Speech Recognition Using Long Short-Term Memory Recurrent Neural Networks

Shamit Bhatia, [1] **Leuber Leuterio,** [2] **Pragnavi Ravuluri Sai Durga** [3]
Department of Computer Science, Tandon School of Engineering
New York University
6 MetroTech Center, Brooklyn, NY 11201
sb8028@nyu.edu, [1] ll4407@nyu.edu, [2] pr2370@nyu.edu [3]
https://github.com/pragnavi/Speech-Recognition

Abstract

This Speech Recognition and Classification project leverages the AudioMNIST dataset, aiming to classify speech files of spoken digits 0-9 with a high level of accuracy. Mel Frequency Cepstral Coefficients (MFCCs) were used as the digital audio representation, which subsequently served as the input into the neural networks studied. We compared the performance of a Long short-term Memory model (LSTM) and a bi-directional LSTM. The bi-directional LSTM outperformed the LSTM model with an acccuracy of 99.73% on the test data.

Introduction

Speech recognition has many significant applications that include aiding those that suffer from blindness and illiteracy, as well as facilitating human-computer interaction. The process involves listening to and analyzing audio signals. It is at the center of modern AI technology, such as virtual assistants, automatic speech recognition, and speech-to-text applications [5].

There are many different ways to represent audio signals. These include spectrograms, chromagrams, scalograms, Mel-spectrograms, and Mel Frequency Cepstral Coefficients. Audio signals are one dimensional. They consist of time series of varying amplitudes. Because neural networks require fixed dimensional inputs, it is necessary to convert/adapt raw digital audio signals into better formats which neural networks are able to process efficiently [10].

Artificial Neural Netwoks (ANNs) can categorize small acoustic-phonetic units such as separate phonemes, but they are limited in modeling long-term dependencies in acoustic signals. Recurrent Neural Networks (RNNs) combat this issue since they allow the processing of sequential data of various length. However, RNNs are limited because they look back in time for roughly ten time-steps, otherwise they suffer from the vanishing gradient problem. This has the effect of having earlier parts of the input sequencce being "forgotten." The Recurrent Neural Network with Long Short-Term Memory (LSTM-RNN) overcomes this issue through the use of gates, which allow the network to decide what information to remember [7]. The Bidirectional Recurrent Neural Network with Long Short-Term Memory (BiLSTM-RNN) can further improve performance by introducing the concept of bidirectionality [2].

Problem Statement

In word recognition, we are presented with separate recordings, where each recording consists of a person speaking a single word. Given an audio recording in .wav format, the task is to determine the word that was spoken in each of the recordings.

This is a classification task, where each class represents a word: $c \in \{1, 2, ..C\}$. In the case of Audio MNIST, each class represents a digit. For each sample-label pair, $\{(X^n, y^n)\}_{n=1}^N$, each sample is some digital representation of the audio recording. Our digital representation is a time series whose duration differs from each recording.

Related Work

Previous works demonstrated that the digital representation of audio features influence significantly the accuracy of the classification results. Turab et. al investigated the use of three different audio feature representations: Mel Spectrogram, Mel Frequency Cepstral Coefficients, and Zero Crossing Rate. They conclude that representing audio features using Mel Spectograms and MFCCs were most successful [9]. Stowell et. al report that the Mel-Frequency Cepstral Coe±cients (MFCC) remain most popular due to their computational efficiency and noise robustness. In addition, they overcome the issue of the Mel spectrogram being highly-correlated as a result of overlapping during the windowing stage. [8].

According to Lezhenin et al, LSTM networks are more efficient at learning temporal dependencies. Using Mel Spectograms as the audio representation, they examined the use of an LSTM model on the UrbanSound8K Dataset. They report that LSTMs outperform Convolutional Neural Networks (CNNs) [6]. Graves et. al compared Bidirectional LSTMs to various other neural network architectures on speech frames. They found that compared to standard RNNs and MLPs, LSTMs are much faster. Also, Bidirectional LSTM networks perform better than unidirectional LSTM networks [**?**].

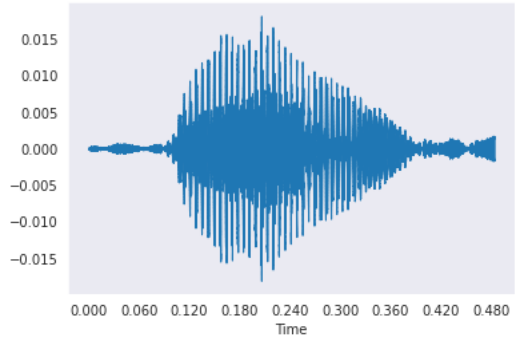

Figure 1: Raw Audio Visualization: Speaker 1, Spoken Digit 1. Each recording in the Audio MNIST data set is a raw .wav file

The model used in this study has used the MFCC features for its prediction and compares the performance of both LSTM and bidirectional LSTM networks.

Methodology

Audio MNIST Dataset

The AudioMNIST dataset1 consists of 30000 audio recordings (ca. 9.5 hours) of spoken digits (0-9) in English with 50 repetitions per digit for each of the 60 different speakers. Recordings were collected in quiet offices with a RØDE NT-USB microphone as mono channel signal at a sampling frequency of 48kHz and were saved in 16 bit integer format. Speakers' ages ranged from 22 - 61 years. 12 females and 48 males were represented from various origins. Each audio sample is recorded as a .wav file [1].

Mel-Frequency Cepstral Coefficients (MFCCs)

The raw audio files are transformed into Mel-Frequency Cepstral Coefficients using the TorchAudio transforms library. MFCCs are a collection of features that can be used to represent a section (or frame) of audio. For these experiments, we chose 39 MFCCs (40 but with the constant offset coefficient removed) and 16 kHz as the sampling rate. We chose the MFCCs as the digital representation of the speech signal since it is known to be very computationally efficient and robust to noise.

The very first MFCC, the 0th coefficient, does not convey information relevant to the overall shape of the spectrum. It only conveys a constant offset, i.e. adding a constant value to the entire spectrum. Therefore, we performed a transformation by trimming the first coefficient.

Each MFCC is then normalized by subtracting the mean of each coefficient and dividing by the standard deviation from all frames. Subsequently, each MFCC is input into the neural network.

Models

We compared the performance of both unidirctional LSTM and bidirectional LSTM models. The transformed and standardized MFCCs are fed into the each network. To account

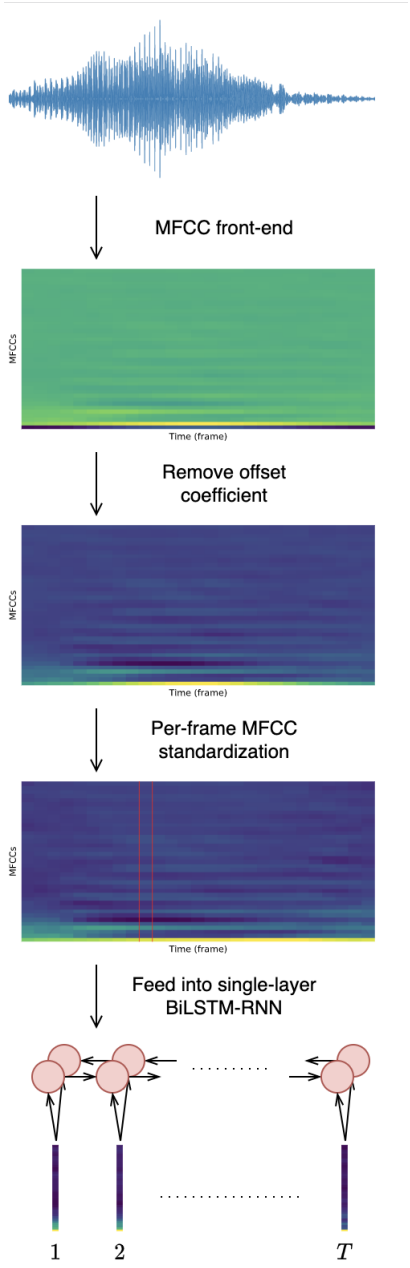

Figure 2: Preprocessing. Each raw audio .wav file is transformed. The samples are first converted into Mel-Frequency Cepstral Coefficients (MFCCs). Then, the first coeficent is trimmed and then standardized. At this point, it is now ready to be fed into the neural network.

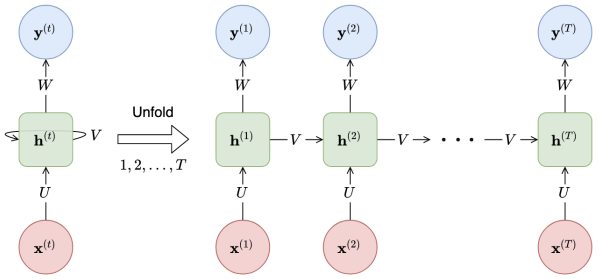

Figure 3: RNN Architecture. RNN cell unfolded to accept an input sequence with time steps. The final hidden state $h^{(T)}$ provides context for the entire sequence and is used for classification. Here, the input sequence is an MFCC consisting of T coefficient frames.

for varying MFCC lengths, the MFCC vectors are padded with 0s until the max MFCC vector length.

Standard LSTM-RNN Architecture

The LSTM model [3] is an RNN architecture, which is capable of learning complex dependencies across time. The RNN is used to accept each coefficient frame in the MFCC sequentially and then map alongside the contextual information into a latent space in a recurrent form. Furthermore, LSTM cells were incorporated into the RNN model (LSTM-RNN) to address the weakness of the RNN in learning long-term memory. LSTM RNNs address the vanishing gradient problem of basic RNNs by employing gating functions together with the state dynamics.

The main structure of LSTM consists of unique segments known as "memory blocks" in the hidden layer. The LSTM block consists of cells and the input and output gates. The "forget gate" f_t resets the cell variable leading to the 'forgetting' of the stored input c_t , whereas the input and output gates manage the reading of inputs from the feature vector, x_t , and writing of output to h_t ,respectively [4].

The LSTM model propogates in the forward direction. The last hidden state from the LSTM RNN is passed to a Dropout Layer, which is then passed to a Linear Feed Forward Neural Network. This is then followed by a ReLU Activation and another Dropout Layer. This is followed by another pass through a Linear Feed Forward Neural Network. A LogSoftmax function is finally applied to get the classification result.

Bidirectional LSTM-RNN

The architecture of the Bidirectional LSTM is identical to the standard LSTM, with the exception that it now contains two cells versus one. One cell is used to process the input sequence in the forward direction and a second cell is used to process the input sequence in the backwards direction.

In the the Bidirectional LSTM RNN, the final hidden state vector represents the final hidden states in the forward and backward directions: $h_f^{(T)}$ and $h_b^{(T)}$. As in the standard LSTM model, this is passed to a Dropout Layer, which is

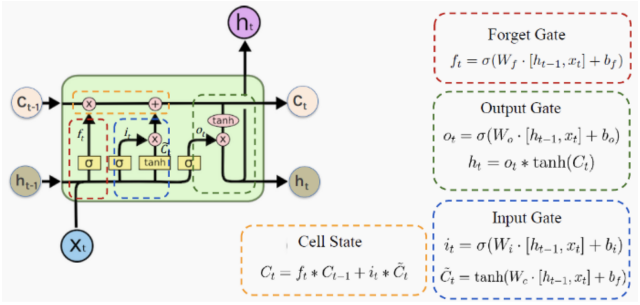

Figure 4: Standard LSTM Block

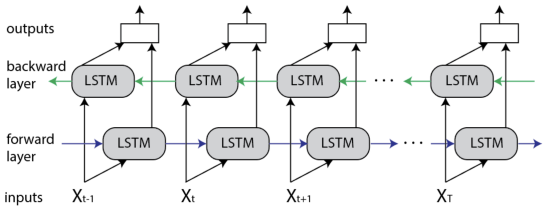

Figure 5: Bidirectional LSTM-RNN Architecture consists of LSTM blocks within an RNN framework in both the forward and backward directions.

then passed to a Linear Feed Forward Neural Network, followed by a ReLU Activation and another Dropout Layer. This is then passed through a Linear Feed Forward Neural Network. To obtain the final classification result, a LogSoftmax function is applied.

Training

The experiments in this study have been performed using the Python programming language as well as the PyTorch, TorchVision, and TorchAudio libraries. In addition, all experiments were run on the GPU. The Matplotlib and SkLearn libraries were used for visualizing the results.

The Audio MNIST dataset consists of 30,000 separate

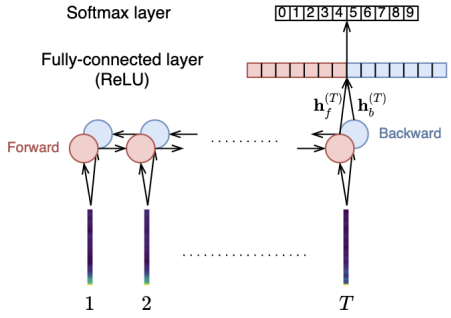

Figure 6: The final hidden state vector in Bidirectional LSTM is used for classification. Tbe vector contains information about the final hidden state in the forward and backward directions.

spoken digit recordings. This is split into the following datasets: 80 percent for training, 10 percent for validation, and 10 percent for testing.

Each raw audio .wav file underogoes the following transformations. First, the raw audio .wav file is transformed into its MFCCs using the Torch Audio library, with the number of MFCCs set to 39 and the sampling rate set to 16 kHz. The MFCC is trimmed discarding the first coefficient. This is then standardized by subtracting the mean and dividing by the standard deviation of all the frames.

The neural network receives as input the sequences of MFCC features. The sequence of MFCC features are padded with 0s until the max sequence using the "pad sequence" method available in PyTorch. The neural netowork then gives as output the class of the spoken digit.

In this case, the neural network is either the standard LSTM model or the Bidirectional LSTM model. Both the LSTM and Bidirectional LSTM models were instantiated using the LSTM module available in PyTorch. In the case of the bidirectional LSTM model, the bidirectional parameter was set to true.

The LSTM and Bidirectional LSTM networks were both instantiated with the following hyperparameters. The number of LSTM layers has been fixed to 1. The hidden size has been fixed to 50 with the with Rectified Linear Unit "ReLU" as a non-linear activation function.

The Dropout layer is used to prevent the over-fitting where the choice of which units to drop is random. Two Dropout layers are inserted after the LSTM output with a dropout probability of 0.5.

The number of input features to the output layer is fixed at 50. The number of output features is defined by the number of classes (10 classes: Digits 0-9). Using the PyTorch LogSoftmax activation function, the activation function is applied on the output layer to obtain the classification result

Hyperparameters

The training model that performs best on the validation set is kept for final evaluation on the test set.

The following hyperparameters were found to be optimal on the validation set. All models were trained for 15 epochs. The batch size was fixed at 64. The learning rate was fixed at 0.002, with ADAM used to optimize the models. The number of RNN layers was fixed at 1. The optimal number of hidden state dimensions was 50. The optimal number of units in the fully-connected layer was 50. The negative log likelihood loss was used as the loss function since this loss function is particularly useful to train classification problems with C classes.

Results

The experiments were each ran 5 times on the standard LSTM neural network as well as the Bidirectional LSTM neural network. Table 1 displays the test accuracies of these experiments, as well as the overall average test accuracy. Both models achieved promising results with test accuracies exceeding 99%. Overall, the Bidirectional LSTM model performed slightly better than the standard LSTM architecture.

Table 1: Test Accuracies for LSTM/BiLSTM

Trial	Standard LSTM	BiLSTM
1	99.67%	99.87%
2	99.50%	99.67%
3	99.53%	99.73%
4	99.43%	99.73%
5	99.63%	99.63%
Average	**99.55%**	**99.73%**

Table 2: Test Accuracies for Models and their Inputs on the Audio MNIST dataset

Model	Input	Test Accuracy
AlexNet [1]	Spectogram	95.87%
AudioNet [1]	Waveform	91.74%
LSTM	MFCCs	99.55%
BiLSTM	MFCCs	99.73%

The average test accuracy for the BiLSTM was 99.73% versus 99.55% for the LSTM model.

Table 2 displays previous reported test accuracies on the Audio MNIST dataset using various inputs. Those test accuracies are reported alongside results for models studied in this report. Becker et. al reported that using AlexNet and Spectogram as the audio representation, they achieved an average 95.87% test accuracy. They further reported that using AudioNet and Waveform as the audio representation acheived an averae 91.74% test accuracy [1]. Using the MFCCs as the input into the LSTM and BiLSTM networks is one of the factors that likely contributed to the improved test accuracies observed versus previous reported models.

Both models achieved the highest test accuracy during the first trial. The training/validation accuracy history, as well as the training/validation loss history is displayed for the best performing models. The confusion matrix on the test dataset from the best performing models is also presented.

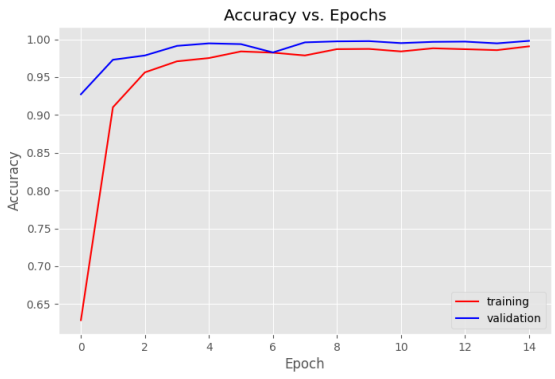

Figure 7: LSTM Training and Validation Accuracy

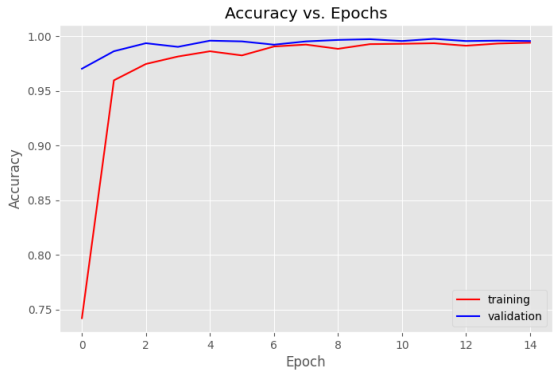

Figure 8: Bidirectional LSTM Training and Validation Accuracy

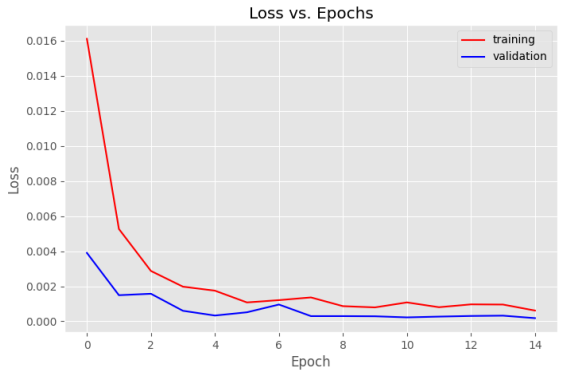

Figure 9: LSTM Training and Validation Loss

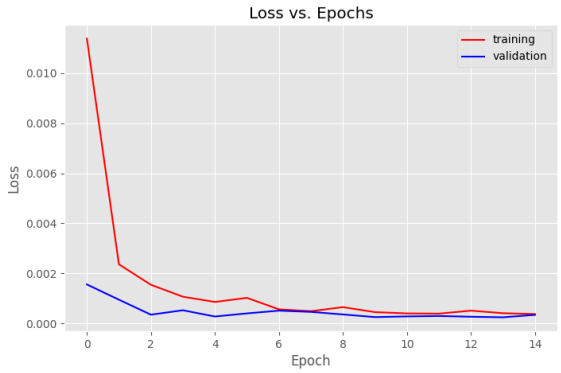

Figure 10: Bidirectional LSTM Training and Validation Loss

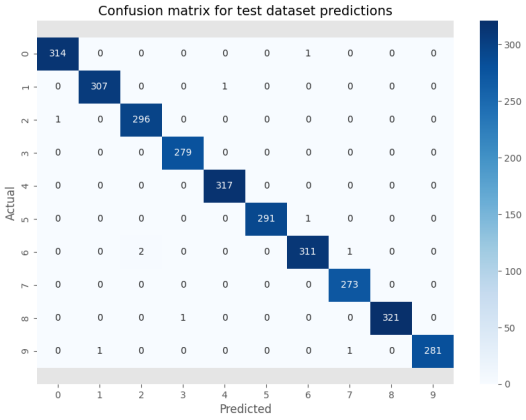

Figure 11: LSTM Confusion Matrix

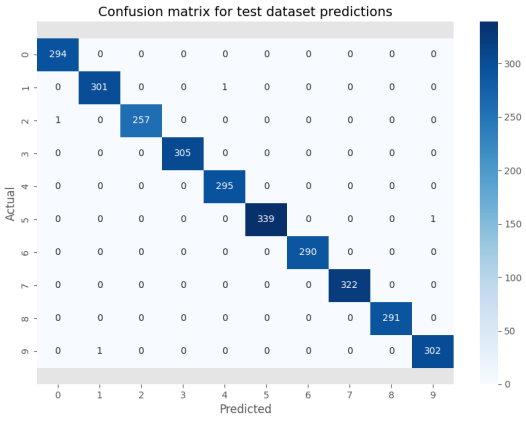

Figure 12: Bidirectional LSTM Confusion Matrix

Conclusion

In this study, the objective was to solve the speech recognition problem on the Audio MNIST dataset. An LSTM model has been proposed that utilizes Mel-Frequency Cepstral Coefficients (MFCCs) as an audio digital representation. LSTMs overcome the vanishing gradient issue that RNNs present. Bidirectional LSTMs can further improve performance on RNNs by utilizing forward and backward cells. As such, we utilized MFCCs to extract features from voice files and compared the performance of a standard LSTM architecture versus a Bidirectional LSTM model.

The model training was carried out with the use of the PyTorch/TorchVision/TorchAudio libraries in Python script, running on the GPU. The training set was based on 24000 audio samples out of a total of 30000, with the rest of the data used for validation and testing. The hyperparameters have been optimized for training accuracy optimization purposes. Both models produced promising results, but the Bidirectional LSTM model slightly outperformed the LSTM model, achieving an average test accuracy of 99.73% versus 99.55%, both with minimal loss. We further compared the use of digital audio representations that were reported with previous studied models. The LSTM and BiLSTM models using MFCCs showed a significant improvement versus AlexNet and AudioNet using spectograms and waveforms as input respectively.

Future Work

One of the shortcomings for this study was the large training time. The Audio MNIST dataset is a large dataset and it took quite a long time for the training process. Ideally, more experiment would have been carried out. Another shortcomming is that the simplicity in the Audio MNIST dataset may have likely contributed to the high test accuracies observed. This is because each .wav file contains a single utterance of a word. Any further improvements to the proposed model may require a more complex dataset in order to see a significant gain in test accuracy. Further studies can be conducted using Gate Recurrent Units (GRUs) as an alternative to the LSTM cell. Like LSTMs, GRUs allow for long-term dependencies but are more computationally efficient.

References

[1] Sören Becker, Marcel Ackermann, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Interpreting and explaining deep neural networks for classification of audio signals. *CoRR*, abs/1807.03418, 2018.

[2] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 273–278, 2013.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[5] TELUS International. What is audio classification?, Mar 2022.

[6] Iurii Lezhenin, Natalia Bogach, and Evgeny Pyshkin. Urban sound classification using long short-term memory neural network. In *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 57–60, 2019.

[7] Jane Oruh, Serestina Viriri, and Adekanmi Adegun. Long short-term memory recurrent neural network for automatic speech recognition. *IEEE Access*, 10:30069–30079, 2022.

[8] Dan Stowell and Mark D. Plumbley. Automatic large-scale classification of bird sounds is strongly improved by unsupervised feature learning. *PeerJ*, 2:e488, jul 2014.

[9] Muhammad Turab, Teerath Kumar, Malika Bendechache, and Takfarinas Saber. Investigating Multi-Feature Selection and Ensembling for Audio Classification. *arXiv e-prints*, page arXiv:2206.07511, June 2022.

[10] B. Vimal, Muthyam Surya, Darshan, V.S. Sridhar, and Asha Ashok. Mfcc based audio classification using machine learning. In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–4, 2021.