Report for Assignment-1

EE 603A

Assignment 1 : Audio Classification

Ayush Kumar (200246)

Department of Electrical Engineering

1. Introduction

Machine Learning is one of most opportunity filled emerging fields in computer science and mathematics which has a huge application in almost every domain one can come across. It's emerging in the sense that new techniques faster than their predecessors are still coming out and they create a system that can learn itself about a particular type of data. Audio Signal Processing is no exception to the trend of Machine Learning applicable domains. The most renowned use of Machine Learning and it's subset Deep Learning is found a lot in Automatic Speech Recognition (ASR), Natural Language Processing (NLP), Pitch Detection, Music Generation and more. In this report a **Deep Neural Network (DNN)** model is used to classify different types of sounds if given the spectrograms of those sounds.

2. Methods

Data Pre-processing

1. The shape of the .npy matrix is (1, 128, X) where X is a variable varying from 29 to 2584.
2. We need to make the shape of all the .npy matrix of equal dimensions.
3. To do that we can concatenate a 3D numpy array filled with zeroes such that the dimension of all the matrix is equal to the maximum dimension from all the given .npy matrix that is (1, 128, 2584).
4. Now after concatenating the matrix we need to flatten the matrix of each .npy array using np.flatten() and take it's mfcc using the given parameters which are sample rate = 44.1 kHz, hop_length = 512, Nfft = 2048, nmels = 128 and windowing function = hann.
5. The matrix we get by taking the mfcc of a .npy array of shape (1, 128, 2584) is a matrix with shape (128, 647). Now this matrix is ready to be used in various models like logistic regression, Deep Neural Network & Convolution Neural Network.

Logistic Regression

1. For Logistic Regression we can import Logistic Regression Model directly from the sklearn library.
2. To fit the data, we need to flatten each of the .npy array into a 1-D array.
3. Using this model is F1 Score is 0.88, precision is 0.89 & recall is 0.88 by splitting the given 1000 samples into 80% train and 20% test data.

Deep Neural Network

1. We can use the DNN by importing Sequential() model using keras and can add multiple layers using model.add(Dense = 32, activation = ReLU) .
2. Here the shape of each .npy matrix will be flatten and of 1 dimension.

3. We can adjust the no. of neurons and no. of layers by hit & error & iterative method to fine tune the model such that the accuracy on the test data is good enough.
4. Using this model in 20 epochs the train accuracy reaches 99% while the test accuracy reaches 75%. The F1 Score is 0.749, Recall is 0.74 and Precision is 0.74 by splitting the 1000 sample data into 80% training and 20% testing samples.

Convolutional Neural Network

1. Using the CNN Model, we get a F1 Score of 0.88, Recall of 0.89 & Precision of 0.89
2. This is the best model trained so far. We get an accuracy of 87.8% in test data and 99.5% in training data.
3. Here we need the reshape the each .npy array into a 3D matrix of dimension (128, 647, 1).

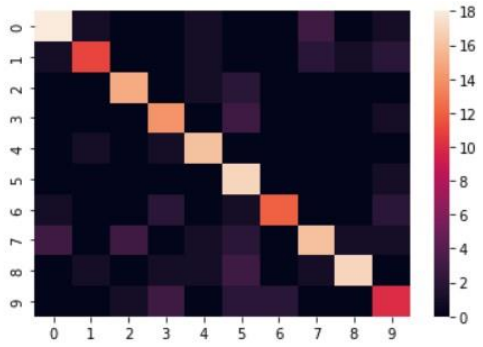
3. Model Performance metrics

Precision: - `74.422336887585`

Recall: - `72.636815920398`

F1 Score: - `72.5891931629739`

Confusion Matrix



```
[[18  1  0  0  1  0  0  3  0  1]
 [ 1 11  0  0  1  0  0  2  1  2]
 [ 0  0 15  0  1  2  0  0  0  0]
 [ 0  0  0 14  0  3  0  0  0  1]
 [ 0  1  0  1 16  0  0  0  0  0]
 [ 0  0  0  0  0 17  0  0  0  1]
 [ 1  0  0  2  0  1 12  0  0  2]
 [ 3  0  3  0  1  2  0 16  1  1]
 [ 0  1  0  1  1  3  0  1 17  0]
 [ 0  0  1  3  0  2  2  0  0 10]]
```

4. Observations and Discussion

We observe that CNN gives the best accuracy and F1 Score among all the models i.e., DNN, Logistic Regression. CNN shows the best accuracy as they account for the spatial differences in the input by using a sparse locally connected structure. CNN model the time and the frequency components between adjacent audio samples. Therefore, CNN has outperformed both DNN & Logistic Regression.