Dense Hair Net: Exploration of Modern Convolutional Architectures using 1-Dimensional Sound

Eric Loreaux
Stanford University
eloreaux@stanford.edu

Yuri Zaitsev
Stanford University
yz4321@stanford.edu

Abstract

In this paper, we explore new neural network architectures for 1D convolution for the purpose of speaker recognition. The first of these architectures is meant to replace the input layer of a classical CNN model, and mimic the Fourier-like, frequency parsing capabilities of human cochlear hairs (HairNet), and the second architecture is meant to replace a classic CNN architecture altogether. This second model is a 1D implementation of DenseNet (1DenseNet) [1]. Our goal with these architectures is to improve the capabilities of 1D CNN models for raw waveform processing and make them competitive with other popular methods of waveform processing such as 2D spectrogram CNN's. The results show that our DenseNet architecture fails to outperform a classic CNN design in the speaker recognition task, and our HairNet input layer has little effect on network performance.

1 Introduction

Auditory data, like visual data, contains a large amount of contextual information that humans can process with little effort. Imbuing a computer with the ability to discern this context, however, is nontrivial. To date, one of the most popular methods for processing auditory data for learning tasks is to turn the 1D waveform into a spectrogram image for 2D convolution [2]. These spectrograms display audio data in both the time and frequency axis. While they have proven effective at displaying auditory information, there are several downsides to using 2D convolution, such as: the potential for image data to take up more memory, and the need to preprocess audio input, hindering its use on real time data. One way to reduce the complexities involved with 2D spectrogram data is to use the raw audio in a 1D convolutional network.

In this paper, we evaluate 1D convolutional networks, and attempt to boost their performance using techniques and architectures based on some of the latest convolutional neural network research. Specifically, we design a new, Inception-inspired network component (HairNet) that is intended to be attached to the beginning of any neural network [3]. We also design an entirely distinct, DenseNet-inspired network architecture (1DenseNet) that is intended to replace the standard 1D convolutional architecture [1]. Both of these designs are intended to improve our capability to parse raw audio waveforms.

2 Model

2.1 ConvNet – Control Model

We create a standard convolutional neural network (ConvNet) to serve as a control and to benchmark the performance of newer designs. The model is comprised of 10 convolutional layers, each layer consisting of the following sequence: dropout, convolution, batch normalization, ReLU, and max pooling. Dropout probability is set to 0.5, and convolutional kernel size is set to 5. Max pooling kernels have a size of 2 and a stride of 2. The first 4 layers contain 16, 32, 64, and 128 filters respectively, and the remaining 6 layers continue to be comprised of 128 filters. The output of the tenth convolutional layer is fed through a fully connected layer of size 128 and then to a softmax function with 5 possible labels to match the number of speakers. We also create a 2D version of this exact model and

run it on the same data set, however use spectrograms rather than raw audio waveforms. The purpose of this 2D version is to compare the performance of our 1D models with the current standard [2].

2.2 HairNet – Inception-style Input Layer

HairNet is an attempt to give frequency parsing capabilities of a Fourier transform for a 1D convolutional network. Intended to be placed as an input layer at the beginning of any network, HairNet transforms a single-dimensional, single-channel waveform into a multichannel representation with the same length as the original. Modeled after human cochlear hairs, whose natural frequencies cause the vibrations that allow our auditory system to carry out Fourier-like frequency analysis, a series of kernels of varying length, referred to as "hairs," are convolved with the input waveform in parallel, and combined into multiple channels. This concatenation approach takes inspiration from Szegedy et al., who also convolve with multiple filters and then join tensors channel-wise [3].

The range of hair filter lengths is designed to reflect the frequency range of human speech, which is roughly approximated from 50 Hz to 10 kHz. With an audio sampling rate of 22050 Hz, this ranges roughly equates to filter lengths between 1 - 500 sample points based on equation 1.

$$Equation\ 1\ Filter\ Size = \frac{1}{Frequency} \times Sampling\ Rate$$

Hair filter lengths are logarithmically distributed across this range. The number of these filters is the primary hyperparameter for HairNet, and we hypothesize that with increasing number of filters, the network is able to achieve higher performance faster by taking advantage of more dense collection of dedicated filters. The final result is a tensor with the same length as the original auditory vector, and a number of channels that reflect the number of hair filters used (Figure 1).

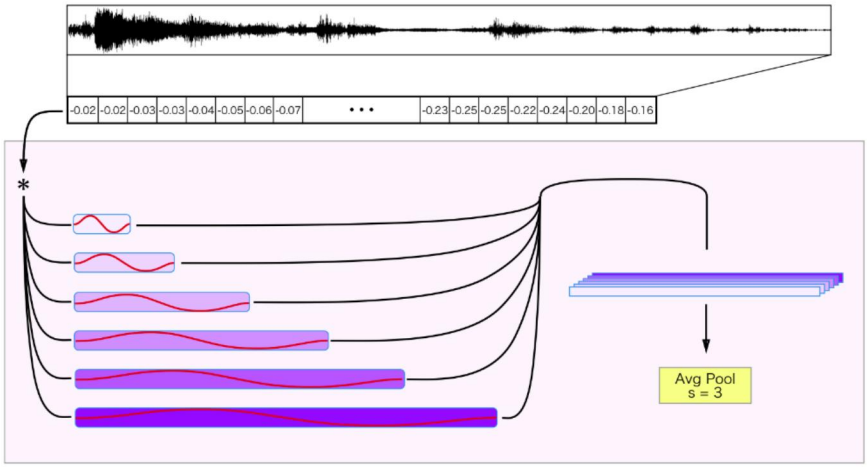


Figure 1: HairNet architecture

2.2 DenseNet Implementation

1Densenet is a network with extremely high interconnectivity. It is comprised of 7 dense blocks. Each dense block contains four convolutional sublayers whose output is propagated to all subsequent sublayers in the same dense block. This is accomplished by concatenating all outputs of previous sublayers into one input tensor for the next sublayer. Each sublayer consists of the following sequence: batch normalization, ReLU, dropout, and convolution. Dropout probability is set to 0.5, and convolutional kernel size is set to 5. Average pooling with a size and stride of 3 is used between dense blocks. The number of sublayer filters within a dense block is set to 12. These dense blocks output into two fully connected layers, one of size 384 and one of size 100, before reaching a softmax function with 5 outputs. This structure is more parameter-efficient than the ConvNet and allows early features to be propagated to later layers of the network (Figure 2) [1].

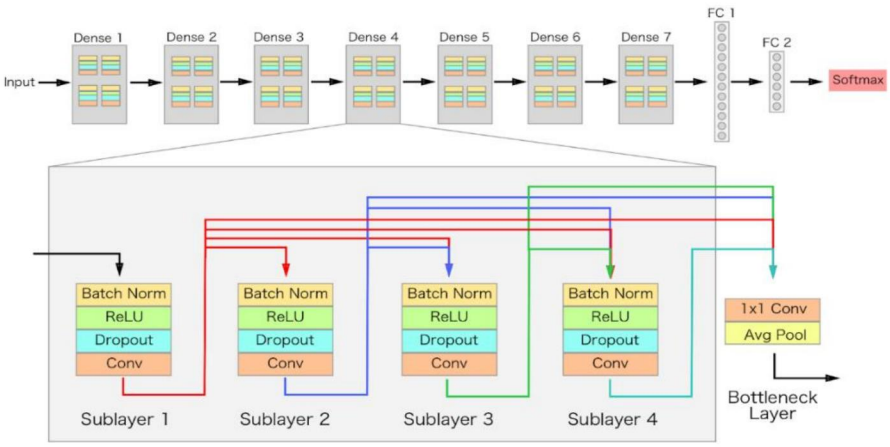
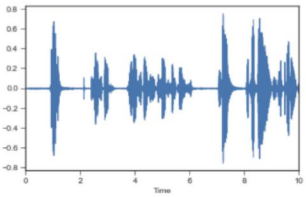


Figure 2: 1DenseNet architecture

3 Data

The architectures were evaluated using speech data collected from the previous Presidents of the United States (approximately 3 hours of data per President). This data was obtained from the American Presidency Project, which is a repository of media and other data relating to the American presidency. All data was converted into 10 second .WAV audio clips. These clips were sampled at the standard rate of 22050 Hz. The files were labeled according speaker (for example: Obama - 0, GWB - 1, Clinton - 2, GHWB - 3, Reagan - 4). The clips were then loaded as 1D, single channel, np.float arrays using the Librosa python module.

Example data point: *Obama: "Good evening. As we speak, our nation faces a multitude of challenges. At home, our top priority is to recover…" [10 second clip]*



Obama.wav	array([1.82474760e-05, 3.60148738e-06, -3.22135602e-05, ..., -1.21263368e-02, -7.64999213e-03, -4.43573901e-03], dtype=float32)		
type()	numpy.ndarray	**.shape**	(220500,)

As mentioned before, many previous works on audio tasks use spectrograms instead of 1D arrays. Spectrograms display auditory data in the frequency-time domain with an image. We convert this entire set of presidential audio recordings into spectrograms as well and feed it to a classic 2D CNN in order to directly compare the efficacy of both data types.

4 Evaluation

We run a variety of experiments in order to test the performance of these new architectures on a speaker recognition task. We compare the performance of a standard 1D convolutional network (referred to as ConvNet) with our DenseNet architecture. We also compare our HairNet input layer attached to both models with a simple convolutional kernel of fixed size. The size of this kernel is set to 20, which is the median on the log scale range of HairNet kernel sizes. For the original ConvNet, we also tune the HairNet's primary hyperparameter (number of "hairs" - discussed above), but it had little effect on performance – this data is included in the appendix.

	No HairNet	**HairNet**
ConvNet	**ConvNet**	**ConvNet w/ HairNet**
1DenseNet	**1DenseNet**	**1DenseNet w/ HairNet**

The efficiency of each architecture is measured by testing performances with different sized datasets, starting with 50 examples and moving all the way up to the full dataset of 4600 examples. For each training session, the models are run through 10 epochs of the data with a mini batch size of 16. Learning rate is set to 0.001. Adam optimization is used with the standard Adam parameter values. These four architectures are also graphically compared with the 2D ConvNet.

5 Results

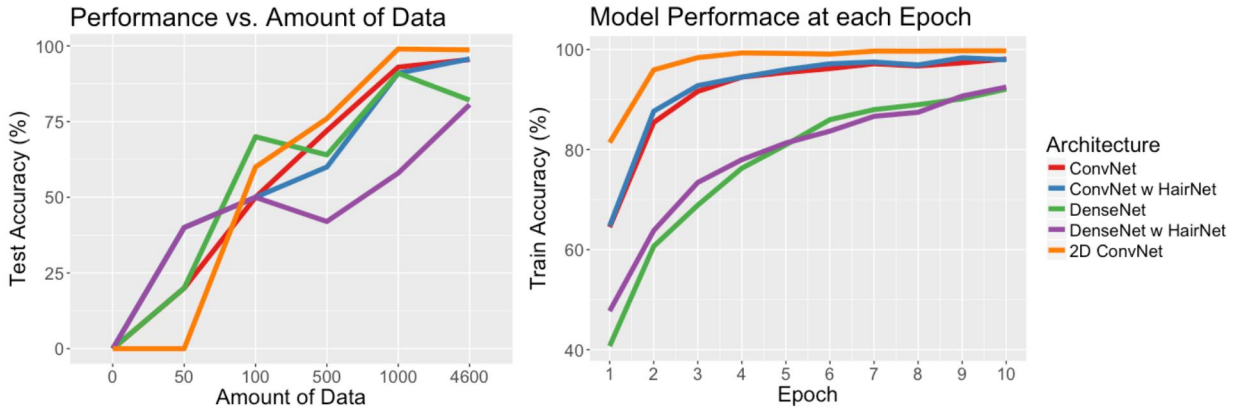


Figure 3: Performance results of the experimental architectures

Figure 3 shows the performance results on the Presidential data set. The raw data can be found in the Appendix. All models were relatively similar in their performance on smaller dataset sizes (left). The 1D networks outperformed 2D network at the smallest data size. This could be due to the stochastic nature of such a small dataset, however it also may be due to the fact that subtle features are not reinforced enough in the 2D convolution, whereas the raw waveform has some initial information advantage.

1DenseNet underperforms when compared to the standard ConvNet architecture. This could be because the number of filters present by the final layer varies significantly between the two models, with the ConvNet reaching 128 channels and the 1DenseNet only reaching 24. In addition, the HairNet input layer had no effect on model performance. In fact, due to the size of the for-loop needed to implement this layer, the amount of time needed to run HairNet models was higher and increased rapidly based on the number of hair filters used. 2D convolution still outperforms all other tested architectures, making it the current best model architecture. Figure 4 shows the confusion matrices for both the ConvNet and 1DenseNet models. The vertical axis is true label and the horizontal is prediction.

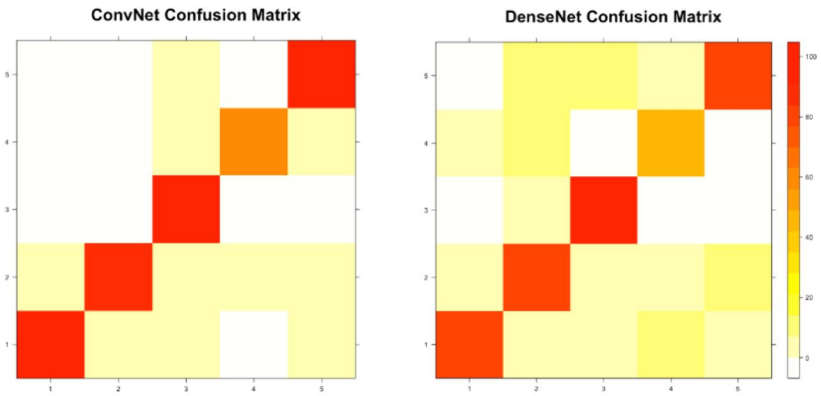


Figure 4: Confusion matrices of predicted vs. actual results. Left to right: Obama, GWB, Clinton, GHWB, Reagan

These plots help us understand the nature of the errors our models are making. While we expected these errors to be predominantly influenced by the audio and sampling quality differences between older and newer recordings, this was not the case. The most modern recordings (GWB and Obama) were the most confusing for our networks, although our 1DenseNet model did significantly worse on older recordings. Bill Clinton was the most accurately labeled president.

6 Discussion

It is with heavy hearts that we report that both of our innovative 1D convolutional network implementations fail to improve model performance when compared to 2D convolution. 1DenseNet performed significantly worse compared to the standard 1D convolutional network model, which we hypothesize to be due to a lack of comparable depth. It is, however, worth noting that the 1DenseNet model performance had not leveled off quite as much as the other models by the end of 10 epochs and could very well continue to approach the other models in performance.

HairNet also did not have an observable effect on the performance. The standard kernel we compared it to performed equally well, and it's vectorized nature makes it faster to implement. This could be because standard sized filters are free to learn any features, while HairNet may be artificially limiting the scope of possible learned features. The architectures were also trained on the CSTR VCTK Corpus. This was done to check the performance of the networks on a larger dataset that has more speakers (22 hours of data from 20 speakers) and has consistent quality audio. The networks were observed to have a high degree of overfitting.

7 Conclusion

In this paper, we describe the failed implementation of two unique convolutional architectures: HairNet, a multi-convolutional input layer modeled after human cochlear hairs, aimed at helping 1D convolutional networks perform frequency parsing; and 1DenseNet, a 1D highly connected convolutional variation of the recent DenseNet style architecture [1]. We tested these implementations on a speaker recognition task, using a dataset of about 10 hours of audio recordings for 5 presidential speakers from the American Presidency Project. 1DenseNet performed worse than the control architecture on the recognition task, and our HairNet augmentation had no effect on model performance. Spectrogram-based 2D convolutions remain the most effective way to build models for auditory tasks, possibly because there is more information contained in a spectrogram when compared to a waveform.

If we were to carry this experiment into the future, there are several avenues worth exploring. Applying these models to a more difficult task would allow for a more in-depth analysis of the strengths and weaknesses of each. This requires more work into solving the overfitting problem observed during training on the CSTR VCTK dataset. DenseNet is a method best used to create very deep neural networks, and so another promising avenue of research would be to create models much deeper than models we created for this comparison, and to continue pushing the performance of these models closer to that of 2D convolutional networks.

8 Contributions

Almost all work completed for this project was completed concurrently by both team members, who were both new to deep learning and python programming. Both team members participated in an initial literature review, an ideation phase, as well as the creation of our first neural network model. This model resembled the 2D convolutional model used as a comparison in this paper. We were unhappy with the preprocessing needed to use this model, causing a significantly deeper dive into the use of 1D convolutional networks. The work was then split in parallel, with one team member working to obtain additional data and build the data pipeline while the other team member worked on setting up the 1D convolutional network architecture and running the project on AWS. We then worked concurrently in designing the experiment, reviewing data, and drafting the report and poster presentation during longer training sessions.

7 Github

https://github.com/cs230yurieric/cs230_final_project

References

[1] G. Huang, Z. Liu, L. v. d. Maaten and K. Q. Weinberger, “Densely Connected Convolutional Networks,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 2261-2269.

[2] C. Donahue, J. McAuley, and M. Puckette, "Synthesizing Audio with Generative Adversarial Networks," arXiv:1802.04208, 2018.
doi: 10.1109/ICASSP.2017.7952190

[3] C. Szegedy et al., “Going deeper with convolutions,” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9.
doi: 10.1109/CVPR.2015.7298594

[4] J. Lee, J. Park, K. L. Kim, and J. Nam, “Sample level deep convolutional neural networks for music auto-tagging using raw waveforms,”
arXiv:1703.01789, 2017.
P. Rajpurkar, A. Hannun, M. Haghpanahi, C. Bourn, A. Ng, “Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks,”
arXiv:1707.01836, 2017.
W. Dai, C. Dai, S. Qu, J. Li and S. Das, "Very deep convolutional neural networks for raw waveforms," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, 2017, pp. 421-425.

Appendix

1D Architecture Raw Data

Data Points	**50**			
Network	**HairNet**	**Train acc [%]**	**Dev acc[%]**	**Test acc[%]**
DenseNet	yes	100.0	40.0	40.0
ConvNet	yes	97.0	60.0	40.0
DenseNet	no	97.5	60.0	20.0
ConvNet	no	100.0	60.0	20.0
Data Points	**100**			
Network	**HairNet**	**Train acc [%]**	**Dev acc[%]**	**Test acc[%]**
DenseNet	yes	100.0	60.0	50.0
ConvNet	yes	100.0	50.0	50.0
DenseNet	no	100.0	50.0	70.0
ConvNet	no	97.5	50.0	50.0
Data Points	**500**			
Network	**HairNet**	**Train acc [%]**	**Dev acc[%]**	**Test acc[%]**
DenseNet	yes	44.5	50.0	42.0
ConvNet	yes	97.0	80.0	60.0
DenseNet	no	99.3	82.0	64.0
ConvNet	no	99.0	84.0	72.0
Data Points	**1000**			
Network	**HairNet**	**Train acc [%]**	**Dev acc[%]**	**Test acc[%]**
DenseNet	yes	71.7	59.0	58.0
ConvNet	yes	98.4	83.0	91.0
DenseNet	no	99.4	94.0	91.0
ConvNet	no	98.9	90.0	93.0
Data Points	**4500**			
Network	**HairNet**	**Train acc [%]**	**Dev acc[%]**	**Test acc[%]**
DenseNet	yes	92.5	75.3	80.6
ConvNet	yes	97.9	94.0	95.7
DenseNet	no	92.1	83.8	82.1
ConvNet	no	97.3	96.6	95.5