Classifying Spoken Digits Using Deep Learning

MARK WEINSTEIN
CS 3891: Deep Learning
Vanderbilt University
April 2020

Abstract

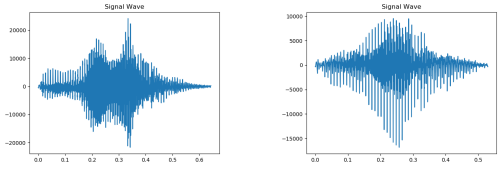
Identifying speech is useful across many domains. In this project, the Free Spoken Digit Dataset is explored and the recordings are converted to Mel-Frequency Cepstral Coefficients (MFCC) which concisely summarize audio as an "image" similarly to a spectogram but with fewer features. Then, a Convolutional Neural Network is trained on the MFCC features in order to classify the digit the speaker was saying.

The code can be found on Github: https://github.com/mweinstein91/Deep-Learning-Audio-Digit-Recognition

I. INTRODUCTION

Audio recognition is emerging as a key technology that powers some of today's smartest tools and gadgets. From Siri to Shazam, the ability for technology to recognize audio cues and then turn that insight into something actionable is immensely powerful. Audio recognition can allow more access to powerful tools to people simply by using their voice.

In Deep Learning, the MNIST digit recognition dataset [1] has been the benchmark dataset for new deep learning algorithms.The Free Spoken Digit Dataset emulates the MNIST dataset reframing the digit classification problem as audio recognition rather than handwritten images. This open source project has 4 unique speakers repeating each digit (0-9) 50 times and saved as wav files.



(a) Waveform of subject Jackson saying zero

(b) Waveform of subject Jackson saying 1

Fig. 1: The same participant saying a different digit

In this report, the recordings in the Free Spoken Digit Dataset is explored and a Neural Network (NN) is trained to detect digits based on the audio file.

II. METHODOLOGY

A. Problem

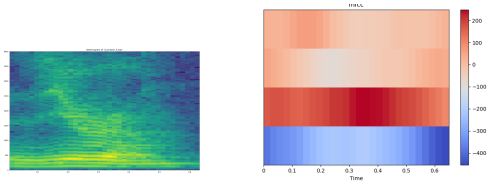
The task for this dataset is similar to the task in the original MNIST dataset: to classify a observation as one of ten digits. The only difference here is that instead of black/white images, the observations are audio files. As such, we need to carefully think about dealing with the feature space.

One of the main problems of classifying audio is finding the best way to represent the wav file as features. Audio files can be converted into images as a spectogram. A spectogram is a visual representation of the spectrum of frequencies of sound as they vary with time. However, dealing with an Spectogram audio representation creates a feature space that is unnecessarily large and leads to a NN that has too many parameters. Not all parts of the audio recording is crucial to recognizing the digit. Instead, we can extract features from each audio recording to reduce the number of parameters needed to train the model. To do this, Mel Frequency Cepstral Coefficients will be obtained from each audio file and used as the feature space.

B. Feature Transformation

As previously noted, the problem with simply creating a Spectogram and then using the image with a CNN or other image classification techniques is that it creates too large of a feature space to be explored. Instead of having a feature for every time the audio was sampled, we can reduce the feature space by trying to only discern the parts of the audio that help identify the linguistic content.

To do this the technique of Mel Frequency Cepstral Coefficients (MFCC) is used to transform each audio file. The MFCCs reduce each audio file to a smaller set of features that will give a concise "description" of the audio. MFCC works by assuming that we can frame a signal into shorter frames (as not much changing in short periods) then transform these frames several times, eventually applying the discrete cosine transform to decorrelate these frames. Luckily, we can use the a python library called Librosa that can load and perform transforms on wav files. [**?**]



(a) Spectogram **(b)** MFCC Plot

Fig. 2: Comparing a Spectogram of the full audio with the visualization of the concise MFCC for the subject Jackson saying the digit zero

C. Model Description

In order to find an architecture that learn our classification task quickly and generalize well to new observations. I considered two different architectures. The first was a Deep Feedforward network (FFN) with 2 hidden layers with relu activation function and an output layer that had a softmax activation function. This Deep Feedforward network had batch normalization and dropout [2] after each dense layer. The other was a Convolutional Neural Network (CNN) that had 2 convolutional layers and 2 dense layers. Each hidden layer had a relu activation function and a batch normalization following it, while the dense layers had dropout. Finally, the output layer had a softmax activation function.

After testing these intitial architectures, I ended up selecting the CNN architecture as it had far fewer parameters to train than the FFN due to the parameter sharing nature. The FFN required training over 3.5 million parameters whereas the CNN required only 513,000. However, I ended up not using the dropout or batch normalization as they did not help the model learn the data. As we are treating our audio files as pseudo-images for the sake of classification, the CNN approach appears to be robust[3].

D. Training Procedure

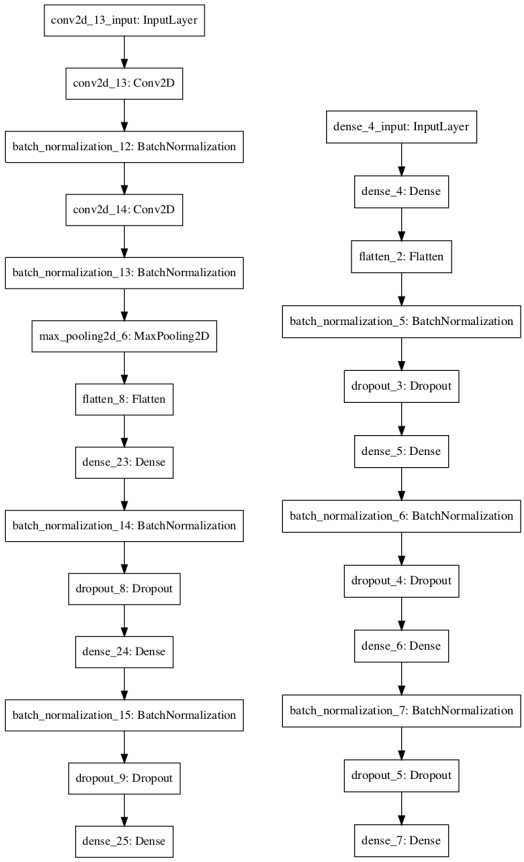
Using Keras to specify the architecture, the CNN model is trained for 50 epochs, using a minibatch size of 32 with the Adam optimizer with default parameters [4] and categorical cross entropy loss which is defined below in Equation 1.

$$J(\hat{y}_i, y_i) = -\sum_{i}^{C} y_i log(\hat{y}_i) \tag{1}$$

The validation and training loss curve are shown in Figure 4.

E. Model Evaluation

In order to evaluate the model, a simple accuracy score can be used. In this case, an accurate prediction is if the model correctly predicts the digit that the speaker was saying.



(a) Convolutional Network Proposed Architecture **(b)** Feedforward Network Potential Architecture

Fig. 3: Comparing the two potential architectures

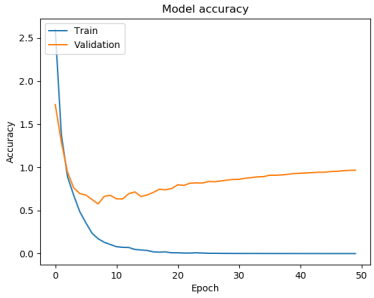


Fig. 4: Loss curves (epochs=50)

When making predictions on the test set, the model was able to correctly classify 92% of the spoken digits.

III. Conclusion

As new techniques come available in the domain of speech recognition, it is important to have datasets that can test the usefulness of new algorithms. In this report, the Free Spoken Digit Dataset was explored and shown to be very useful in benchmarking new audio algorithms.

A CNN was trained on MFCC transformed data and was fairly successful in determining the digit that was spoken in an audio example. Overall, audio recognition is an exciting field that can be greatly improved with Deep Learning.

REFERENCES

[1] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010.

[2] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[3] D. P. W. E. J. F. G. A. J. R. C. M. M. P. D. P. R. A. S. B. S. M. S. R. J. W. K. W. Shawn Hershey, Sourish Chaudhuri, "Cnn architectures for large-scale audio classification," *ICASSP 2017*, 2017.

[4] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.