Noise Reduction Neural Network (NRNN)

Paris Estes, Daniel Bates, Devon Gates and Tallon Coxe

Abstract

We train a neural network on samples of whitenoise and samples of males speaking english. We then combine whitenoise and our voice audio together into one file and use the neural network to try and split the audio into a noise and denoised file. We followed closely with a research paper and code repository we found while researching this project, and used Short Time Fourier Transform (STFT) spectrum data and Mel-frequency Cepstral Coefficients (MFCCs) from the audio files to analyse and train on our data.

Data

For this project we used two different types of data. The first was the mozilla database of voices from both genders, we narrowed this data down to only male voices and used about 2000 different clips of people talking to train the model

The other data we used was a sample of a 10-hour white noise video from youtube. We split this noise into small clips to have some variety.

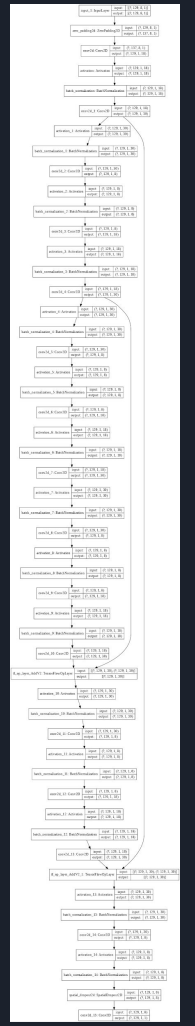
We trained the model on the human voices and then we combined the human audio with the background noise and fed that to our trained model to denoise it.

Algorithm

We used Tensorflow and Keras to build a Convolutional Neural Network for this project. The model we based ours off uses symmetric encoder-decoder architectures and a total of 33 thousand layers for the total neural network.

The projects uses a one-dimensional Cascaded Redundant Convolutional Encoder-Decoder Network (CR-CED) as discussed in this paper.

To the side is an image we generated of the major layers of our Neural Network using the plot functionality in Keras.



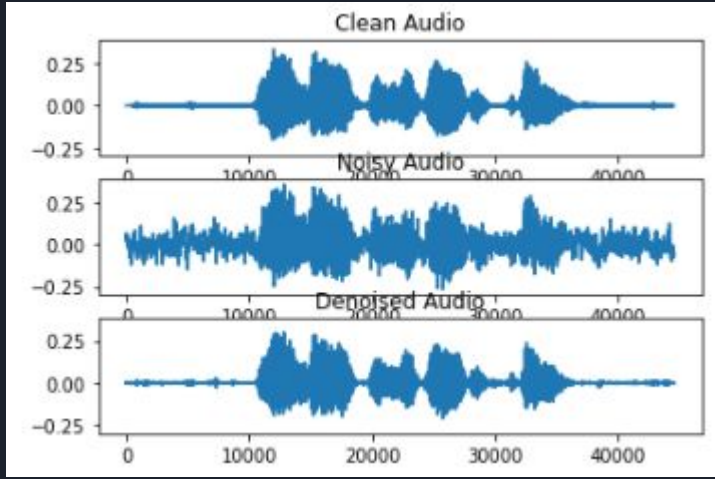
Problems

Some of the problems included.

- Learning how to turn our data into tfrecords files that could be used by the neural network
- Having to teach ourselves how to make a neural network because we didn’t go over it in class till very late.
- Finding time when we were able to meet to work on the project.
- Understanding what the problem was and how to solve it

Results

Based off of the graph we can see that the neural network does infact get rid of the noise. While it is not 100% (you can see some extra noise) it works very well. The project was tested on a few random audio files, and all worked as expected.



Conclusion

Our undertaking of the project seemed very intimidating at first, but taking a good and long look at the code base we based our project off gave us a great understanding of the process of setting up and training a neural network.

Most importantly, we learned a lot about the importance of dataset creating, and the importance of using the correct data from our dataset.

Citations

Code base that we used to create the majority of our project: https://github.com/daitan-innovation/cnn-audio-denoiser/blob/master/SpeechDenoiserCNN.ipynb

Paper on the CNN: https://medium.com/better-programming/how-to-build-a-deep-audio-de-noiser-using-tensorflow-2-0-79c1c1aea299

Paper on CR-CED: https://arxiv.org/abs/1609.07132