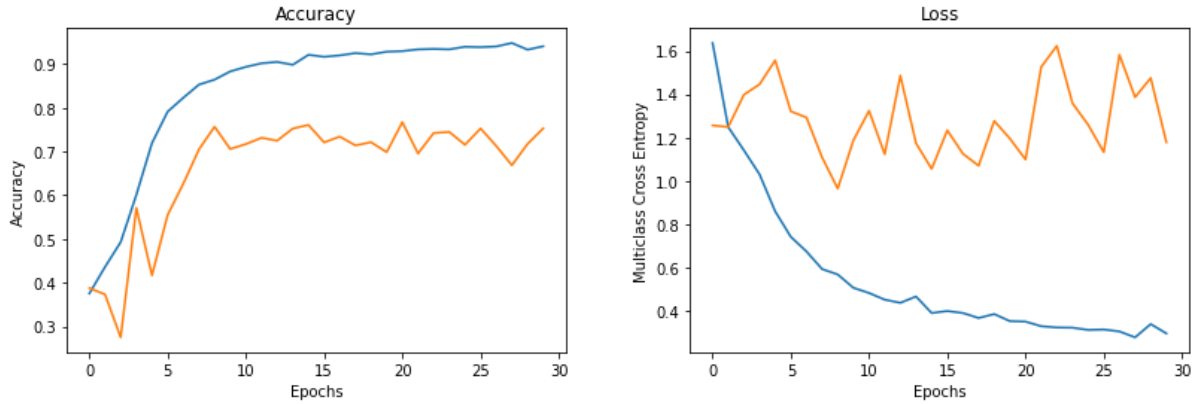
Ways to get the final model: several ways are tried. In my first trial, the validation set is generated by randomizing slice, which means after all audios are preprocessed by librosa, several slices of each audio are selected randomly by train_test_split() function. In this way, the training loss and validation loss decrease simultaneously, and the accuracy on training set and validation set are quite high, achieving just below 90% with a single GRU layer with 256 unit, which is amazing.

But after examining the procedure, I cannot find another audio which is totally unseen by the network to train the whole system. So, I believe the validation manner above is not suitable considering the scenario when the network is used in real life where the test audio could not be learnt by the system. The next thing I do is to split the training set by slice before the random shuffle. In this way, there will be some audio files unseen by the network, and the validation accuracy is surely much lower than the manner before, only about 50%, while the accuracy of the training set remains high still.

To refine the accuracy on validation set, the length of each slice is set to 2000, which means every 20 second is used to train the system. The whole dataset is re-examined to get rid of the abnormal data, such as the wrongly classified audio as an English audio is supported as Hindi, and some extra-long audio which stretched to about 30 min and is quite slow to figure out the language. The silence is reduced by setting top_db=35 in the function librosa.effects.split(), and **each audio file is then removed silence by selecting the intervals with energy above 35dB**. To get a ratio among these three classes and making them multiple of 2000, the later part of each classes is discarded, leaving English an (5600000, 64) matrix, Hindi a (1400000, 64) matrix, Chinese a (4200000, 64) matrix. The validation set is selected by 0.8*number of features. The later part of the whole dataset is set as validation set. Then the data from three dataset is combined together to form training set and validation set separately, and then shuffled. Another thing I've done is to set weight to each class. According to the ratio offered above, the weight of English, Hindi and Chinese is set as 0.75, 3, 1 respectively. To avoid overfitting, l2 regularizer with parameter 0.001 is used in GRU layers both in recurrent matrix and kernel matrix, and the forward dropout rate is set to 0.4. Besides, two batch normalization layers and one dropout layer with dropout rate 0.4 are used.



(a) Accuracy
blue one denotes training set while orange one denotes validation set.

(b) Loss
blue one denotes training set while orange one denotes validation set.

Fig. 1 Training Curve

In the final model, two layers of GRU are used to get a better result. When converting to streaming model, the last GRU unit is used as streaming model to fit for the output shape of the former GRU. The final training accuracy is 94.02%, the final validation accuracy is 75.27%. The training history is shown in

Fig. 1, the plot of the model is shown in Fig. 2. From the description above, the last few files could be regarded as test files. The first 60 second is used to test the effect of streaming model, the results are shown in Fig. 3

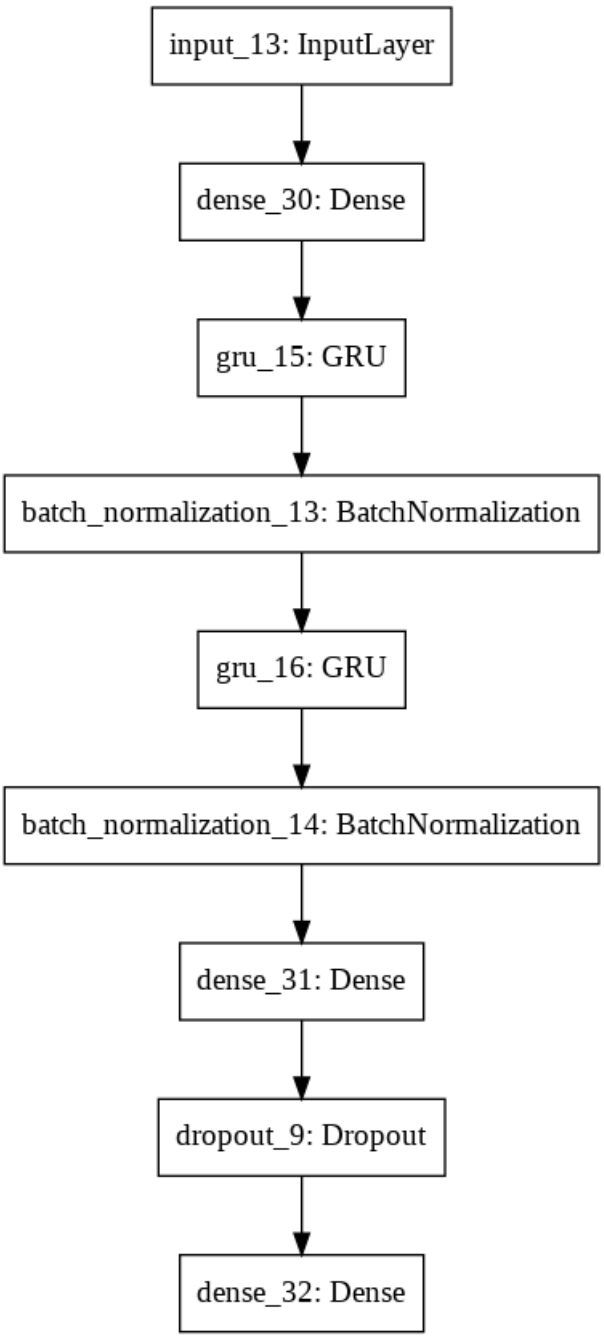
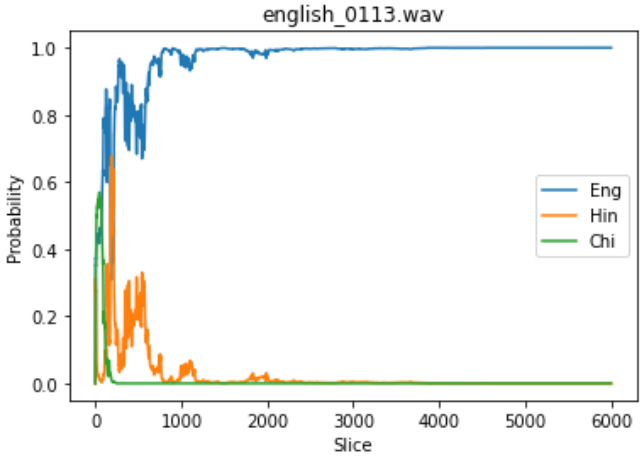
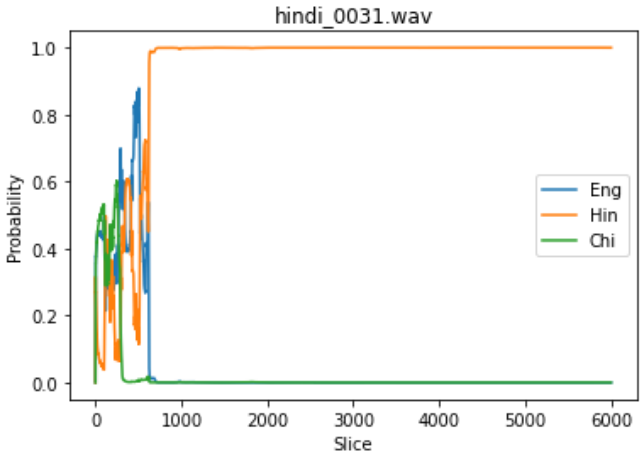


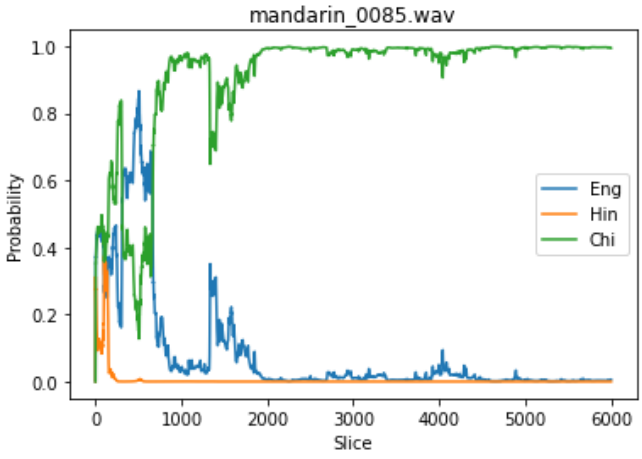
Fig. 2 Plot of the Model



(a) Test with English 0113



(b) Test with Hindi 0031



(c) Test with Mandarin 0085

Fig. 3 Plot of Testing Trained Streaming Model with Different Audio Files