Automatic Speech Recognition (Australian English)

Sivaramakrishnan Sankarapandian

February 3, 2018

Abstract

Automatic Speech Recognition is a widely explored field with its early systems based on Hidden Markov Models (HMMs) - Gaussian Mixture Model (GMM), but with the advent of neural networks, hybrid systems using HMM - DNN (Deep Neural Network) occupied this space. Later when the Connectionist Temporal Classification (CTC) Loss came about, end-to-end systems started to show promising results. This project is an attempt to obtain state of the art results for Australian English audio samples.

1 Introduction

Automatic Speech Recognition is the problem of determining the probability of obtaining a word given an audio sample.

$$W^* = argmax P(W|A) = \frac{argmax P(A|W)P(W)}{P(A)} \\ W^* - mostlikelyword, A - audio, W - word \tag{1}$$

Traditionally, this was done different stages. P(A|W) is modeled using Hidden Markov Models, P(W) - language model, what are the words that have high probability of occurring together, and taking argmax is the decoding stage. But with a paper on Connectionist Temporal Classification Loss which has its roots from forward-backward algorithm (which the traditional methods were following), the trend of ASR shifted towards end-to-end systems where ASR consists of a single architecture which accepts pre-processed audio samples as input and give characters as output.

2 Architecture and Preprocessing

2.1 Architecture

Since temporal dependencies in the audio signals have to be captured, RNNs/GRUs are used predominantly in the architecture. The first few layers are convolutional as well with a final layer being fully connected. CTC loss is being used for optimization which takes care of the alignment of the characters and the audio signals automatically.Usually these characters are converted into words using a language model in the end. Figure.1 shows the architecture.

2.2 Preprocessing

Though many papers have been published to use audio signals directly, use of MFCC(Mel Frequency Cepstral Coefficent) and spectograms were shown to give better results and in this case, spectrograms are used as inputs to the architecture. The process of taking spectograms from an audio is by moving a window over the audio samples (audio is considered to be invariant inside this window) and taking FFT of that resulting signal. The graph of frequency vs time forms the spectogram of the audio. Figure.2 shows an example.

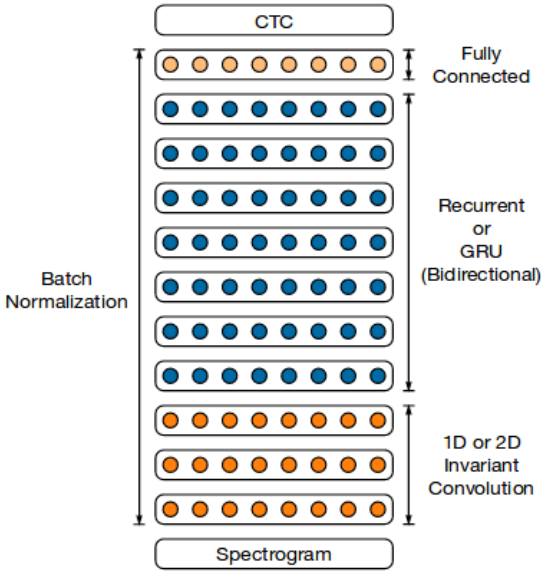

Figure 1: Architecture of Deep Speech 2

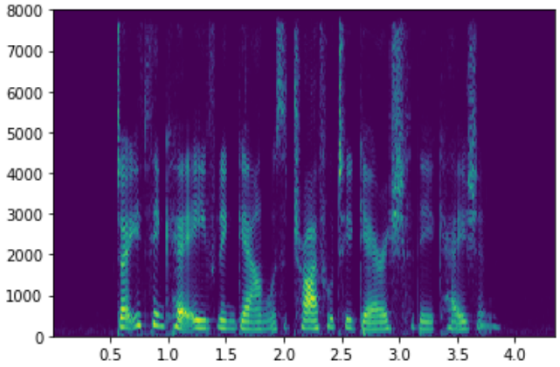

Figure 2: Spectogram of an audio sample from Aus Talk

Hyperparameter	Value
Learning rate	3e-4
Epochs	90
Annealing rate	1.01
Batch size	20
RNN layers	400
RNN units	4

Table 1: Hyperparameters used.

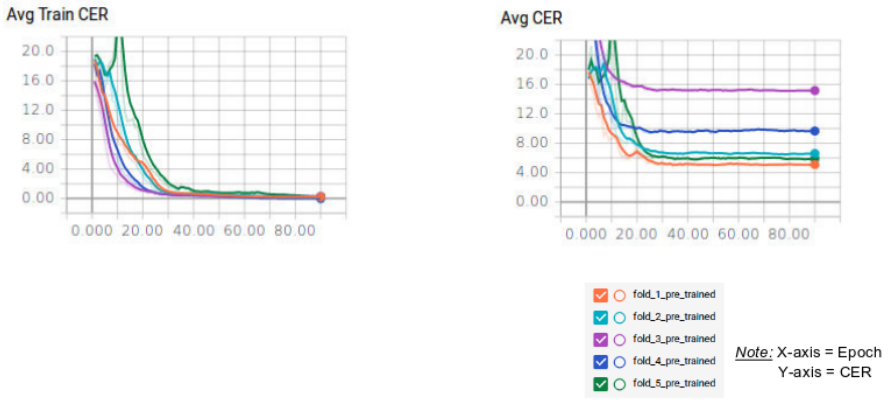

Figure 3: Character Error Rate after pruning the pre-trained model

3 Training and Results

3.1 Training

Initially, Deep Speech 2 was trained with the limited labelled dataset - AusTalk (1100 Australian English audio samples of 7 seconds each), using the hyperparameters (shown in Table.1), but later Deep Speech 2 was trained with LibriSpeech (1000 hours of annoted US english audio samples) for 45 epochs and pruned it with 1100 audio samples from AusTalk for 90 epochs. For validation 5-fold validation was used.

3.2 Metrics

Metrics used to validate the results were Character Error Rate(CER) and Word Error Rate(WER) which is the sum of added, deleted and substituted characters and words from the ground truth respectively. Lower the CER/WER the better the model.

3.3 Results

The results were recorded using Tensorboard and comparison of results before and after pre-trained has also been provided.

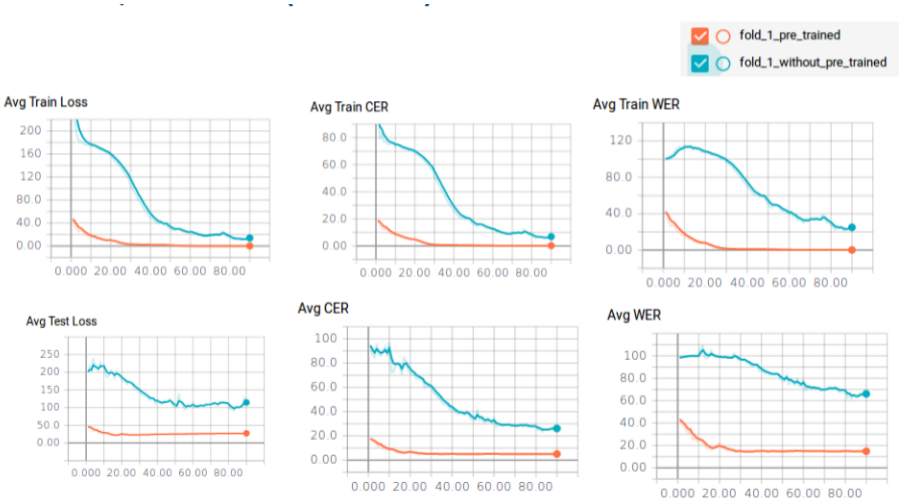

Figure 4: Train/Test loss,CER,WER for one of the folds of cross-validation