Singing Voice Separation using Deep Recurrent Neural Networks

Ryan Bhular

Department of Electrical and Computer Engineering, University of Rochester

ABSTRACT

Over the last few years deep learning has become dominant in the computer vision and audition fields. Online resources for computer vision are plentiful and very friendly towards people attempting to learn the content. While audio deep learning techniques share many core fundamentals with computer vision the resources are lacking, causing anyone who is trying to learn this topic to spend hours of research in order to relate other deep learning applications to audio.

One prevalent topic is audio source separation from audio mixture. This topic has numerous applications from removing noise from speech for natural language processing as well as source separation of music so that the sources can be found from any audio mixture.

Through this project a comprehensive pipeline for audio deep learning applications is explored. This may prove useful for future researchers when starting their own deep learning algorithms. We have used Keras with a TensorFlow backend, as this is quickly becoming one of the most widely used frameworks.

OBJECTIVE

The goal of this project is to successfully reimplement the Singing Voice separation of monaural recording using deep recurrent neural networks proposed by Po-Sen Huang et al [1]. In order to achieve this preprocessing of each audio file and creating a database containing training, validation, and testing datasets from these audio files.

In this report a complete guide of audio preprocessing in Python using standard and scientific libraries is explored. We will go over preparing the audio using spectral components and the correlation between audio in this domain and other deep learning applications.

Then we will explore the deep recurrent neural network architecture and break down each layer and function used in the model, as well as how to implement this in Keras. For our purposes we will simplify the model in order to only return one output from the network as the singing voice. This will make the model align with an exercise for basic recurrent neural network models and will still give us favorable results.

Finally we will be using the DSD100 dataset [3]. This dataset will provide us with a large enough dataset in order to effectively train, validate, and test the model.

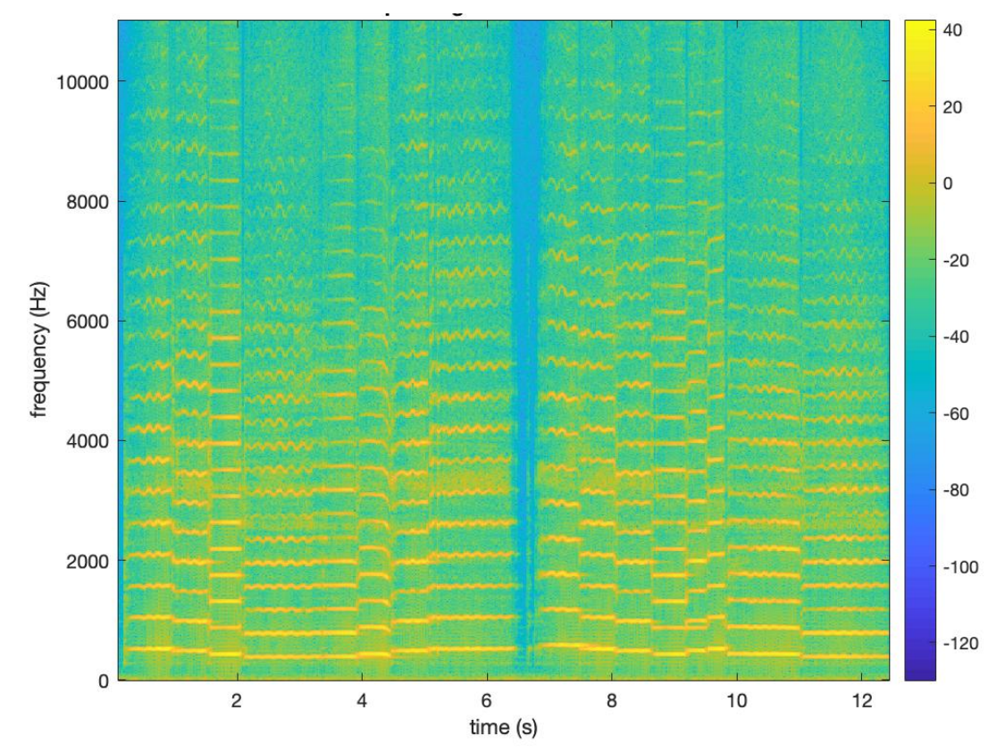
AUDIO PREPROCESSING

Short-Time Fourier Transform

The first part of preprocessing the audio data is converting it from the time domain to another feature. In these domains the features of the audio become much more noticeable and especially if you look at the intensity of each feature in small time segments of the audio.

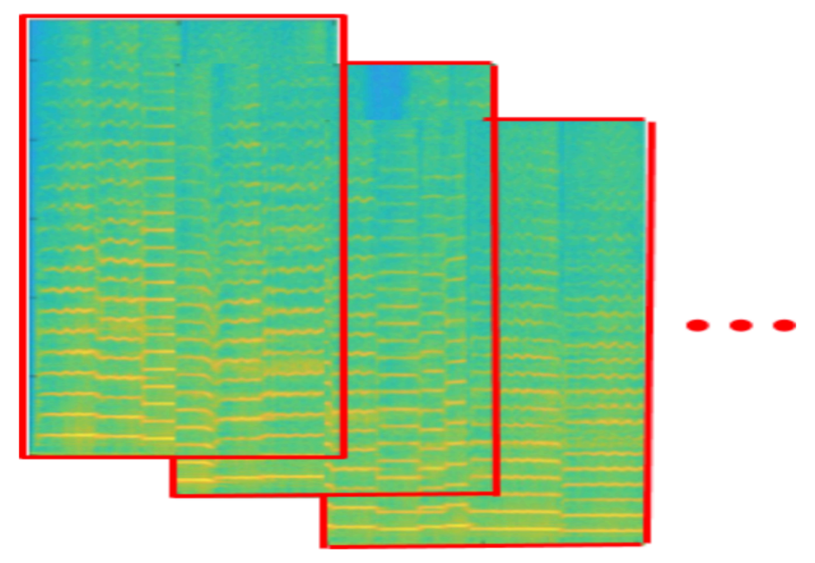
The two most common transforms that obtain these results are he Mel Frequency Cepstral Coefficients(MFCC) or the Short Time Fourier Transform(STFT). For our application we will use the STFT, though using MFCC may also work.

The STFT evaluates the frequency content of small time segments of the audio. An example of an intensity plot of an STFT is blow.



Audio Segmentation

Looking at the spectrogram data it can be easily seen that for each audio file the total number of timesteps (x dimension) will differ. For deep learning applications this poses a potential problem as the layers of the model want to take in a consistent shape for each pass. The way to work around this is to split the spectrogram into evenly sized chunks, cutting along the time axis. By doing this we can obtain three dimensional matrix for each audio file in the format: (segment or chunk, timesteps, frequency bins).



Preparing the Dataset

Using the audio segmentation technique for each audio file in the data set we can create subsets for training, validation, and testing. For our purposes we did a split of 35:15:25 (respectively) for these subsets and. We can complete all of the preprocessing of these datasets and store the data for reduced training time.

Inverse STFT

This whole process is done using the magnitude spectrum of the audio. In order to effectively convert the audio back into the time domain using the inverses short time Fourier transform we will have to reintegrate the phase information of the spectrogram.

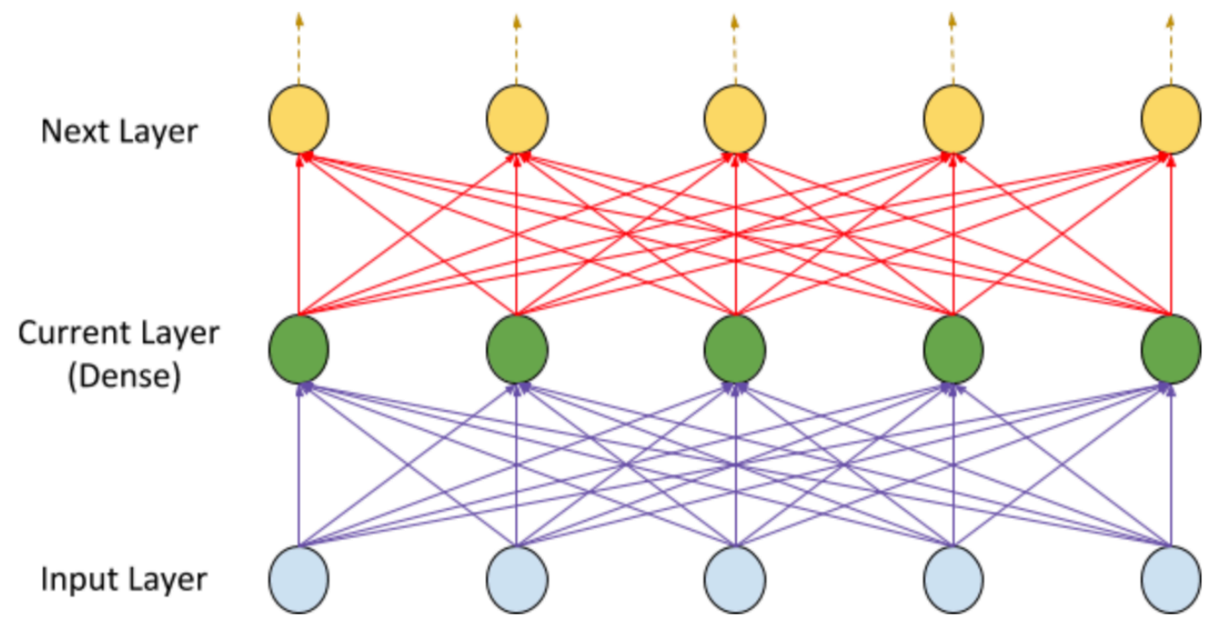
DEEP RECURRENT NEURAL NETWORKS

Deep/Fully Connected Layer

Deep learning models are comprised of connected 1-dimensional (typically) layers. These layers are comprised of tensors (or nodes) which are used to connect each layer to the next.

One of the most common layers used ids the dense or fully connected layer. This layer relies on connecting every tensor from the current layer to the next. Typically these layers are not subjected to sequential data.

For our purposes we wilsl use a dense layer as the output of our model. The input dimensionality of this will be set based on the number of frequency bins in the STFT.

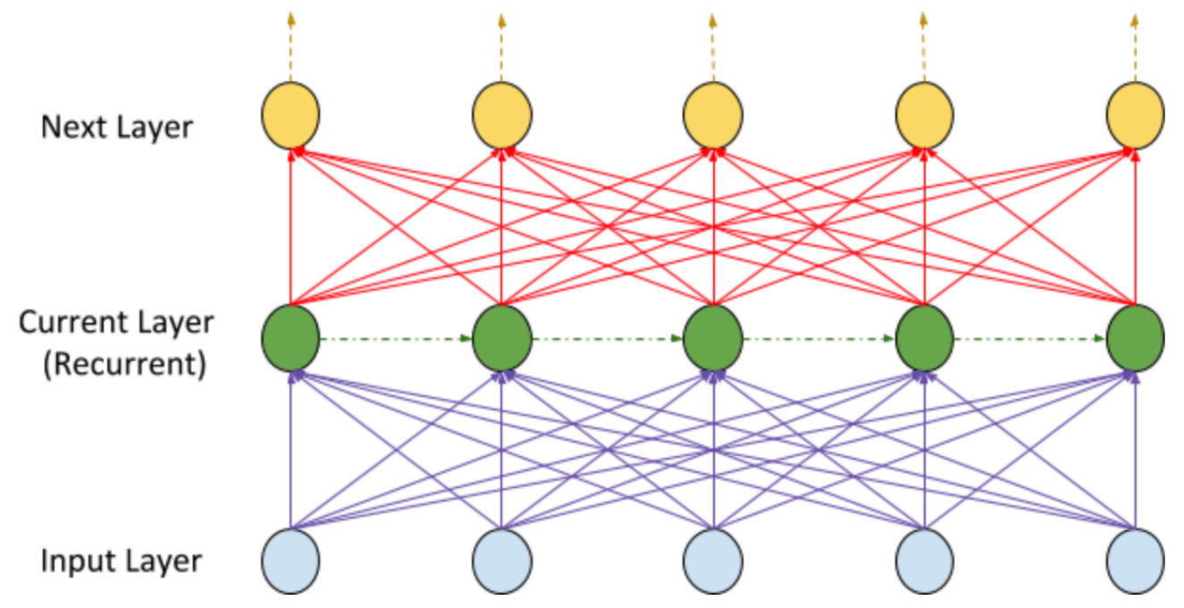


Recurrent Layers

The recurrent layer used in this approach is the Long Short Term Memory (LSTM) layer, though most of the concepts apply to a "vanilla" recurrent neural network layer.

These layers are dependent on sequential information feed into the network. This is useful for audio applications because audio is dependent on temporal features. Because of this we can assume that features from ties close together affect each other. With this we can connect the each tensor within the layer to the next as well as have a fully connected input/output from the neighboring layers.

For our purposes we will use this for the hidden layers for feature extraction of each segmented spectrogram. The input and out put dimensionality of this array is based on the number of frequency bins in you STFT.

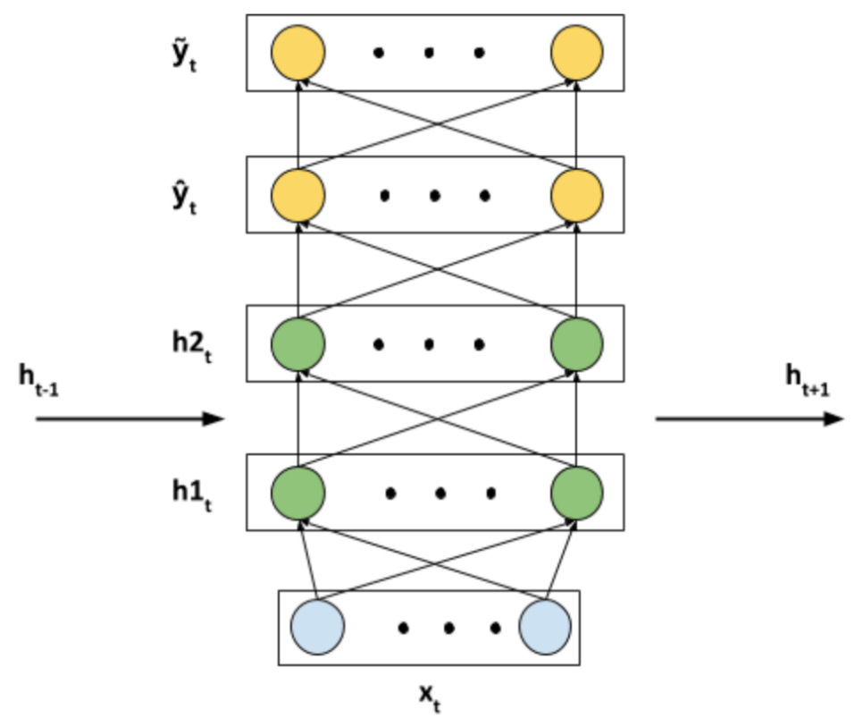


Loss (Binary Cross Entropy)

During training of a neural network model the progress of the model we must compare the output of the model with ground truth data. We do this with loss functions. As the model is trained we can monitor these values and the goal is that they converge to zero. For our application we will use binary cross entropy to monitor loss.

DRNN MODEL

As stated before this model is based on the network architecture proposed by Po-Sen Huang et al in this paper [1]. Because this is aimed be be a guide to deep learning we will instead explore a less complex model proposed by the same lab [2]. We will also remove the second output and only target one output as the separated voice. This model can be seen below.



Masking

For audio source separation a common approach is to use the network to obtain a mask. Masking is a digital image processing technique. Classically the image is multiplied element by element with a binary matrix of the same size in order to filter out any unwanted special features. A better approach is to multiply the image with a time frequency soft mask, meaning that the values in the mask range from 0 to 1 instead of being fixed to those values. This method can be used on the spectrogram of the mixture in order to separate the source (vocal) audio defined by the mask from the rest of the mixture.

NEXT STEPS

Now we must continue to train the model and validate the data. Because this project is a comprehensive guide to deep learning for audio applications in Keras the model does not have to train extensively to get perfect results. Currently the model is being adjusted to improve the performance. These results should be available in a couple days in the accompanying paper.

REFERENCES

[1] P. S. Huang, S. D. Chen, P. Smaragdis, and M. HasegawaJohnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), Mar. 2012, pp. 57– 60.

[2] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis. Deep learning for monaural speech separation. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014.

[3] Liutkus, A., St¨oter, F.R., Rafii, Z., Kitamura, D., Rivet, B., Ito, N., Ono, N., Fontecave, J.: The 2016 signal separation evaluation campaign. In: International Conference on Latent Variable Analysis and Signal Separation, Springer (2017) 323–332

[4] D. Wang, "Time-frequency masking for speech separation and its potential for hearing aid design," Trends in Amplification, vol. 12, no. 4, pp. 332–353, 2008.