Layout/Techniques Of Project

- I prepared data for deep learning applications using Python and an audio analysis library called Librosa. I preprocessed an audio dataset and got it ready for music genre classification. Specifically, I implemented code to process the music dataset, in order to extract MFCCs and genre labels. Then, I saved the data in a json file, in a format that's convenient for retrieval when training the classifier.

- I trained this dataset using neural networks and implemented forward-propagation, back-propagation and gradient descent using tensorflow and keras.

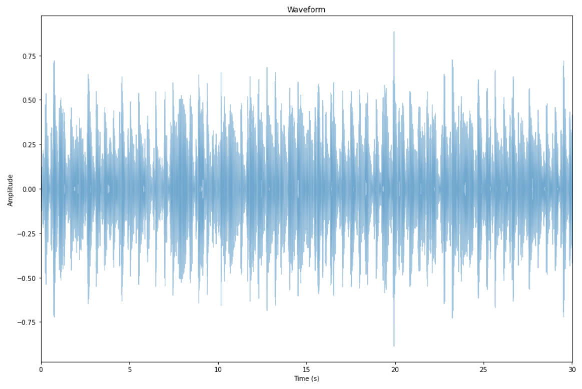
- I also introduced a few fundamental deep learning concepts such as binary/multiclass classification, rectified linear units, batching, and overfitting.I also implemented dropout and regularisation in the music genre classifier.

- -While building the network I also introduced few fundamental deep learning concepts such as binary/multiclass classification, rectified linear units, batching and overfitting.

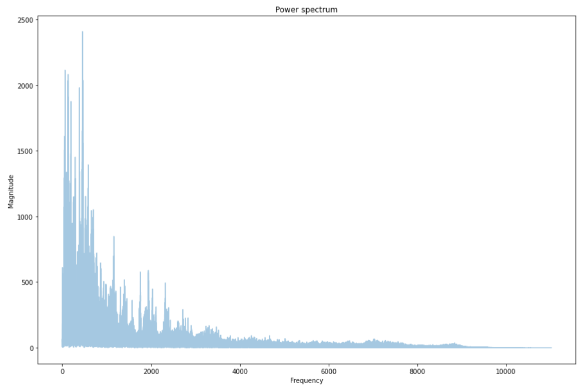
- I also implemented **RNN, LSTM** for music genre classification in Tensorflow.

I studied audio dataset and extracted different waveforms of an audio file

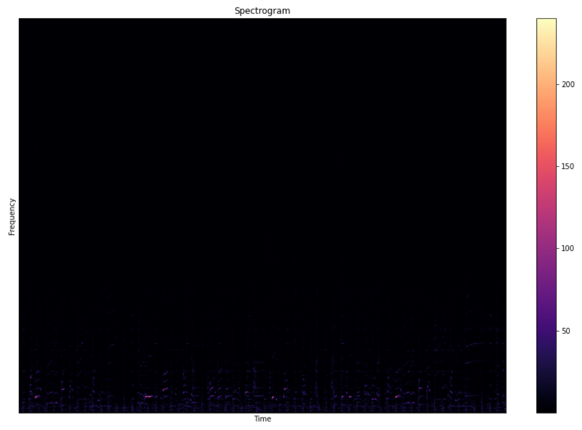
1. Waveform (time domain)



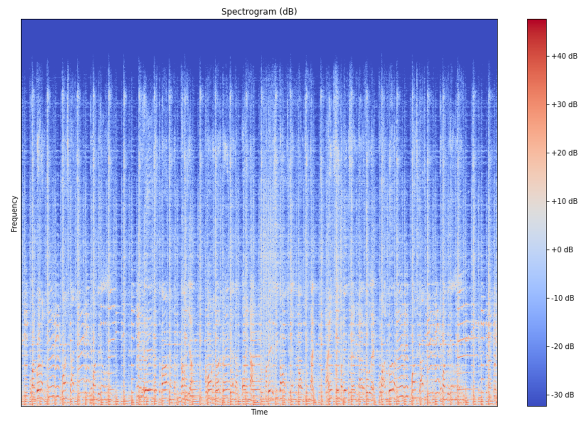
2. Waveform (frequency domain)



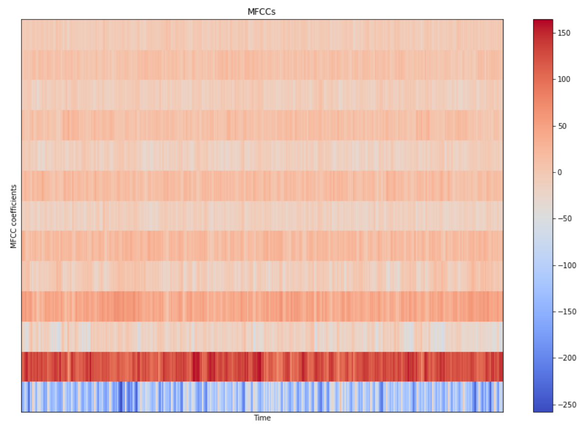
3. Spectrogram



4. Spectrogram (in decibels)

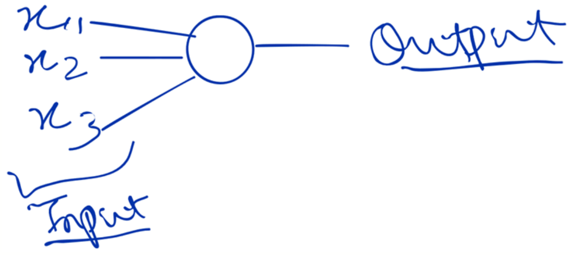


5. 13 MFFCs



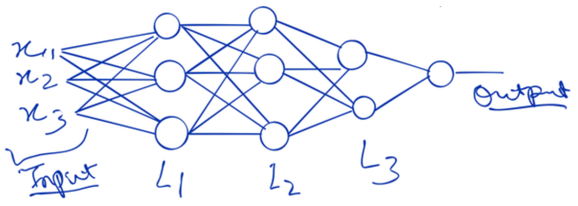
Neural Networks

A neural network is used in Deep Learning. For visualization, we consider it to be made up of lines and circles(will come into their intuition later). Each bunch of circles is called a layer and layers are connected using the lines. Input and Output layers are present for all neural networks. For a single training example, the input layer consists of the features and the output layer gives the output.



Now, let's move into the circle. We can consider the circle as a function, which receives multiple inputs and gives out a single output, also known as the activation. Each line in the network has a certain weight, which is randomly initialized in the beginning. The input of a function is the weighted sum of the activation of the previous layer. Basically it can be written as

Input for a circle of layer (l+1)=Summation (Output(Activation) of a circle in the layer l) * (the weight of the line which connects the circle)



This is then given to a function and that gives the output(activation) of a circle of layer l. This way, we reach the output layer from the input layer. We have to define a cost function here, which will tell us how far is our output from the actual output. Our output should be far away from the actual output as the weights are initialized randomly. This is called forward propagation.

Now, we have to find the derivative of the cost function with respect to every weight. After we have found the derivatives, we have to reduce each weight by (learning rate * derivative) . This step is called backpropagation.

I implemented these 3 types of Neural Networks:-

- **Artificial Neural Network(ANN)**
- **Recurrent Neural Network(RNN)**
- **Long Short Term Memory(LSTM)**

I implemented these two methods to solve overfitting

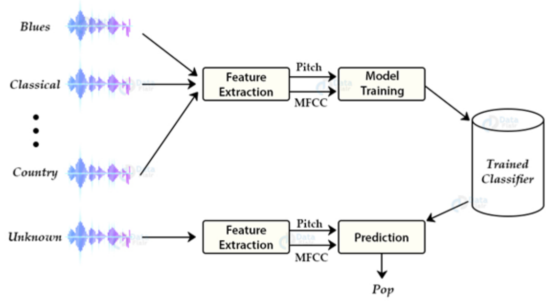
Dropout

- Randomly drop neurons while training

Regularization

- Add penalty to error function
- Punish large weights
- L1 and L2

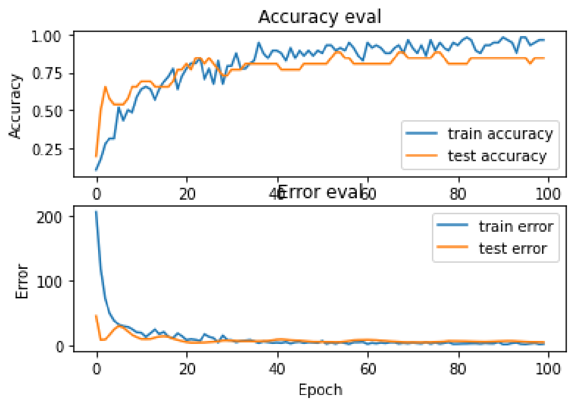
Block diagram



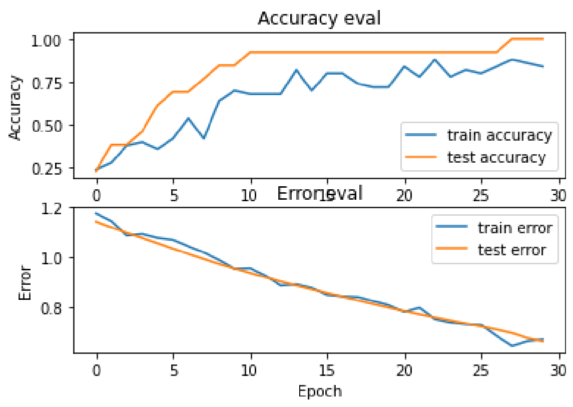
Conclusion

I implemented following two models:-

1. Deep Neural Networks



2. LSTM network


Accuracy eval
Accuracy
train accuracy
test accuracy
Error eval
Error
train error
test error
Epoch