WaveMedic: Convolutional Neural Networks for Speech Audio Enhancement

Kyle Fisher and Adam Scherlis, *Stanford University*
{kefisher, scherlis}@stanford.edu

***Abstract*—In 2016, DeepMind announced a deep neural network-based, generative model [1] of audio waveforms which led to never-before-seen text-to-speech (TTS) and speech recognition accuracy. The model, WaveNet, is a convolutional neural network framework which uses blocks of dilated causal convolutions with residual and skip connections to make learning from raw audio and generating waveforms tractable. Using the model, DeepMind was able to generate highly realistic speech from text and recognize words with unprecedented accuracy. In our project, we aim to utilize the WaveNet model toward *speech audio enhancement* by using the framework to build an application which can recover degraded speech audio. This is a novel approach to speech audio enhancement which we believe could rival other sophisticated techniques with sufficient engineering and optimization. We build a prototype of the speech enhancer called *WaveMedic* by heavily re-purposing the TensorFlow-WaveNet implementation of the WaveNet waveform generator [2], and find that our enhancer is able to learn to recover audio which has suffered a variety of degradations without ever being programmed explicitly to do so.**

I. Introduction

Although voice telephony bandwidth has been gradually increasing with time, many of todays speech transmission systems still use a highly compressed representation of voice audio data which noticeably degrades the perceived quality of the received voice signal to human listeners. Typically, these systems sacrifice details in the spectrum above approximately 4-8 KHz, resulting in lossily-compressed speech which may be considered intelligible but which requires additional listening effort. [3] Historically, one popular approach to improving the faithfulness of low-bandwidth voice codings has been the inclusion of quantized high-band formant power and phase information paired with a more detailed low-band signal (e.g. [4]). In this type of approach, the receiver decodes the formant information, resynthesizes it in the time domain, and superimposes it onto the low-band signal using techniques such as those presented in [5]. More recently, there have been efforts to improve the perceived detail in the high-band by applying artificial bandwidth extension (BWE) to narrow-band voice codings. [6] [7] Modern BWE techniques commonly integrate learning-based algorithms in which bandwidth extension is applied in a predictive manner, such that the one-to-many mapping from narrow-band to wide-band produces relatively accurate reproductions in the high-band([8]). However, there are known limits to the performance of these techniques, and certain types of unnatural artifacts are still introduced in their processes, such as frequency distortion and whistling [3] [9], so a novel approach toward post-processing narrow-band voice codings is to be desired.

In our approach to speech audio enhancement, we deemphasize the frequency domain and instead attempt to correct the raw, time-domain signals of the degraded audio. We build upon Google DeepMinds recently-released WaveNet, a convolutional neural network model originally designed as a framework for text-to-speech synthesis. In our project, we adapt WaveNets toward the purpose of reconstructing high-quality voice audio from degraded signals by training the WaveNets on synchronized clean and degraded audio waveforms, then repurpose the WaveNet generator to perform audio *enhancement*. Once trained, the input to our application is the degraded audio and the degradation type, while the output is a prediction of the original, non-degraded audio. With this prototype – which we call *WaveMedic* – we experiment with enhancing audio which has suffered from various degradations, and find that WaveMedic is capable of correcting low-frequency waveform characteristics and removing some warbling artifacts in degraded audio, but it introduces a significant amount of noise which we attribute to limited training and model sizes. In this report, we provide the details of our experiments and the results we find therein.

II. WaveNet for Audio Enhancement

A. Background: DeepMind's WaveNet (2016)

DeepMind's WaveNet is a convolutional neural network (CNN) model originally designed for TTS synthesis. However, the creators indicated that the model is much more general. In this project, adapt WaveNets to reconstruct high-quality voice audio from degraded recordings.

At its most basic, WaveNet is a generative model that operates directly on time-domain audio data. It predicts the conditional probability for the value of an audio sample in one timestep t given phonetic context $h(t)$ and the audio amplitudes in previous timesteps $(x_1, \ldots, x_{t-1})$ by

$$y_t = arg\,max_{x'_t} p(x'_t | h(t), x_1, \ldots, x_{t-1}).$$

During training, the distribution of $p(x'_t|...)$, is compared to the ground truth x_t. The error in the prediction processed and subsequently passed back to the network through backpropagation. After being sufficiently trained in this manner, the network can generate realistic speech sounds by seeding x_1 and proceeding providing the phonetic contextual

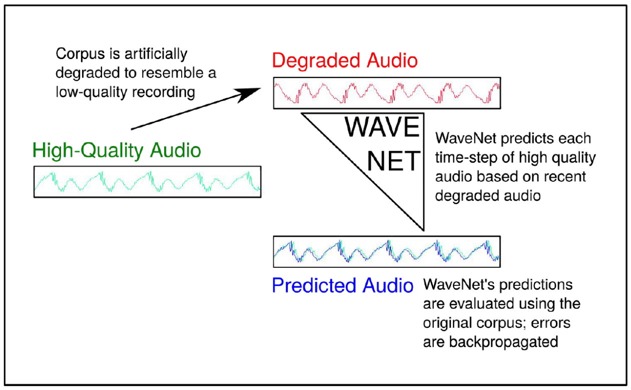
Corpus is artificially degraded to resemble a low-quality recording
Degraded Audio
WAVE NET
WaveNet predicts each time-step of high quality audio based on recent degraded audio
High-Quality Audio
Predicted Audio
WaveNet's predictions are evaluated using the original corpus; errors are backpropagated

Fig. 1. Training a WaveNet for Speech Audio Enhancement

input $h(t)$ to predict x_2, x_3, etc.

B. Adapting WaveNet for Audio Enhancement

To apply WaveNets to speech audio enhancement, we alter its training and generation scheme so that it is no longer dependent on the phonetic information $h(t)$, but instead our model learns to predict non-degraded audio provided only the degraded audio waveform. The steps of our process are as follows:

1) Preparing a Training Set: Before training our model, we create a training set which emulates the degradations which we are trying to recover from. In our experiments, we use a large corpus of high-fidelity English speech as the ground truth and render a copy of it with some destructive effect applied (e.g. clipping, low-pass filtering, etc.) as the "degraded" version.

2) Training the Model: To train our model, we provide an instance of WaveNet with a window of degraded audio and consider its output to be a prediction of the non-degraded audio at the rightmost (i.e. most latent) sample. Since at training time we have access to the non-degraded audio, we can evaluate its prediction using a loss function and use this result as a training step. That is, the loss is back-propagated into the network and the process is repeated many times until the predictions become sufficiently accurate.

3) Enhancing Degraded Audio: To enhance degraded audio, we first apply degradation to a piece of non-degraded audio to simulate it undergoing real degradation. Then, we use our trained WaveNet instance to draw predictions for each sample of what the non-degraded sound is most likely to be.

The training and enhancement steps of this process are expressed graphically in Figures 1 and 2.

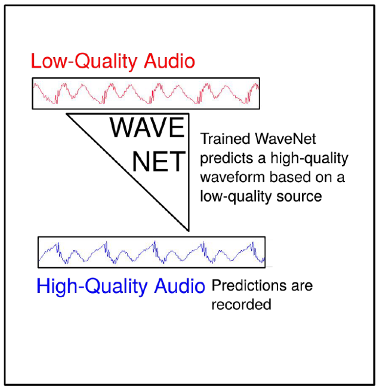
Low-Quality Audio
WAVE NET
Trained WaveNet predicts a high-quality waveform based on a low-quality source
High-Quality Audio
Predictions are recorded

Fig. 2. Enhancing Audio With a Trained WaveNet

III. Experiments

A. The Training and Test Sets

To enhance generic speech audio, it is important to train the TensorFlow-WaveNet model using corpora with high-quality recordings of a wide variety of speakers. A particularly useful corpus for this purpose is the VCTK corpus produced by The Centre for Speech Technology Research at the University of Edinburgh. [10] For the purposes of evaluating our prototype, we select the recordings of an individual speaker from VCTK to make training times more tractable. For the downsampling and MP3 experiments, we select certain recordings from within the speaker's corpus to be withheld from the training set for use as test data. For the clipping and low-pass experiments, we use test audio from a different speaker with a similar accent.

B. Creating Degradations Artificially

Since our training model uses both degraded and non-degraded versions of the inputted speech recordings, it is imperative that we select degradations which are representative of those which would be encountered in real-world applications. In many cases, degradation is the result of destructive voice codings (e.g. narrow-band encoding) or recording artifacts (e.g. clipping).

Given the technical complexity of WaveMedic, our first experimental degradation was designed as a quick "sanity-check" – a distortion function with the I/O characteristic given in Figure 3. Since we attain encouraging results with this distortion ("light clipping"), we also apply an even heavier distortion with a slightly steeper curve at the origin, which we call "heavy clipping".

We next emulate lossy data compression by applying uninterpolated downsampling to the corpus. While this is not precisely representative of actual voice compression algorithms, it serves as our initial demonstration of WaveMedic's ability to reconstruct lost data. To accomplish this, we apply a 5x downsampling following function, which serves to downsample our 48KHz input to 9.6KHz:

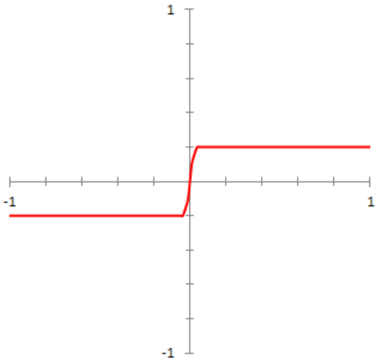

Fig. 3. Light clipping function: amplitude characteristic

$$x_t := x_i \big| i = \lfloor \frac{t-1}{5} \rfloor \cdot 5.$$

Next, we experiment with frequency distortion as an audio degradation by applying a 10th-order Butterworth filter with a cut-off frequency of 2KHz. Upon examining spectrograms produced from this degradation, we clearly see the loss of the sibilance and high-pitched resonant tones in the speaker's voice.

For our final experiment, we lossily compress the speech audio with the LAME MP3 encoder using a bit rate setting of 8 kbps, resulting in a 96x reduction in the corpus file size. At this bit rate, we find the speech to be intelligible but requiring significant listening effort. The spectrum is still very detailed, but there are some obvious warbling artifacts, and certain features of the speech (such as sibilance) are muddied.

C. Configuring WaveMedic's WaveNet Instance

WaveNets are flexible in that the number of layers and the magnitude of the dilations (see [1]) can be adjusted before training time. Additionally, we adjust the sampling rate and depth of the audio to the settings matching those of [1] such that the training and generation are more computationally feasible. For the downsampling recovery tests, we operate at the full 48 KHz to mitigate aliasing errors – otherwise, we work at 16 KHz. We find that with a NVIDIA GRID K520 GPU with 4096 MB memory supporting the TensorFlow backend, our training time ranges from 5 to 10 hours depending on the number of training steps (i.e. the number of batches processed). With this training time, our loss function – a cross entropy of the distribution of the predicted amplitude with a one-hot encoding of the ground-truth – appeared to be approaching a plateau. The reader may find it useful to know that enhancements with our prototype run at approximately 1% realtime. In summary, for our experiments, we use the following parameters:

Sample rate	16 KHz or 48 KHz
Receptive field size	3069 samples
Amplitude quantization	256 levels (via mu-law)
Training steps	5,000 or 10,000
Residual channels	32
Dilation channels	32

D. Evaluating WaveMedic's Performance

When evaluating the performance of WaveMedic, we focus on three metrics:

1) Pearson's r: *Pearson product-moment correlation coefficient*, the statistical correlation between two time-domain waveforms; equivalent to Euclidean distance between normalized waveforms. We report r^2. Higher is better.

2) LSD: *Log-Spectral Distortion* in decibels, the root-mean-square (RMS) difference between the logarithms of two power spectra. We compute this for 20ms frames and then take the RMS over time. Lower is better.

3) PESQ: *Perceptual Evaluation of Speech Quality*, a metric used in telecommunications for estimating the perceived quality of speech audio. Five-point scale (MOS-LQO). Higher is better. See [11].

IV. Results

In this section, we narrate our findings for each of the experiments individually, then summarize them in a table which lists our numerical findings for each experiment. The details of the setups for each experiment may be found in the Experiments section.

A. Recovering from Clipping

When we train WaveMedic to enhance clipped audio, we find that it performs surprisingly well in recovering the general shape of low-frequency time-domain signals. However, it could not accurately predict and reproduce the high-frequency spectrum. Figure 4 shows a slice of audio which has been artificially degraded using the light clipping function. It is then enhanced using WaveMedic, producing the waveform seen at the bottom of Figure 4. While WaveMedic has little difficulty recovering the low-frequency information of the original waveform, it is not capable of superimposing a reasonable expectation of the high frequencies in the sound, and thus it produces faint white noise of varying intensity in place of the true harmonic tones. From this experiment, we can infer that the degradation may be too severe for WaveMedic to attempt to recreate the high-frequency harmonics which were found in the original sound. We find similar results with heavy clipping, and the differences between the light and hard clipping are mostly found in the quantitative analysis from the two experiments (which we provide later in Section IV-E). We do find that the accuracy improves (as measured by r^2 and LSD), and the perceived quality (as assessed by PESQ) improves for light clipping but not heavy clipping.

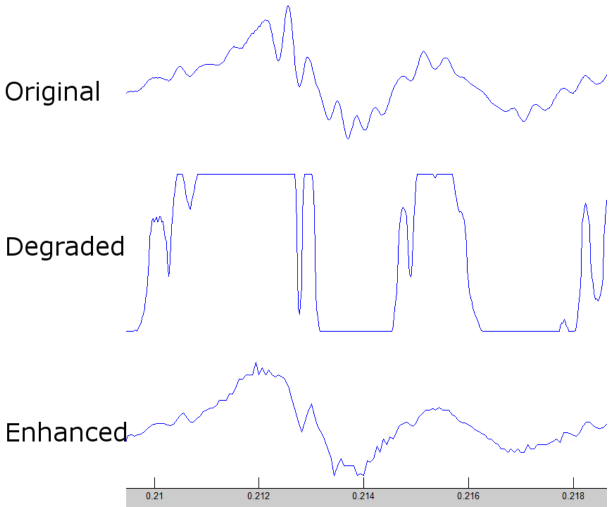

Fig. 4. Recovering from a clipping degradation

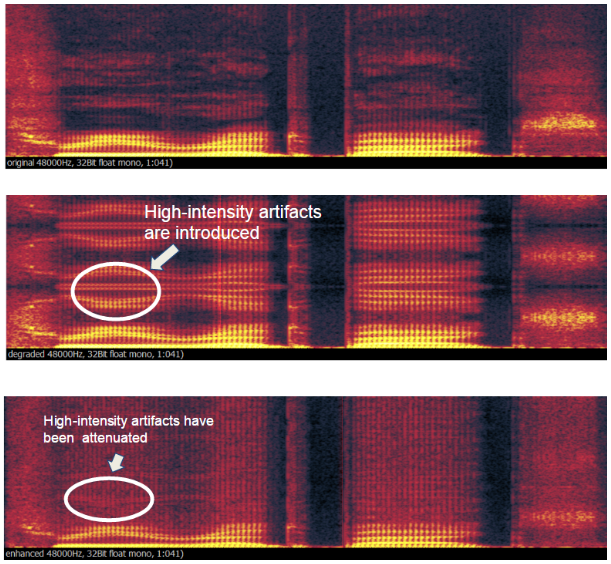

Fig. 5. Recovering from downsampling. *Top: original, middle: degraded, bottom: enhanced.* The lower end of each spectrogram corresponds to the low-frequency bands.

As a comparative reference, we also apply an off-the-shelf purpose-built declipper on the degraded sound. The selected declipper is the Audacity *Clip Fix* plug-in (see [12]), which attempts to recover the audio by interpolating audio samples which have been destroyed by the clipping. This plug-in has higher LSD and PESQ than WaveMedic, but has worse time-domain accuracy (r^2). Subjectively, WaveMedic and Audacity repair light clipping comparably well, with different kinds of artifacts present. At heavy clipping, WaveMedic's output sounds significantly better. PESQ indicates that the heavily-clipped audio is better than either recovered version, which is strongly contradicted by our subjective assessment.

B. Recovering from Downsampling

When we use WaveMedic to recover uninterpolated-downsampled audio, we find that it adds a significant amount of noise to the time-domain signal, which we believe is the result of insufficient training on a corpus which is too small with a model which is too small (i.e., there is a significant noise component in the predictions), particularly since this test ran at 48 KHz. In the spectrogram (see Figure 5), we find that WaveMedic does act to filter out many of the high-band distortions which arise from the uninterpolated downsampling, but this behavior could be reproduced with a simpler low-pass filter; WaveMedic again fails to reconstruct high-frequency structure. We do find it interesting, however, that WaveMedic learned to behave like a noisy low-pass filter (attempting to interpolate the uninterpolated samples in a stochastic dithering-like fashion).

The Pearson r^2 for the degraded recording is very high, and decreases slightly for the WaveMedic output in this test, while the LSD improves moderately. This is consistent with a small frequency-domain improvement over the degraded audio. From a subjective standpoint, we find that the WaveMedic output sounds much better than the degraded audio, but is not comparable to the original quality and still contains some buzzing.

C. Recovering from Low-Pass Filtering

In our next experiment, we investigate WaveMedic's ability to recover audio which has been degraded by low-pass filtering. The filter used is a 10th-order Butterworth filter with a cut-off frequency of 2KHz. The cut-off was chosen to remove significant voice frequencies, rather than subtle details. Quantitatively, the accuracy r^2 decreases by a greater degree than it did for downsampling, but the LSD actually indicates a massive improvement. This supports the impression that WaveMedic managed to reproduce the approximate intensity in the high-band, if not the structure. In comparison, the low-pass degraded audio is almost completely silent at high frequencies, incurring a huge LSD penalty. On the whole, our findings are that WaveMedic again struggles to estimate the spectral structure; the high frequency spectrum varied over time but was uniform in the frequency domain, i.e., white noise. This is an unexpected result, since the hallmark of DeepMind's research with WaveNets is the production of extremely realistic-sounding speech with high quality spectral properties entirely from trained WaveNets. [1] Our first inclination is to attribute this outcome to insufficient training. However, we also argue that the inclusion of preprocessed spectral information with the degraded waveforms (in the spirit of [4]) may improve the accuracy of the enhancements – we leave this as future work.

D. Recovering from Heavy MP3 Compression

Using WaveMedic to recover audio degraded by heavy MP3 compression yields more interesting results, especially in the frequency domain (see Figure 6). When clean audio is compressed into the 8-bit LAME MP3 encoding, we see that the spectral information is greatly simplified, giving rectangular blocks separated by even time steps. However, when we feed this degraded audio into WaveMedic, we see that it regains a more amorphous shape which more closely resembles that of natural speech. Again, we find that there is some white noise superimposed onto the signal, which we attribute to insufficient training. However, with close listening we find that the characteristic ”warbling” artifacts in the MP3-encoded audio are removed, a noteworthy result for WaveMedic.

The value of r^2 again remains steady from degraded to enhanced while LSD drops significantly. This is a smaller quantitative improvement compared to that of the low-pass filter experiment, but is more encouraging because the MP3 algorithm is designed to preserve spectral information, rather than discarding it.

E. Summary of Quantitative Results

In the tables below, we summarize the performance of WaveMedic in all of the experiments we performed.

In order to compare WaveMedic’s output to the original and degraded 48kHz files, we found we needed to convert the latter to 16kHz. We use two Python libraries for this, *librosa* and *scipy*. For the downsampling test we use 48kHz output and skip this step. We adjust waveforms to eliminate any relative phase shift before computing r^2.

The r^2 and LSD values differ minutely between libraries (*scipy* is shown), but PESQ gives significantly different results (even on files that sounded identical through headphones). PESQ also fails to run successfully on many of our files. We have reported the PESQ values we are able to obtain below; where two numbers are present, the first is for *librosa* and the second is for *scipy*. Where one is present, it is for *librosa*.

The best value for each metric is bolded.

Light Clipping	Degraded	WaveMedic	Audacity
Pearson r^2	0.521	**0.881**	0.572
LSD	18.9 dB	15.8 dB	**11.3 dB**
PESQ	1.22-1.72	1.98	1.33-**2.18**

Heavy Clipping	Degraded	WaveMedic	Audacity
Pearson r^2	0.391	**0.769**	0.210
LSD	21.6 dB	16.5 dB	**11.3 dB**
PESQ	1.31-**2.20**	1.19	2.13-2.15

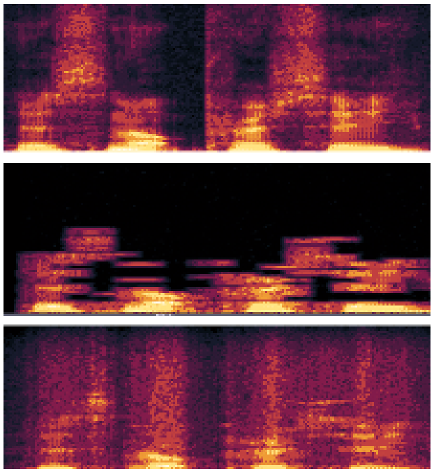

Fig. 6. Recovering from heavy MP3 compression. *Top: original, middle: degraded, bottom: enhanced.* The lower of each spectrogram corresponds to the low-frequency bands.

Downsampling	Degraded	WaveMedic
Pearson r^2	**0.980**	0.975
LSD	13.5 dB	**11.9 dB**

Low-Pass Filter	Degraded	WaveMedic
Pearson r^2	**0.962**	0.942
LSD	37.3 dB	**14.5 dB**

MP3 Encoding	Degraded	WaveMedic
Pearson r^2	**0.941**	0.938
LSD	25.4 dB	**12.6 dB**

V. Conclusion

In this project, we present our prototype, WaveMedic, a WaveNets-based speech audio enhancement tool for recovering from audio degradations. We find that WaveMedic is capable of correcting low-frequency waveform characteristics and removing some warbling artifacts in degraded audio, but it introduces a significant amount of noise which we attribute to limited training and model sizes. It can reconstruct the intensity of the highband spectrum, but not its shape. Our prototype performs better than an existing audio declipping tool by our subjective assessment; quantitative metrics were more mixed. It is noteworthy that WaveMedic is a more versatile tool than traditional purpose-built enhancers, since is capable of learning to recover arbitrary degradations. In future experiments, we hope to see improved performance of WaveMedic when we add contextual data (such as spectral information) and increase the computing resources available during training time.

References

[1] A. van den Oord *et al.*, “Wavenet: A generative model for raw audio,” *CoRR*, vol. abs/1609.03499, 2016.

[2] I. Babuschkin, “A tensorflow implementation of deepmind’s wavenet paper,” 2016. [Online]. Available: https://github.com/ibab/tensorflow-wavenet

[3] A. E. Rosenberg, R. W. Schafer, and L. R. Rabiner, “An investigation of the effects of smoothing and quantization of the parameters of formant-coded speech.” *Journal of the Acoustical Society of America*, vol. 49, no. 1A, p. 123, 1971.

[4] E. W. Yu, “Harmonic+noise coding using improved v/uv mixing and efficient spectral quantization,” *International Conference on Acoustics Speech and Signal Processing*, pp. 477–480, 1999.

[5] D. G. Childers, *Speech processing and synthesis toolboxes. D.G. Childers.* New York : Wiley, [2000], 2000.

[6] J. Kang, “Adaptive speech streaming based on speech quality estimation and artificial bandwidth extension for voice over wireless multimedia sensor networks,” *International Journal of Distributed Sensor Networks*, 2015.

[7] S. L. J. Berisha, V. Sandoval, “Bandwidth extension of speech using perceptual criteria,” *Synthesis Lectures on Algorithms and Software in Engineering*, vol. 5, no. 2, pp. 1–83, 2013.

[8] S. Yao, “Block-based bandwidth extension of narrowband speech signal by using cdhmm,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2005.

[9] P. Jax, “An upper bound on the quality of artificial bandwidth extension,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2002.

[10] J. Yamagishi, “English multi-speaker corpus for cstr voice cloning toolkit,” 2012.

[11] “Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs,” 2000. [Online]. Available: http://www.itu.int/rec/T-REC-P.862/en

[12] “Audacity: Free, open source, cross-platform audio software for multi-track recording and editing.” [Online]. Available: http://www.audacityteam.org/