Speech Recognition with Deep RNNs

Project Team Members:

Niranjan Rao (171EC130)

Pruthvi Raju D R (171EC236)

Neeraj(171IT226)

Chiru Charan T S (171EC112)

Abstract

- We perform Automatic Speech Recognition using end-to-end Deep learning methods.
- One of the key challenges in sequence transduction is sequential distortions such as shrinking, stretching and translating.
- We analyze and evaluate well known Deep Speech 2 and Listen, Attend and Spell architectures
- These architectures are trained and evaluated on LibriSpeech Dataset.
- We proposed modification to both the architectures which show promising results compared to baseline.
- We obtain a WER of **10.2** and **6.7** which is an improvement over baseline models.

Methodology - Dataset and Experiments

We train our data on a subset of LibriSpeech Dataset. LibriSpeech training data consists of about 1000 hours of read audio book.

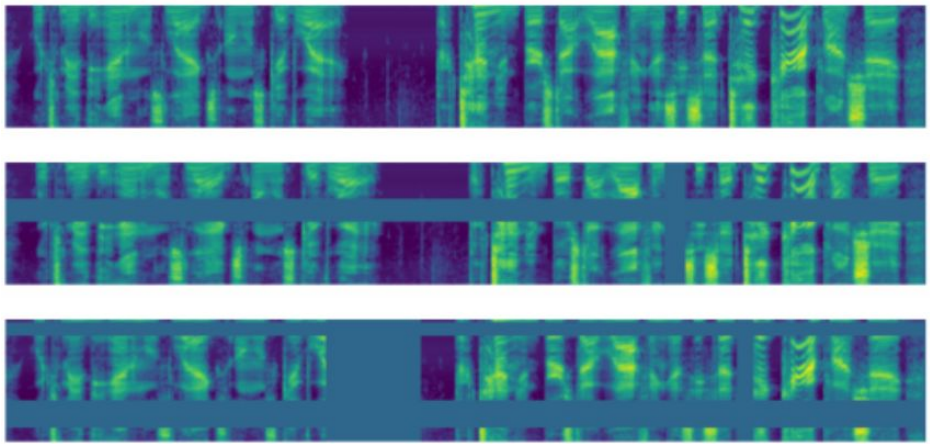
The data consists of 100 hrs of clean speech signals. Our system is tuned based on WER on the clean development set and the final optimized system is evaluated on the clean Test set.

Each sample of the dataset contains the waveform, sample rate of audio, the utterance/label, and more metadata on the sample.

subset	hours	per-spk minutes	female spkrs	male spkrs	total spkrs
dev-clean	5.4	8	20	20	40
test-clean	5.4	8	20	20	40
dev-other	5.3	10	16	17	33
test-other	5.1	10	17	16	33
train-clean-100	100.6	25	125	126	251
train-clean-360	363.6	25	439	482	921
train-other-500	496.7	30	564	602	1166

Methodology - Data Processing and Augmentation

- 128-dimensional filter bank features were computed for each input audio signal and used as the acoustic inputs to the model.
- We perform spectogram augmentation directly to the filter bank coefficients to artificially increase the diversity of our dataset.
- We find that simply cutting out random blocks of consecutive time and frequency dimensions improved the models generalization abilities significantly.
- We perform frequency and time masking as data augmentation.



Methodology - Deep Speech 2

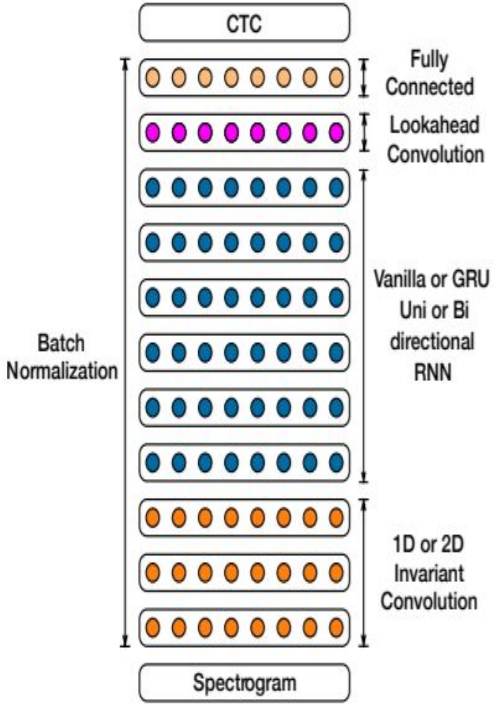
The model follows a recurrent neural network (RNN) with one or more convolutional input layers, followed by multiple recurrent layers and one fully connected layer before a softmax layer.

The network is trained end-to-end using the CTC loss function which allows us to directly predict the sequences of characters from input audio.

The outputs are the alphabet of each language. At each output time-step t, the RNN makes a prediction, is either a character in the alphabet or the blank symbol.

We use 7-layer, 5 RNN, with batch normalization and sorta grad.

Recent research has shown that BatchNorm can speed convergence of RNNs training



Methodology - Deep Speech 2

Proposed Modifications -

Our model will be similar to the Deep Speech 2 architecture. The model will have two main neural network modules -

- 3 layers of Residual Convolutional Neural Networks (ResCNN) to learn the relevant audio features,
- 5 Layers of Bidirectional GRUs (BiGRU) to leverage the learned ResCNN audio features.
- Fully connected layer used to classify characters per time step.

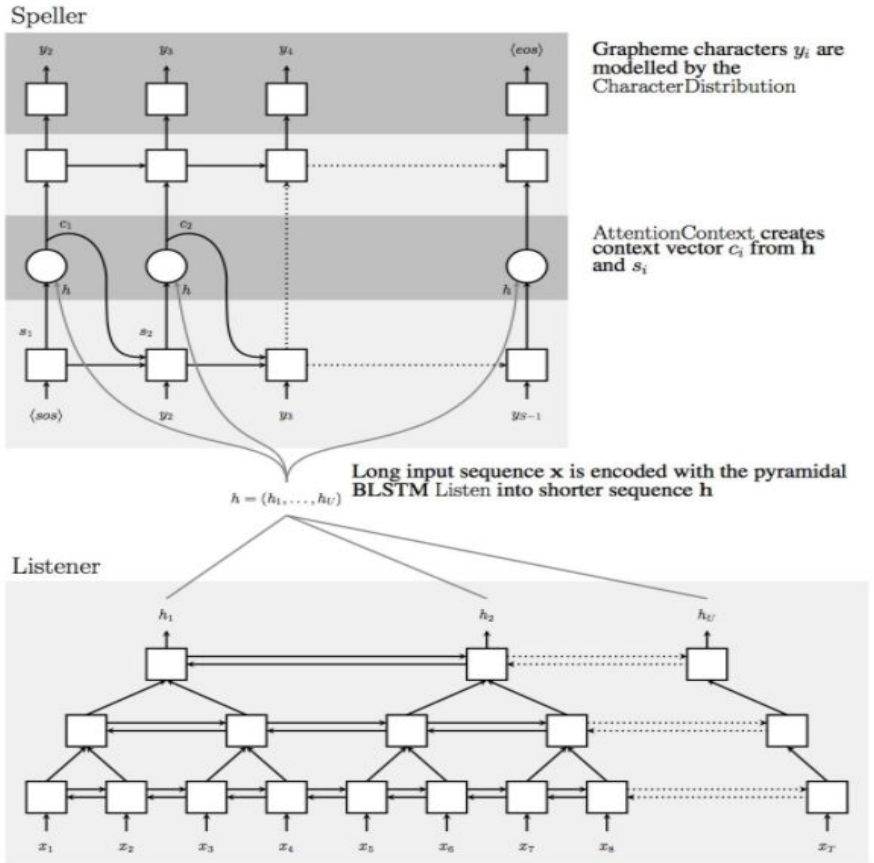
Convolutional Neural Networks (CNN) are great at extracting abstract features, and we'll apply the same feature extraction power to audio spectrograms.

Instead of just vanilla CNN layers, we choose to use Residual connections which helps the model learn faster and generalize better.

Methodology - Listen, Attend and Spell

Listen, Attend and Spell (LAS) is an encoder-decoder system.

- The encoder takes audio frames and encodes them into a denser representation.
- The decoder decodes it into the distributions of characters in each decoding step.



LAS - Listener(Encoder)

The listener is an acoustic model encoder, which takes the input speech features x , and transforms it into high level representation.

We us Bidirectional Long Short Term Memory RNN (BLSTM) with a pyramid structure.

- Applying BILSTM to speech data converges very slowly as input feature can be thousands of frames long
- We use Pyramid BLSTM (pBLSTM). In each successive stacked pBLSTM layer, we reduce the time resolution by a factor of 2.

In the pBILSTM model, we concatenate the outputs at consecutive steps of each layer before feeding it to the next layer,

$$h_i^j = BILSTM(h_{i-1}^j, h_i^{j-1})$$

BiLSTM hidden state

$$h_i^j = pBILSTM(h_{i-1}^j, [h_{2i}^{j-1}], [h_{j-1}^{2i+1}])$$

Pyramid BiLSTM hidden state

We use 40-dimension static filter-bank features. The encoder network architecture consists of 3 pyramid BLSTM layers with 1024 hidden units in each direction.

LAS - Attend and Speller(Decoder)

The function is computed using an attention-based LSTM transducer. At every output step, the transducer produces a probability distribution over the next character conditioned on all the characters seen previously.

- The speller is an attention-based decoder, which consumes *h* and produces a probability distribution of the target sequence.
- The attention mechanism allows the decoder to look at different parts of the input sequence when predicting each output.
- The speller then takes the attention context vector along with the previous prediction and generates a probability distribution over the output.

The decoder is a Transducer with a language model and 2- layer LSTM as encoder with 512 hidden units per layer.

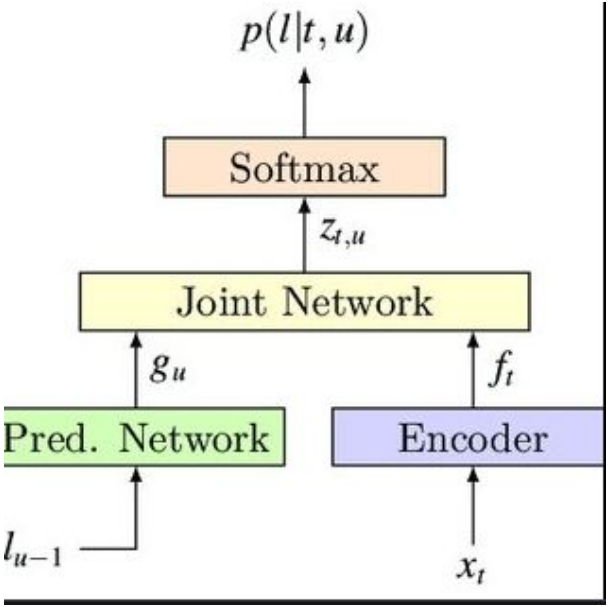
LSTM Transducer

To find the optimal word sequence, we can incorporate a language model with the encoder model.

The transducer consists of: the prediction network and the joint network.

- Predictor is autoregressive: it takes as input the previous outputs and produces feature vector that can be used for predicting the next output, like a standard language model.
- The encoder is a deep LSTM that creates a dense representation of the current input.
- The joiner is a simple feedforward network that combines the encoder vector and predictor vector and outputs a softmax over all the labels

Intuitively, we form a new deep network that makes predictions based on previous contexts and the current observations. This is the concept of RNN transducer which integrates a language model concept into the deep network.



Results - Deep Speech 2

Model Architecture	**Baseline**	**BatchNorm**
2 CNN + 3 RNN Layers	16.5	18.20
2 CNN + 5 RNN Layers	15.1	14.4
2 CNN + 7 RNN Layers	13.42	12.8
3 ResCNN + 5 BiGRU	**11.83**	**10.2**

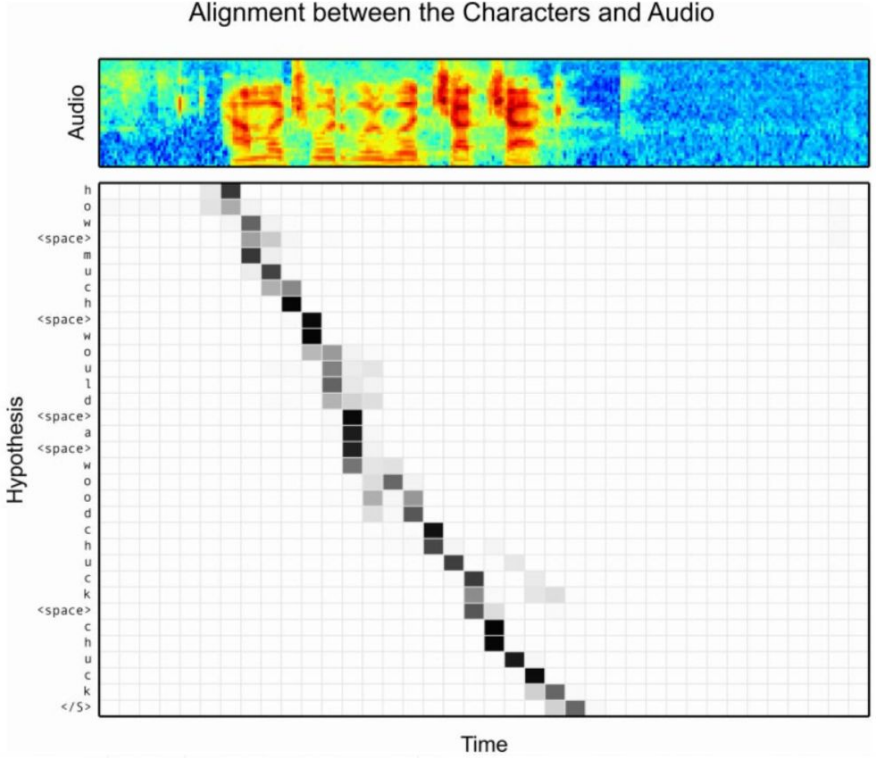
Each model is trained with BatchNorm, SortaGrad, and has 35M parameters.

Results - Listen, Attend and Spell

Baseline :

- **Encoder** - 3 Pyramid BiLSTM with 1024 hidden units
- **Decoder**:
  - Attention based RNN Transducer
  - Transducer - 2 Layer RNN with 512 units

Model Architecture	**Dev**	**Test**
Vanilla Listen, Attend, Spell	9.8	10.0
LAS - LSTM Transducer	8.1	8.8
LSTM Trans + Local Attention	7.40	7.49
SpecAug	**6.58**	**6.67**



Conclusions and Future scope

SOTA Automatic Speech Recognition models are trained on large amounts of data making using of multiple GPUs to facilitate parallel computations. Hence results of SOTA architectures are difficult to reproduce and build up on.

However we show that our proposed modifications to Deep Speech 2 and Listen, Attend and Spell models show competitive results when trained on a small subset of LibriSpeech which only contains 100hrs of data.

We hypothesize that training for more epochs using deeper architectures on 960 hours of training data will give stable and more generalizable results and show significant improvement in WER.

We can further improve our models using Transformer models which have produced significant breakthroughs in sequence to sequence NLP tasks. In this work we have not explored speech processing techniques for augmentation and cleaning data which should be able to bolster performance.

References

- Deep Speech 2 : End-to-End Speech Recognition in English
- Listen, Attend and Spell
- Attention Based Sequence to sequence model for speech recognition.
- SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition
- Sequence Transduction with Recurrent Neural Networks
- Deeply Supervised Net, in Artificial Intelligence and Statistics, 2015 Chen-Yu Lee et al.