Task 1

(TensorFlow Speech Recognition Challenge)

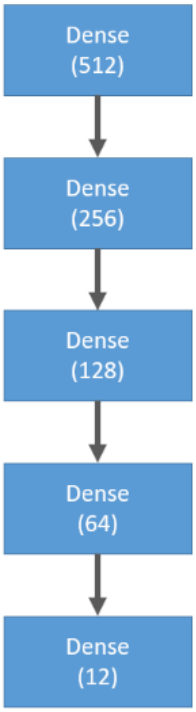
Experiment 1

Dataset Description:

Librosa package is used to load the .wav files. It returns the audio samples and sampling rate of the audio. Sampling rate is the number of samples recorded per second. To create the input dataset, raw audio samples are used in this experiment.

Instead of using all the samples from an audio, every sr/100 samples are taken from the first 20,000 samples, where sr is the sampling rate. Samples are padded with 0's in order to get the maximum input length of 100.

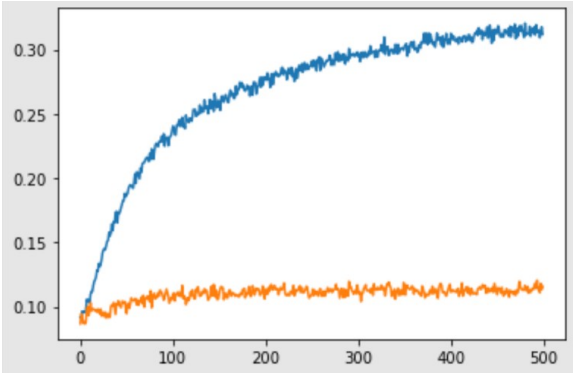
To create samples for "unknown" label, equal number of samples are taken from audio which will not be used for evaluation. For "silence", single audio is broken into chunks having 20,000 samples and the above same procedure is used for creating input sample.



Input Shape Description:

Input Shape: (BATCH, 100)

Output Shape: (BATCH, 1)



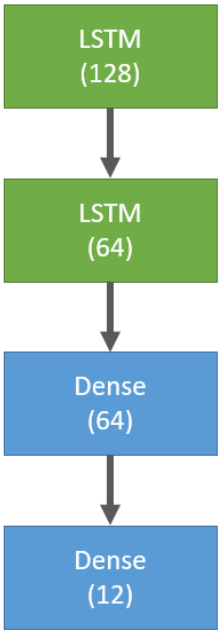
Model Description & Training:

The model developed had 4 hidden layers in between the input and output layers. All the hidden layers were fully connected layers with 512, 256, 128 and 64 dense units respectively.

The model was trained for 500 epochs with batch size of 256. In order to receive the best results from the model, a checkpoint callback was added which monitored validation accuracy and only the best weights were saved. After completing the training, the train accuracy was **~31%** while validation accuracy was only **~12%.**

Experiment 2

In this approach, same dataset as discussed in previous approach is used. But instead of using a dense network, an RNN network is used. The input data is reshaped from (BATCH, 100) to (BATCH, 100, 1), i.e. each input is having 100 timesteps with 1 feature each.

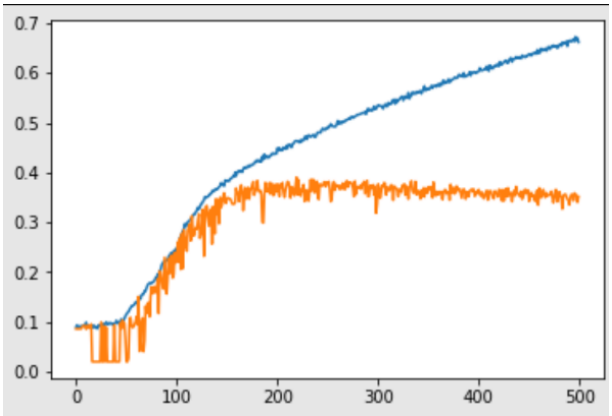


The input data is passed through two LSTM layers stacked together. The output of the layers is fed into a fully connected layers. The layer is followed by another dense layer which is an output layer having 12 tensors denoting the 12 class labels.

After training, both train and validation accuracy improved if compared to previous approach but was still very less.

Train Accuracy: **46.56%**

Test Accuracy: **39.58%**



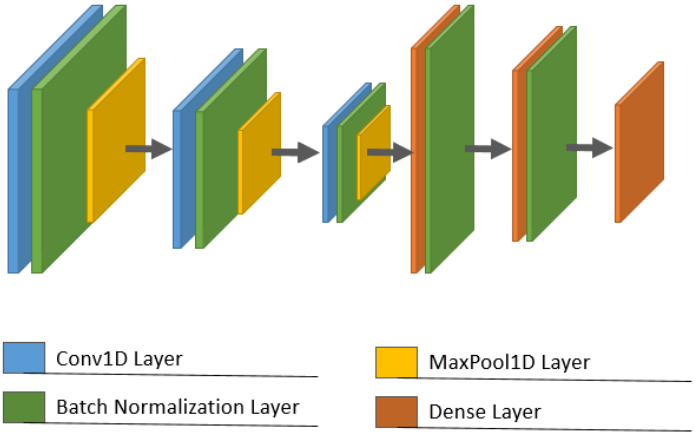
Experiment 3

The previous RNN approach improved the results to some extent. In order to improve the accuracy using the previous approach, new input dataset was created. The new dataset had more timesteps and features. The shape of the data was (BATCH SIZE, 315, 7). This time instead of creating an unknown label, all labels were considered for classification.

Same model (with different number of output tensors) discussed in the previous experiment was used for this experiment but the results were very poor. The accuracy never increased above **0.04%**. The test accuracy was **0.0343%**.

Experiment 4

In this experiment, instead of using an RNN approach, convolutional approach was adopted while using the dataset of previous experiment. The model architecture adopted in this experiment is shown below:



The accuracy achieved in this experiment was significantly high than the previous experiments, but not satisfactory.

Train Accuracy: **~64%**

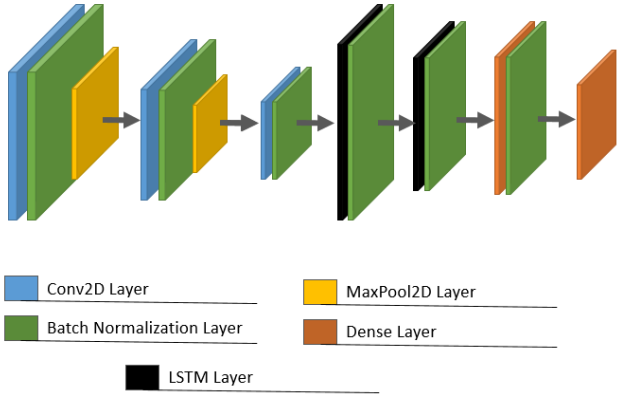
Test Accuracy: **64.33%**

Experiment 5

In this experiment instead of using raw audio samples, MFCC was generated using librosa package and was used as input dataset.

Train Accuracy: **~89%**

Test Accuracy: **81.46%**



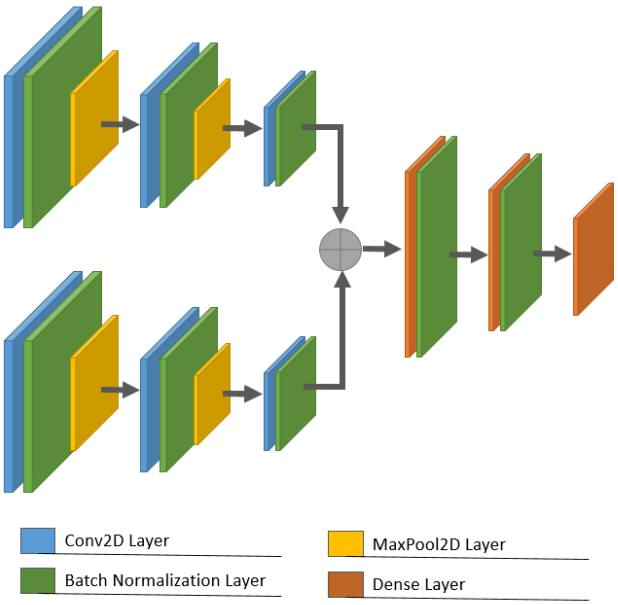
Submission Scores

Private Score: **69.411%**

Public Score: **67.982%**

Experiment 6

In this experiment, instead of using single feature (MFCC) another feature (Chroma STFT) was also used as input. These features were extracted using the librosa package. Both of the features were pre-processed parallelly using a CNN and then were concatenated to pass through dense layers of network. The model architecture is shown below:



Train Accuracy: **98.527%**

Test Accuracy: **93.839%**

Submission Scores

Private Score: **78.550%**

Public Score: **77.796%**

Results:

	Train Accuracy	Test Accuracy	Remarks
Experiment – 1	31 %	12 %	Bad accuracy because raw unstructured audio input was used
Experiment – 2	46.56 %	39.58 %	Accuracy improved as raw audio input was structured for RNN
Experiment – 3	0.04 %	0.0343 %	Still confused why accuracy is too low
Experiment – 4	64 %	64.33 %	Convolutional approach gave better results as data was structured in a spatio-temporal format
Experiment – 5	89 %	81.46 %	Accuracy improved because feature was used instead of raw audio
Experiment – 6	**98.527 %**	**93.839 %**	**Highest accuracy achieved because more number of features were used. It might increase if other features such as: Chroma CENS, Chroma CQT are used.**

References

1. Nagesh Singh Chauhan, Audio Data Analysis Using Deep Learning with Python (Part 1), https://www.kdnuggets.com/2020/02/audio-data-analysis-deep-learning-python-part-1.html
2. Vivek Amilkanthawar, Deep Learning Using Raw Audio Files, https://medium.com/in-pursuit-of-artificial-intelligence/deep-learning-using-raw-audio-files-66d5e7bf4cca