD Dataset.

	Measure	MSD 10 Genre	MSD 25 Genre
one	Error	2.12	2.10
	Accuracy	0.46	0.46
two	Error	2.06	2.12
	Accuracy	0.47	0.47
three	Error	2.16	2.08
	Accuracy	0.46	0.47

Table 1. Comparing one, two and three hidden layers performance in error in the training and validation set over 50 epochs in the 10 Genre MSD Dataset.

4.3.3. Number of hidden units

In this experiment, we explored the performance of using different number of hidden units in the MSD dataset. Here we used 100 units, 200 units and 300 units with three hidden layers in 10 and 25 Genre MSD Dataset respectively. First, we try 50 epoch for all the hidden units. However, as to 200 and 300 hidden units, it needs longer run to get the minimum error for these two model structure. Therefore, we set epoch to 200 for 200 and 300 hidden units. The results are shown in figures and tables below.

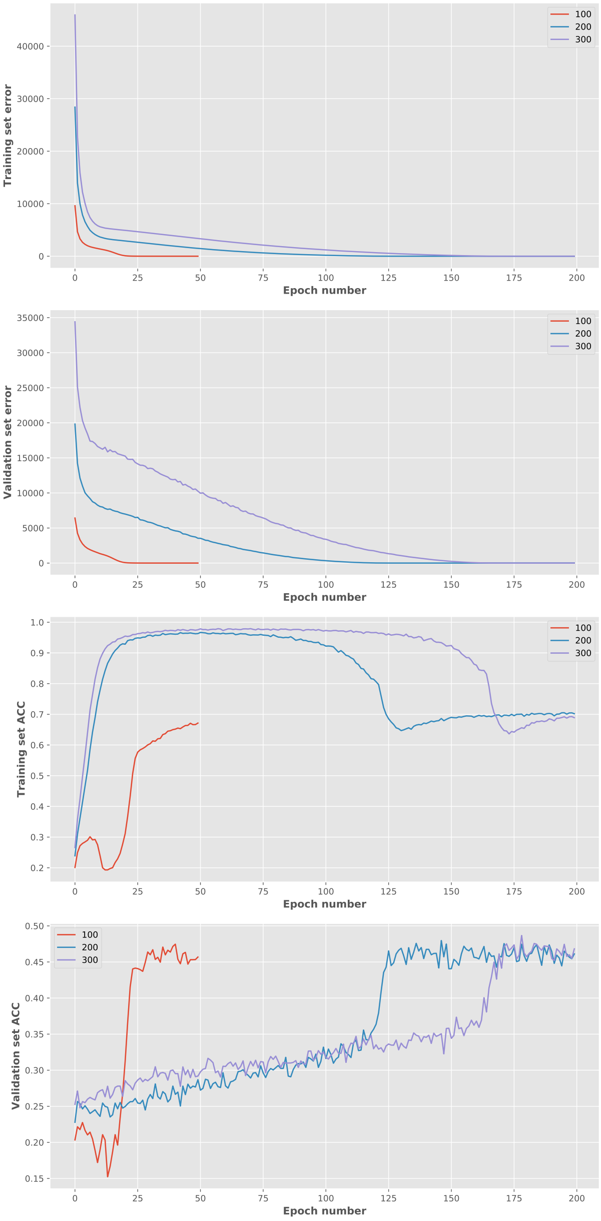

Figure 3. Comparing 100, 200 and 300 hidden units performance in error and accuracy in the training and validation set over 50 epochs in the 10 Genre MSD Dataset.

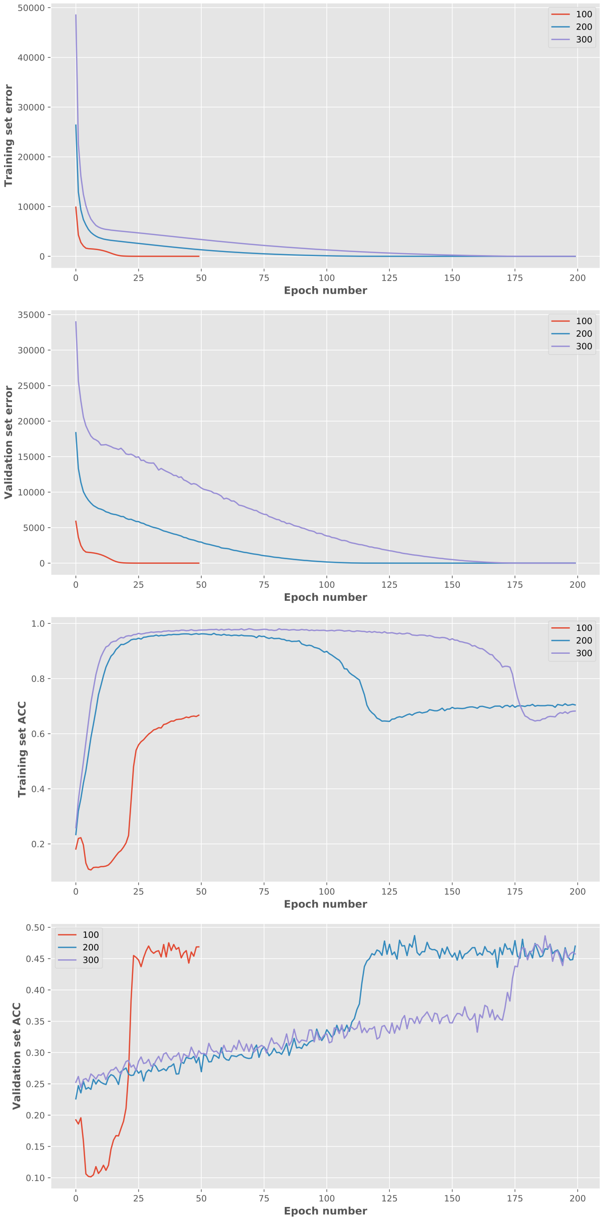

Figure 4. Comparing 100, 200 and 300 hidden units performance in error and accuracy in the training and validation set over 50 epochs in the 25 Genre MSD Dataset.

	Measure	MSD 10 Genre	MSD 25 Genre
one	Error	2.16	2.08
	Accuracy	0.46	0.47
two	Error	2.21	2.20
	Accuracy	0.46	0.47
three	Error	2.17	2.19
	Accuracy	0.47	0.46

Table 2. Comparing 100, 200 and 300 hidden units performance in error in the training and validation set over 50 epochs in the 10 Genre MSD Dataset.

4.3.4. Discussions

From the results we can find that the different hidden units do not increase the performance of the models in both the 10 and 25 genre recognition tasks. Therefore, we just choose 200 hidden units based on the similar experiment before (Scaringella et al., 2006).

5. Optimization Methods

5.1. Objective

In this experiment, we compared different optimization methods, such as adaptive moment estimation(Adam)(Kingma & Ba, 2014) and RMSProp(Tieleman & Hinton, 2012) on the performance of feed-forward networks in the Million Song Dataset with 10 Genre classification task.

5.2. Methods

The optimization methods of feedforward neural networks used until this point of this report were Adam. However, there is an alternative function to the Adam: RMSProp. In order to implement RMSProp in tensorflow, we use the following methods:

- $tf.train.RMSPropOptimizer$ - The optimizer that implements the RMSProp algorithm.

We compare these two different optimization methods based on 3 hidden layers and 200 hidden units on the 10 and 25 genre Million Song Dataset.

5.3. Experiments and results

In this experiment, we use the Adam and RMSPROP optimization methods by a feedforward neural network with 3 hidden layers and 200 hidden units. All the other setting are the same as before.

5.3.1. RMSProp and Adam

In this experiment we studied the performance of using RMSProp and Adam in the 10 Genre MSD dataset. Training and validation error and accuracy plots can be observed in *Table 2* and *Table 3*.

Optimizer	Training Error	Validation Error	ACC
RMSProp	0.06	29.42	0.32
RMSProp(L2)	1.47	2.19	0.45
Adam(L2)	1.47	2.12	0.46

Table 3. Comparing RMSProp and Adam optimizer performance in error in the training and validation set and accuracy in the validation set over 50 epochs in the 10 Genre MSD Dataset.

As to RMSProp, the validation error is higher than training error, which means overfitting. Therefore, we use L2 regularization with $\beta = 10^{-2}$ to fit it. The result is much better for the error of RMSProp optimizer.

Comparing RMSProp and Adam, we found that the accuracy of both RMSProp and Adam are the same. Also, the training speed is almost same between these two optimization methods. As Adam is a much more like default choice(Scaringella et al., 2006), we use Adam as our optimization method on the following experiments.

6. Active functions

6.1. Objective

In this experiment, we compared two activation functions: Relu(Nair & Hinton, 2010) and Tanh(Kalman & Kwasny, 1992) on the performance of feed-forward networks in the Million Song Dataset with 10 and 25 Genre classification task.

6.2. Methods

The activation function of feedforward neural networks used until this point of this report were Relu activation function. However, there is an alternative function to the Relu: the hyperbolic tangent, or tanh function.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \tag{1}$$

which has the gradient:

$$\frac{d}{dx}\tanh(x) = 1 - \tanh^2(x) \tag{2}$$

As the basic logistic sigmoid function can result in model parameters that are updated less regularly than we would like, and are thus âĂIJstuckâĂİ in their current state. Both Relu activation function and Tanh activation function can fit this "stuck" very well. Therefore, we compare these two activation functions based on three hidden layers and 200 hidden units on the 10 and 25 genre Million Song Dataset. In order to implement the Tanh activation function in TensorFlow, the method $tf.nn.tanh$ was used.

6.3. Experiments and Results

In this experiment, we use the Relu and Tanh activation function by a feedforward neural network with different hidden layers and hidden units. As we can see before, we set 200 epoch number for Relu activation function. We first set 50 ephco number for Tanh activation function. Both Relu and Tanh are using L2 regularization to fit overfitting. The results are shown in figures and tables below.

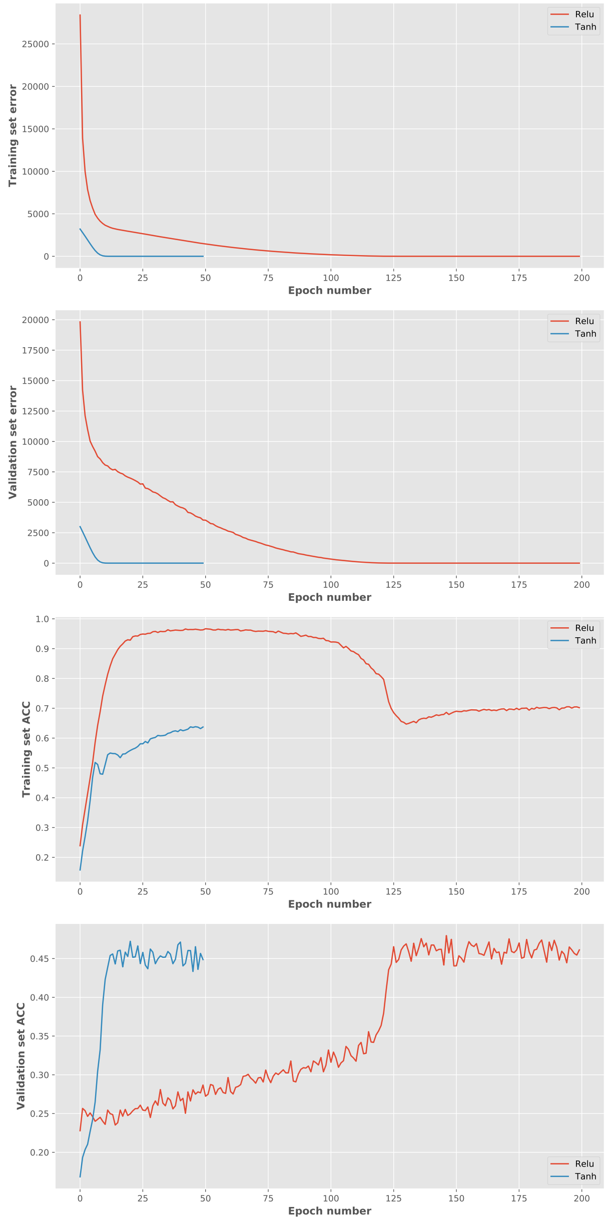

Figure 5. Comparing Relu and Tanh activation function performance in error and accuracy in the training and validation set over 50 epochs in the 10 Genre MSD Dataset.

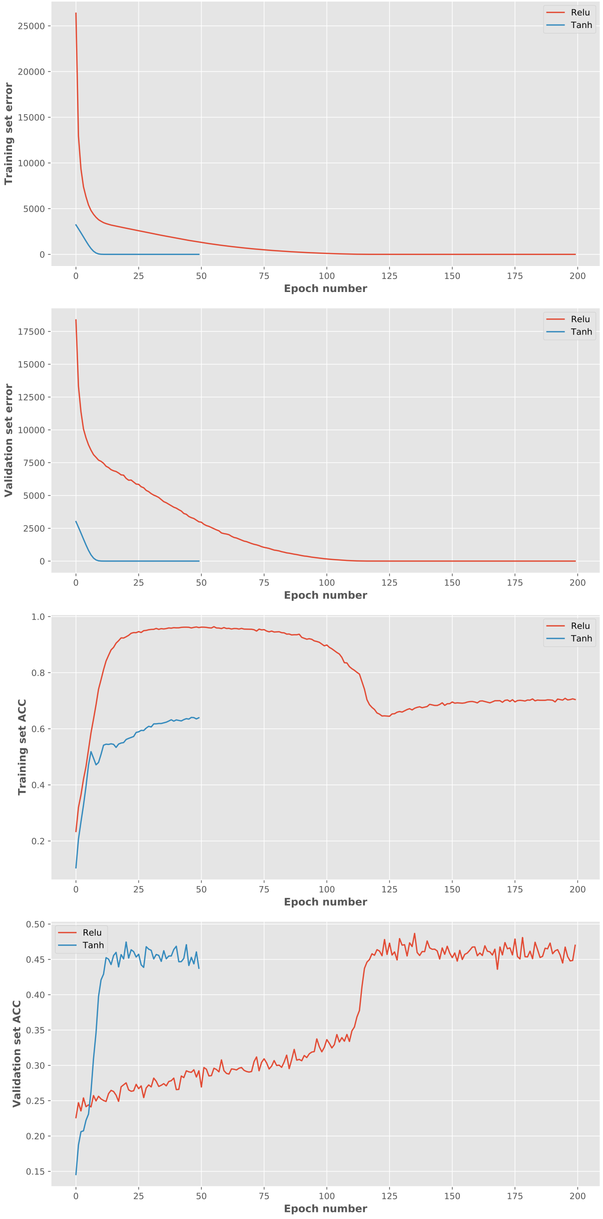

Figure 6. Comparing Relu and Tanh activation function performance in error and accuracy in the training and validation set over 50 epochs in the 25 Genre MSD Dataset.

	Measure	MSD 10 Genre	MSD 25 Genre
Relu	Error	2.21	2.20
	Accuracy	0.46	0.47
Tanh	Error	2.24	2.27
	Accuracy	0.45	0.44

Table 4. Comparing Relu and Tanh activation function performance in error in the training and validation set over 50 epochs in the 10 Genre MSD Dataset.

6.4. Discussion

From the result we can find that both Tanh and Relu activation function have similar accuracy in 10 and 25 genre recognition task. However, as to Tanh activation function, it can only need 50 epoch numbers to reach the minimum error and get nearly 45 percent accuracy. The Relu, though, need 200 epoch numbers to get the similar accuracy. Therefore, we choose Tanh activation function for our baseline experiments. Also, from the training and validation accuracy figures, we can find that both Tanh and Relu activation function have overfitting even with L2 regularization. Therefore, further experiments will consider another network structure, such as Recurrent Neural Network to fit these problem and also consider different regularization methods.

7. Interim conclusions

To sum up, we finally choose the Adam optimization method, Tanh activation function, 3 hidden layers and 200 hidden units as the final Baseline. As this kind of model has both great speed and accuracy for 10 and 25 genre recognition task. However, as the deep feedforward neural network can not capture the temporal elements among different features of segments, we would like to choose RNN for our further experiments to increase the accuracy of our model.

8. Plan

In the process of our experiment, we found that deep feedforward neural network cannot capture and trace the time-series of music features very well. Therefore, we consider using RNN to improve the accuracy of the model in order to predict the category of songs.

References

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.

Nair, Vinod and Hinton, Geoffrey E. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

Scaringella, Nicolas, Zoia, Giorgio, and Mlynek, Daniel. Automatic genre classification of music content: a survey. *IEEE Signal Processing Magazine*, 23(2):133–141, 2006.

Schindler, Alexander, Mayer, Rudolf, and Rauber, Andreas. Facilitating comprehensive benchmarking experiments on the million song dataset. In *ISMIR*, pp. 469–474, 2012.

Schreiber, Hendrik. Improving genre annotations for the million song dataset. In *ISMIR*, pp. 241–247, 2015.

Tieleman, Tijmen and Hinton, Geoffrey. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.