Flow

1. Dataset Overview
2. Features Extraction and Preprocessing
3. Block Diagram of Model
4. LSTM Forward Propagation
5. Softmax and Cross Entropy Loss function
6. Backward Propagation in LSTM
7. Results
8. Challenges
9. Future Work

Dataset Overview

1. Tensorflow Speech Recognition Challenge - November 2017 [1].
2. It includes 65000, 1 second long utterances of 30 short words, by thousands of different people. (1800 -2000 .wav files/word)
3. labels_to_keep = ['yes', 'no', 'up', 'down', 'left', 'right', 'on', 'off', 'stop', 'go', '_background_noise_',’unknown’]. This is the subset of data, I am considering.
4. Each .wav file is of 1 second duration only,
5. So 12 class classification problem.

Dataset Statistics [2]

```
{'_background_noise_': 0,
 'down': 1,
 'go': 2,
 'left': 3,
 'no': 4,
 'off': 5,
 'on': 6,
 'right': 7,
 'stop': 8,
 'unknown': 9,
 'up': 10,
 'yes': 11}
```

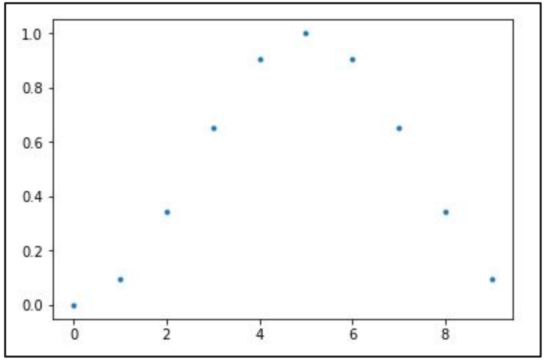
Word	Number of Utterances
Backward	1,664
Bed	2,014
Bird	2,064
Cat	2,031
Dog	2,128
Down	3,917
Eight	3,787
Five	4,052
Follow	1,579
Forward	1,557
Four	3,728
Go	3,880
Happy	2,054
House	2,113
Learn	1,575
Left	3,801
Marvin	2,100
Nine	3,934

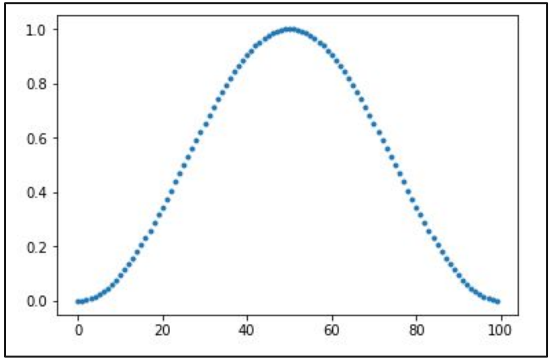
No	3,941
Off	3,745
On	3,845
One	3,890
Right	3,778
Seven	3,998
Sheila	2,022
Six	3,860
Stop	3,872
Three	3,727
Tree	1,759
Two	3,880
Up	3,723
Visual	1,592
Wow	2,123
Yes	4,044
Zero	4,052

Feature Extraction- Log Spectrogram (81 X 100)

1. Hann Window

10 Values Hann Window



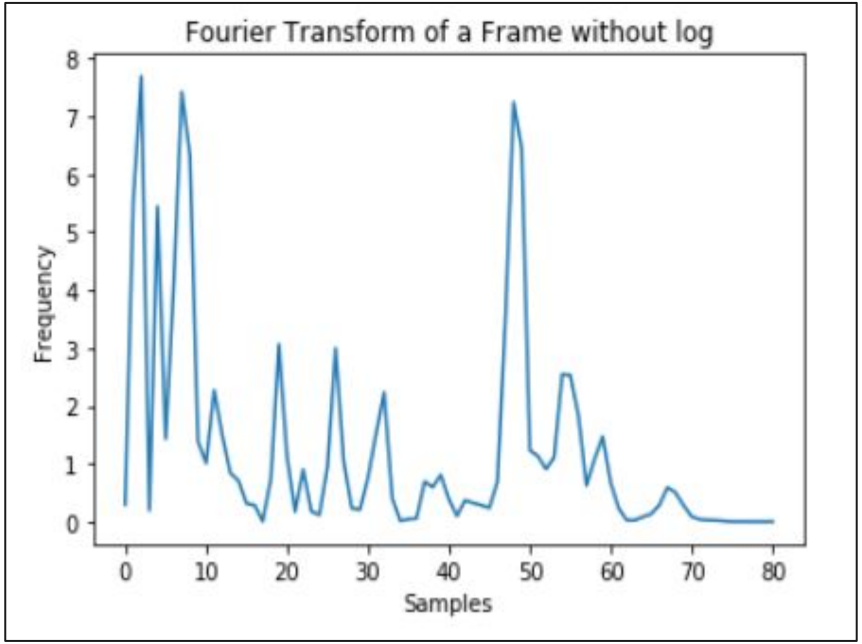


100 Values Hann Window

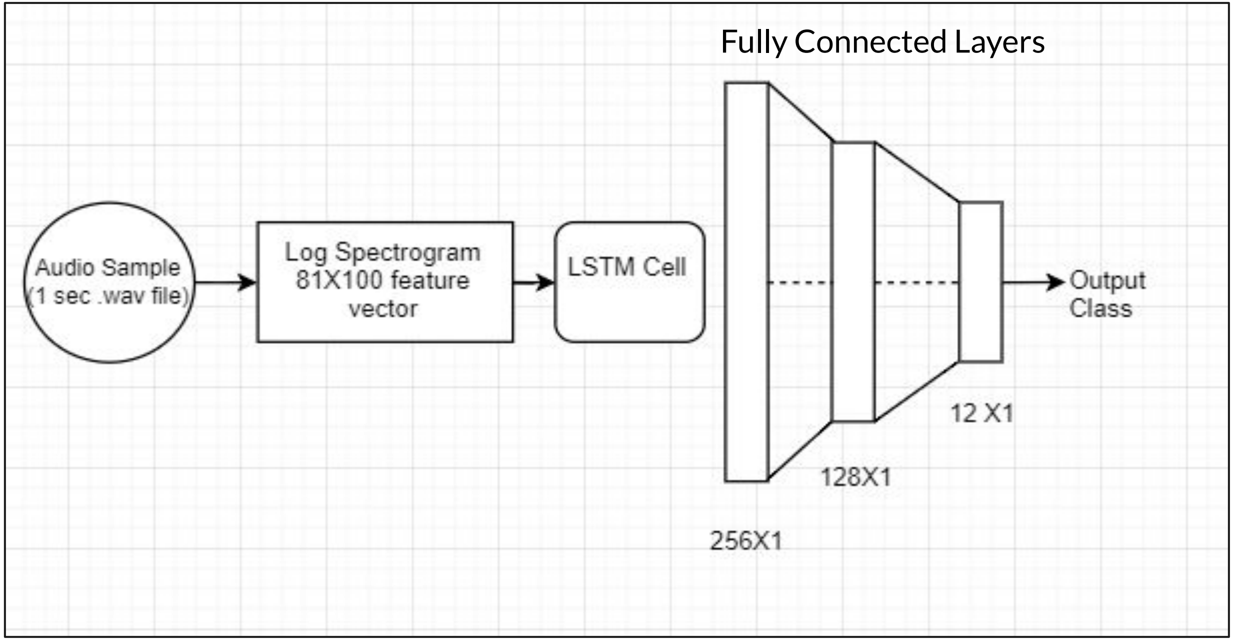
Feature Extraction

1. 1 second .wav file so 1000ms.
2. Window size taken is 10 ms, so 100 frames per audio file are there.
3. Sample rate= 16000 Hz. Each frame of 10ms contains 160 samples.
4. No overlapping of window is done.
5. Computed a log spectrogram with consecutive Fourier transforms over all frames.
6. Each frame spectrogram gives 81 values. Hence 81X100 feature vector.

Same Frame before and after taking log



Block Diagram of Model



Model (Implemented in Keras)

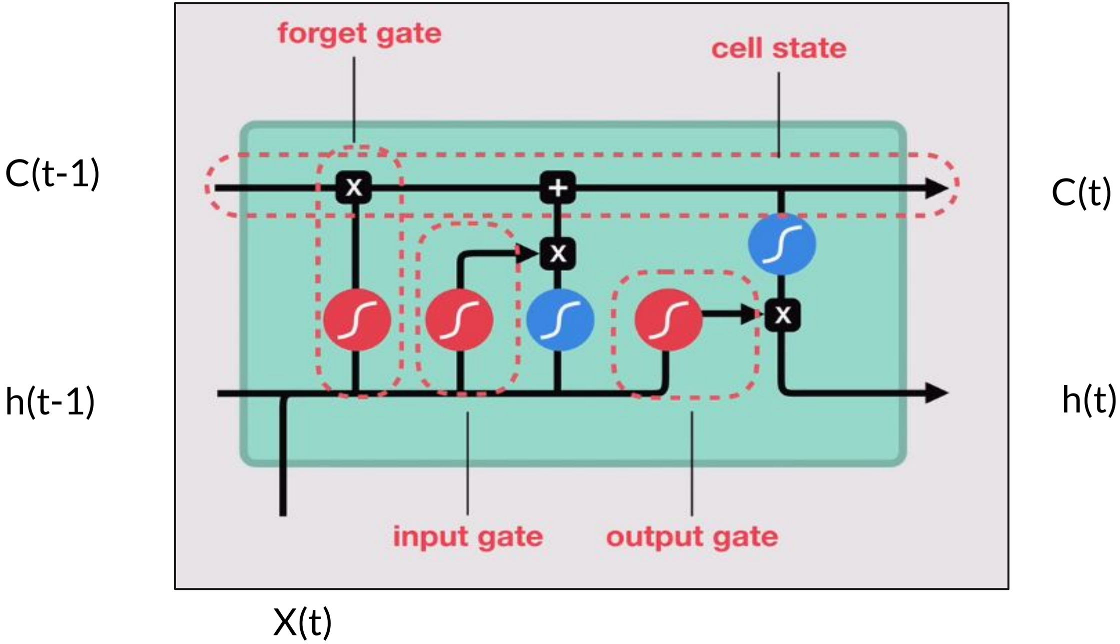
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 256)               365568
_________________________________________________________________
dense_1 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 12)                1548
=================================================================
Total params: 400,012
Trainable params: 400,012
Non-trainable params: 0
_________________________________________________________________
```

Model insights

1. Dropout Used: 0.2
2. Activation at last layer: Softmax
3. Cross Entropy Loss function is used.
4. Gradient Descent used for backpropagation.
5. Training : Testing (80:20)

```
X_train.shape  (46601, 81, 100)
X_test.shape (11651, 81, 100)
Y_train.shape (46601, 12)
Y_test.shape (11651, 12)
```

Basic LSTM cell for sequential data [3]



Activations:

Pink Circle: Sigmoid
Blue Circle: Tanh

C- Cell State
h- Hidden State

Equations inside LSTM cell [3]

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
c_t &= f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
h_t &= o_t tanh(c_t)
\end{aligned}
$$

So it can be clearly seen that weight parameters are given to each operation which will get updated during training the model. Calculating h(t) and c(t) counts for **forward propagation.**

Softmax Function at last layer

1. Given a vector of real numbers, it converts each value to a probability,
2. It gives the P(y=class |x).
3. The output node with maximum probability is assigned the corresponding class.

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}}$$

Error Function- Categorical Cross Entropy Loss

P(A) - Probability of Happening of event A

I(A)= -log(P(A)), Information Content of A

$$\text{Entropy} = -\sum_{i} P_i \operatorname{Log}_2 P_i$$

Loss Function

$$-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

O- Observed/predicted value

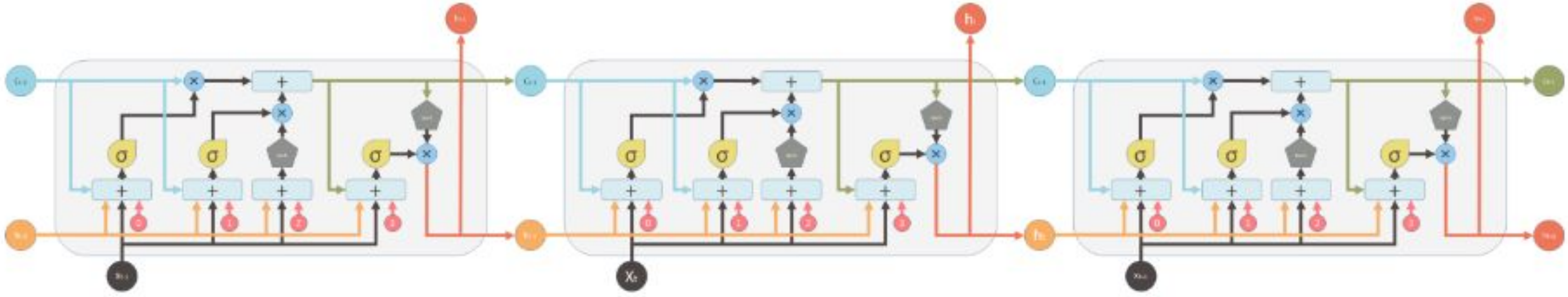
C- Class Label, M Classes

y - Binary value(0/1), 1 if correct classification

p- probability values after softmax

BPTT (Backward Propagation Through Time) [4]

Timestamp t-1 Timestamp t Timestamp t+1



A traditional approach where Average of Derivatives of Timestamps t and t+1 is taken for weight updates of LSTM at (t-1) timestamp.

Results

Comparisons	Winner (31 Classes) [6]	Paper [7]	Our Model (12 classes)
Accuracy	91.060 %	94 %	88.31 %

Challenges

1. Computation Time for feature extraction is more for bigger dataset.
2. Storage issues: Loading a big feature vector.
3. Deeper the network, greater the time complexity.
4. Deciding Hyperparameters and Model (Layers).

What is Learnt from Project?

1. Learnt about how LSTM is used for sequence to sequence learning.
2. Understood basic ML algorithms.
3. Feature Extraction from Audio.
4. Understood why ASR is a tough problem.
5. Learnt basics of Keras and Pytorch.

Future Work

1. Feature Dimension Reduction can be done. Eg. By using Autoencoder. So that complexity for feature vector processing can be minimized.
2. Simulated Annealing and other approaches can be used instead of Gradient Descent but it’s complexity is high.
3. Dataset containing audio of sentences can be taken as input for output label as text sentences and hybrid models (CNN-RNN-GRU-LSTM) can be used.

References

1. https://www.kaggle.com/c/tensorflow-speech-recognition-challenge
2. Warden. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition. arXiv: 1804.03209, Apr. 2018.
3. K. Greff, R. K. Srivastava, J. Koutn´ık, B. R. Steunebrink, and J. Schmidhuber. LSTM: A search space odyssey. CoRR, abs/1503.04069, 2015. (IEEE transaction paper)
4. A Tutorial On Backward Propagation Through Time (BPTT) In The Gated Recurrent Unit (GRU) RNN . Minchen Li Department of Computer Science , The University of British Columbia, minchenl@cs.ubc.ca
5. A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., May 2013, pp. 6645–6649.
6. Winner: Heng CherKeng, Deep Learning/Computer Vision Freelancer, Singapore
7. Jaejun Lee, Raphael Tang, and Jimmy Lin. 2018. JavaScript Convolutional Neural Networks for Keyword Spotting in the Browser: An Experimental Analysis. arXiv:1810.12859.