Application of Convolutional Neural Networks in Accent Identification

Keven Chionh
kchionh@andrew.cmu.edu

Maoyuan Song
maoyuans@andrew.cmu.edu

Yue Yin
yyin1@andrew.cmu.edu

Abstract

The English language is spoken by many around the world, with different accents originating from various geographical locations. Accent identification is an important problem in technology today. Unfortunately, accent identification is suboptimal for certain accents. The purpose of this project is to improve the accuracy of accent identification in the English language for the Arabic, Italian, Japanese, and Korean accents.

The training data was taken from the Center for Spoken Language Understanding's Foreign Accented English v1.2 dataset. This data was used to train a convolutional neural network (CNN). In this project, we observed how the accuracy varied with the number of convolutional layers. We found that the 2-layer CNN performed best, with a training accuracy of 0.814 and a test accuracy of 0.779.

1 Introduction

1.1 Definitions

An accent is defined to be a distinct mode of pronunciation of a language, especially one associated with a particular nation, locality, or social class. In this project, we categorize accents according to their geographical origin. Naturally, not everybody in the same region speaks with the same accent. Instead, we are referring to the most widespread accent used in the region.

1.2 Motivation

'If you talk to a man in a language he understands, that goes to his head. If you talk to him in his language, that goes to his heart.' - Nelson Mandela

English is currently the most widespread language used for global communication. However, people do not speak English the same way, but instead develop different accents and dialects. As the population of English speakers grows, the 'language-gap' is a diminishing problem, but the 'accent-gap', however, becomes a more pertinent issue.

Furthermore, given the increased interest in smart appliances - Siri, Google Home, and Alexa - that respond to human voice commands, accent classification is increasingly important. It is possible that two entirely different commands spoken with different accents wind up sounding the same. As such, misclassified accents can lead to misinterpretation, which brings frustration to users of these smart appliances.

Most researches on audio recognition consider the problem rather directly: Given an audio file as the input, classify what the speaker is trying to say. We would like to propose a solution the audio recognition problem with a twist of added accent as a two-step process: Identifying the accent, and classify the content given the accent. Our project will focus on only the first step of solving this problem, on identifying and accurately labelling accents from different regions. Given an audio input of some English speech, we strive to identify from which region in the world the speaker originates.

1.3 Problem Statement

We believe that accent identification can benefit from machine learning, in particular convolutional neural networks given their relevance to sequential data. This project attempts to improve the accuracy of accent identification in spoken English.

1.4 Main Objectives

The goal of this project was to increase the accuracy of accent identification. However, there are many accents given the widespread use of the English language. Therefore we limited ourselves to the four accents that are the worst classified: Arabic, Italian, Japanese, and Korean.

We had the following main objectives:

1. Design and train a model to differentiate between the Arabic, Korean, Italian, and Japanese accents in spoken English.
2. Achieve an accuracy of at least 50% and preferably 75%. This target accuracy was decided by a literature review and a study based on human accuracy of identification.

2 Related Work

Many research papers have been published about the problem of accent classification. One of the earliest papers by Teixeira, Trancoso, and Serralheiro proposed the problem and discussed effects of utilizing parallel ergodic nets with context independent hidden Markov Model (HMM) units[1]. Their research mainly focused on identifying English accents from six different European countries. The highest accuracy they achieved was 71.7%. Then in later times, to achieve better work efficiency and classification accuracy, HMMs are often paired up with LSTMs or SVMs. In Yishan Jiao, Ming Tu, Visar Berisha, and Julie Liss's paper, they presentedan accent identification system by combining DNNs and RNNs trained on long-term and short-term features respectively. In their paper, the best accuracy that they achieved for Arabic accent was 42.35%, for Italian accent was 68.08%, for Korean accent was 47.78%, and for Japanese accent was 44.71%.

Another earlier work was done by Humphries, J. J., Woodland, P., and Pearce, D., they used a method of modeling accent-specific pronunciation variations[2]. They trained a set of HMMs on London and South East England speakers, with adapted pronunciation dictionaries for the recognition of Lancashire and Yorkshire accented speakers. Later in times, researcher are already thinking of generalizing accent identification to other languages. In 2001, Tao Chen, Chao Huang, Eric Chang and Jingchun Wang proposed a Mandarin accent identification method based on Gaussian mixture model (GMM)[3]. They explored 4 main types of Mandarin accents. For the 4 test utterances, about 11.7% and 15.5% error rate in accent classification was achieved for female and male speakers.

Based on our literature review, we set our main objective as training a model to accurately classify Arabic, Korean, Italian, and Japaneses accents in spoken English. We strive to achieve at least 50 percent accuracy for each language.

As an additional baseline, we conducted studies on 13 CMU students, none of them from the four regions we listed as our research interests. Participants were asked to listen to 12 audio fragments, 3 of each accent, from the dataset, and identify the accent. We found out that out of 156 identifications, 119 of them were correctly classified, resulting in an accuracy of 76.3%. Thus, we would prefer models that can achieve better than 75% accuracy.

3 Dataset

The audio data was taken from the Center for Spoken Language Understanding (CSLU)'s Foreign Accented English v1.2 dataset. It consisted of continuous telephone-quality speech in English by native speakers of twenty-two different languages stored in WAV format. We focused on accents with the worst accuracy rates in our literature review: Arabic, Korean, Italian, and Japanese.

4 Methodology

4.1 Overview

An overview of our methodology is as follows:

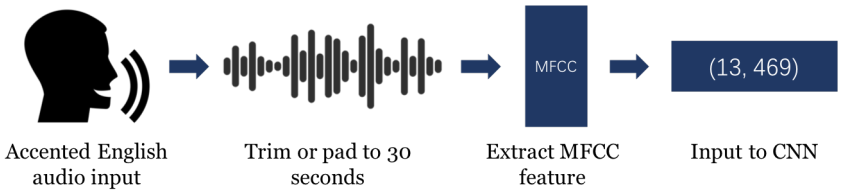

Figure 1. *Overview of methodology.*

The audio data were first preprocessed into uniform length (30 second) audio files, from which features (MFCCs, mel-frequency ceptral coefficients) were extracted. A convolutional neural network was then trained with the features.

4.2 Preprocessing audio data

The audio data were of different lengths. To standardize our inputs, we either padded (via duplication) or trimmed each audio to the same length of 30 seconds for consistency. As even after trimming, a file consists of over millions of frequencies across tons of time frames, this is far too many features as inputs for a neural network. Therefore we reduced the dimension of the audio data using the Librosa library into their mel-frequency cepstral coefficients (MFCCs), which is a representation of the short-term spectrum of sounds. This process involved breaking down the audio clip into several windows and extracting the frequency information in each window. The number of features was further reduced to 13 by discarding all except the lower 13 dimensions of the MFCCs. We chose 13 according to previous literature review that the first 13 frames correspond to the 13 frames that human ear can observe. In other words, these 13 features represented the frequencies most relevant to the human vocal range.

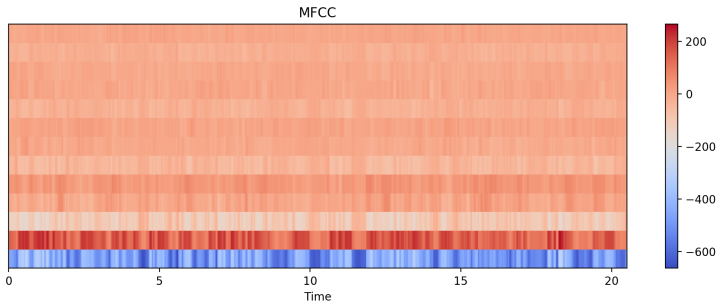

Figure 2. *Example of a MFCCs array on an audio clip. The rows represent the 13 features, whereas the colors represent the intensity of the MFCCs calculated.*

Each 30 second audio datum provided 469 sound frames. Therefore each audio sample was represented as a 13×469 MFCC array.

4.3 Training convolutional neural networks

The MFCC arrays were then used to train a convolutional neural network (CNN).

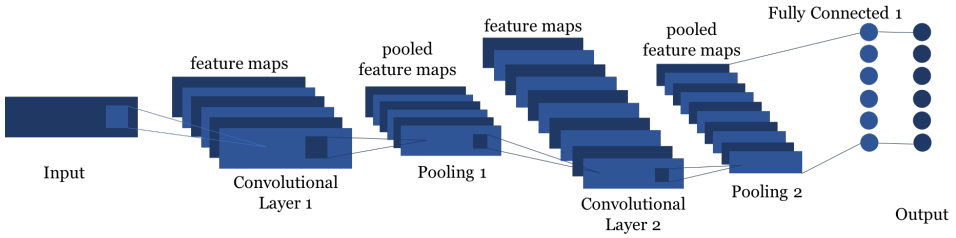

Figure 3. *Convolutional neural network trained with MFCC arrays.*

The CNN consisted of a variable number of convolutional/pooling layers, and finally a fully connected layer. We experimented using different numbers of convolutional/pooling layers to observe the different accuracies. At the same time, a dropout rate of 0.25 was used to prevent overfitting.

For the specific 2-layer CNN design shown in Figure 3, as mentioned above in Section 4.2, size of input to the neural network is 13×469. Then it goes through a convolutiona layer with 32 filters, kernel_size=[1, 50], and relu activation function to extract features. We added a pooling layer of size [2, 2] and stride 2 to reduce dimensions of output to 6×234. Then, it goes through another convolutional layer with 64 filters, kernel_size=[1, 25], and relu activation function to extract more features. This is followed by a second pooling layer.

Our model is mainly implemented with tensorflow.

4.4 Validation

The dataset was randomly split into 2 sets: 85% was put into a training set and 15% into the test set. However, the same ratio for each language was maintained within the test and training sets. This was necessary since we were not only interested in the overall accuracy of the model, but also the accuracy of the model for each accent.

5 Results

Eventually, the best model was a CNN constructed with 2 convolutional/pooling layers.We found that in general, a 3-layer CNN did not yield much improvement over a 2-layer CNN. The accuracies are plotted in the following figure:

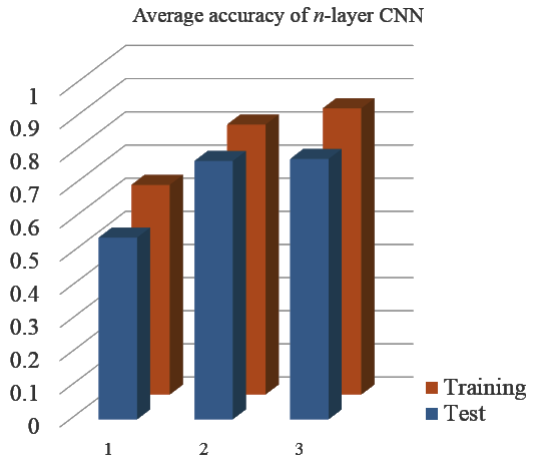

Figure 4. *The average accuracy across the 4 accents was plotted against the number of layers in the CNN.*

There was some noticeable difference in the average test accuracy between the 2 and 3-layer CNN. The 3-layer CNN had a training accuracy of 0.863 whereas that of the 2-layer CNN was 0.814. However, the difference in the average test accuracy was merely 0.785 (3-layer) and 0.779 (2-layer). In light of this insignificant difference in training accuracy, as well as increased divergence between test and training accuracy. Aside from that, 3-layers CNNs also take longer time to train. In order to saturate the nodes, it takes 2000 epochs to train a 2-layer CNN with batch size 100 but 4000 epochs to train a 3-layer one. Finally, our findings indicated that the 2-layer CNN struck a better balance between accuracy and overfitting. We therefore opted to use the simpler model that takes shorter time to train with slightly lower accuracy. However, the 1-layer CNN performed extremely poorly, with a test accuracy of only 0.548. This was far worse than any current result.

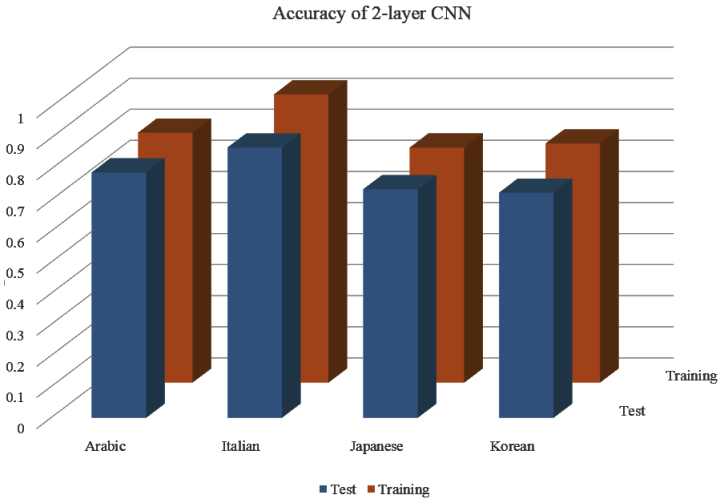

Figure 5. *Training and test accuracies categorized by accents.*

The model performed the best on the Italian accent, with a test accuracy of 0.869. On the other hand, it performed the worst on the Korean accent, with a test accuracy of 0.724. This was consistent with our literature review, which indicated that the Korean accent usually performed poorly.

6 Conclusion

6.1 Summary of results

There were four main findings:

- The 2-layer CNN struck the best balance between accuracy and model complexity.
- The Italian accent performed around 5% better on average than what we found in our literature review.
- The improvements in the Arabic and Japanese accents were less significant, and may have been due to our choice of dataset.
- The Korean accent still performed poorly, consistent with current findings.

6.2 Future Extensions

- A possible extension of this project would be to conduct further investigation on the second step of the audio recognition problem, which is to recognize the content of an audio file given the accent. This can either be done by using different models for different accents, or by attaching the classified accent as an addition feature dimension to the input vector. We expect an increase in model accuracy, and possibly a decrease in model complexity.
- A challenging extension would be adding more finesse to the model so that it will be able to identify more subtle differences and recognize accents within closer geographical proximity. For example, English spoken within Britain alone can be broken into London, Liverpool, Cockney, etc. accents.
- We can apply this logistic of model selection and training to accent identification within other popular languages. Given that mandarin has increased in popularity as a second language, it would be interesting to train a model to classify various accents of spoken mandarin. For example, a CNN could be trained to classify Cantonese mandarin and Taiwanese mandarin.

References

[1] Teixeira, Carlos & Trancoso, Isabel & Serralheiro, António. (1996). Accent identification. *The 4th International Conference on Spoken Language Processing.*

[2] Humphries, J.J. & Woodland, P.C. (1997). Using Accent-Specific Pronunciation Modelling For Improved Large Vocabulary Continuous Speech Recognition. The 5th European Conference on Speech Communication and Technology

[3] T.Chen, C.Huang, E. Chang and J.Wang, "Automatic Accent Identification Using Gaussian Mixture Models", Automatic Speech Recognition and Understanding, 2001. ASRU '01. IEEE Workshop

[4] Jiao, Yishan & Tu, Ming & Berisha, Visar & Liss, Julie. (2016). Accent Identification by Combining Deep Neural Networks and Recurrent Neural Networks Trained on Long and Short Term Features. INTERSPEECH 2016.

[5] Bower, J.M. & Beeman, D. (1995) The Book of GENESIS: Exploring Realistic Neural Models with the General Neural Simulation System. New York: TELOS/Springer–Verlag.

[6] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. Journal of Neuroscience **15***(7):5249-5262.*

[7] Sercan Arik, Adam Coate, Andrew Gibiansky, & Jonathan Raiman. (2017) Deep Voice: Real-time Neural Text-to-Speech. arXiv:1702.07825

[8] Faria A. (2006) Accent Classification for Speech Recognition. In: Renals S., Bengio S. (eds) Machine Learning for Multimodal Interaction. MLMI 2005. Lecture Notes in Computer Science, vol 3869. Springer, Berlin, Heidelberg.

[9] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), Advances in Neural Information Processing Systems 7, pp. 609–616. Cambridge, MA: MIT Press.