Machine Learning (CS60050)

Project- Popularity of Music Records Prediction using Artificial Neural Network

Anushruti Pandey(20CS60R17)
Suyash Tiwari(20CS60R18)
Vishal Gourav (20CS60R21)

March 3, 2021

Introduction

The project uses concepts of Deep Learning, in particular **Artificial Neural Network** to train on a given data set and then predict the output on the test set. Sticking to the modular approach of project development, the functionalities have been divided into different functions which are discussed further in the report.

The functions

The different functions used in making of the project are described as follows:-

Preprocess(data)

The data is processed in this function. The inputs are of int, object and float type which are all converted to float using normalization technique called z-score normalization. The last column i.e. "Top10" is left as int datatype. A point to be noted here is that on every execution of the program the **data set is divided into different training and test sets**, i.e. every execution is unique.

Dataloader()

The data is loaded in form of mini batches of defined size. Generally the sizes are multiples of 2. The default value being 32.

init_weights()

The weights are initialized in this step for every layer of the ANN.

forward_pass()

The function finds the value at each layer of the ANN using *sigmoid* and *ReLu* functions till all the layers are covered for every epoch.

backpropogation()

The function finds the error in weights and backpropogates the values to the previous layers so that they in turn can update their weights. This is also done at every epoch.

training()

The forward_pass() and backpropogation functions are called in this step for every mini batch at evry epoch till all the data is exhausted and all the epochs are finished.

test()

Here the test data is run against the ANN to see how accurately our model is predicting the data.

Results

On running of the program different results were obtained at different hyper parameters which are as follows:-

Accuracy for default

The accuracies obtained for the default hyper parameters are:-

- Training Accuracy: 85,24%
- Test Accuracy: 84.3%

Accuracies reported above may vary slightly as a different data set distribution is done everytime. The graph obtained for the training accuracy and test accuracy after every 10 epochs are as follows:-

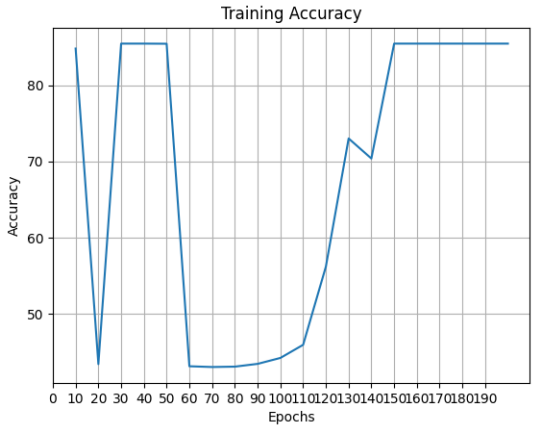

Figure 1: Train Accuracy

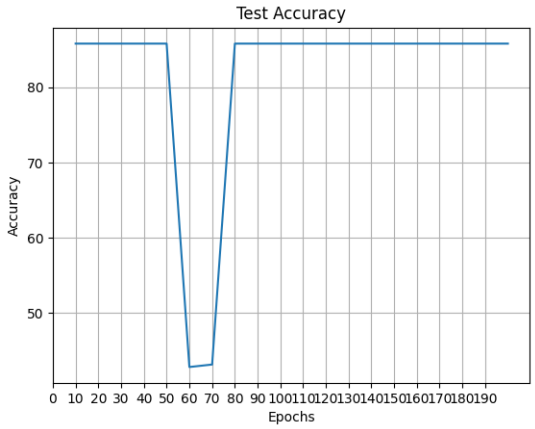

Figure 2: Test Accuracy

Changing the Hyper parameters

The variation of different hyper parameters and their results are as follows:-

- **Accuracy vs Batch Size.** The values of batch size vary in the range 1, 2, 4, 8, 16, 32, 64, 128. The graph is as follows:-

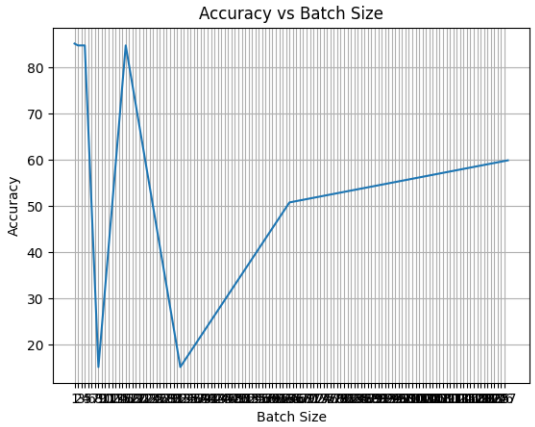

Figure 3: Accuracy vs Batch Size

- **Accuracy vs Number of hidden layers.** The values of the number of hidden layers vary in the range 1, 2, 4, 6, 8 . The graph is as follows:-

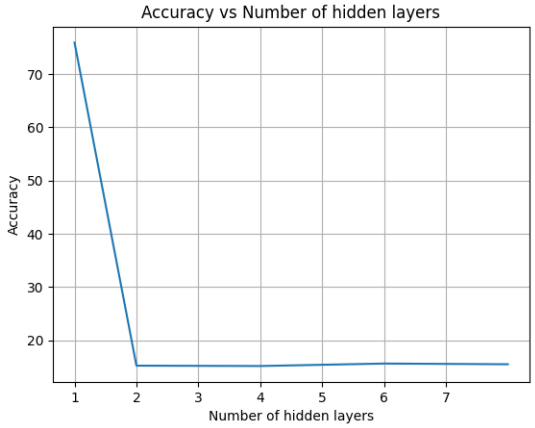

Figure 4: Accuracy vs Number of hidden layers

- **Accuracy vs Learning Rates.** The values of the learning rates vary in the range 0.01, 0.05, 0.1, 0.2, 0.4, 0.8. The graph is as follows:-

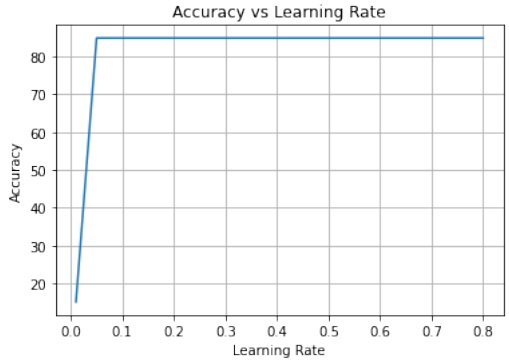

Figure 5: Accuracy vs Learning Rate

Best Accuracy

The best accuracy was obtained at Batch size:4, learning rate:0.01 and hidden layers:8 is 85%.

Overfitting

Here we have tried to train the model for the first 1000 data points for 200 epochs. This leads to the model overcompensating (i.e. overfitting) the weights of the trained model. Therefore the testing leads to the accuracy increasing during the initial epochs and then going down again. The same can be observed in the below graph:-

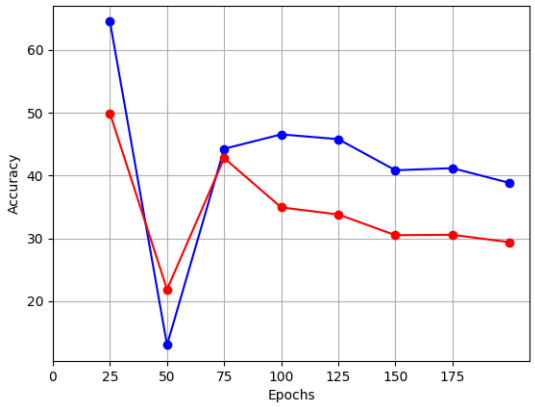

Figure 6: Effect of Overfitting on Training(Blue) and Test(Red) accuracy