Audio Compression using Neural Networks

Yeskendir Koishekenov

KAIST, 2018

Contents

Abstract

Traditional speech and music compression algorithms rely on encoder/codec pairs (codecs) that are designed to achieve specific tasks, such as MP3, AMR-WB, therefore they lack adaptability. In this work we present deep neural network model for the lossy audio compression with focus on speech. Our approach to the problem is based on the optimization of autoencoders, which is difficult task due to the non-differentiability of compression loss. We show all steps of speech compression pipeline end-to-end from raw speech data. The results of our experiments are similar and sometimes better than existing audio coding format.

1. Introduction

Data compression is used nearly everywhere on the internet – streaming online videos and music, sharing images in social networks, storing thousands of videos and songs on a cloud. Therefore data compression is a well-studied problem, where engineers designed codes for a given discrete data ensemble with minimal entropy.

Data compression has two categories: lossless and lossy. The former data compression algorithms compress files without losing information, in other words it is reversible. On the other side, lossy data compression may lose some information, but in general lost information is indistinguishable for human ear. Lossless audio compression algorithms rarely reach a compression ratio larger than 3:1, where compression ratio is the ratio of the size of original file and compressed one, while lossy audio compression algorithm obtain compression ratios up to 50:1 and higher. Therefore, lossy audio compression is dominating technology and we will work on this category of compression in our work.

All modern compression standards are the result of hand-designed domain-specific research. The one of most recent ones is Opus [1], which is based on two initially separate standard proposals from Skype Technologies S.A. (now Microsoft) and Xiph.Org Foundation. However, the new hardware technology, new media formats , diverse requirements and content types demand more flexible compression algorithms than existing ones.

Deep learning is a branch of machine learning based on a set of algorithms that attempt to model high level abstractions in data. Neural networks are currently the state of the art when it comes to ‘cognitive’ tasks like speech recognition [2][3], denoising autoencoders [4], image classification [5], and etc. Data compression has been one of the problems which neural networks were suspected to be good at. Promising first results were achieved in image compression using generative models [6], recurrent neural networks [7], and autoencoders [8]. However, there is not much work in audio compression.

Until now, there are only two published papers [17][18] where authors try to solve audio compression problems using neural networks. In this work we will show a framework for the end-to-end optimization of an lossy audio compression using autoencoders. The main difficulty in training is the fact that lossy compression is non-differentiable problem. In other words, due to quantization derivatives of coefficients are zero at all points except at integers, where it is undefined. One possible simple solution of this problem is rounding-based quantization [9].

2. Autoencoders

An autoencoder (Fig. 1) is a neural network that is trained to attempt to copy its input to its output. It has hidden layer that describes a code used to represent the input. The network has two parts: an encoder function and a decoder function that produces a reconstruction. Autoencoders can be thought of as being a special case of feedforward networks and can be trained with all the same techniques, such as mini batch gradient descent following gradients computed by backpropagation [10].

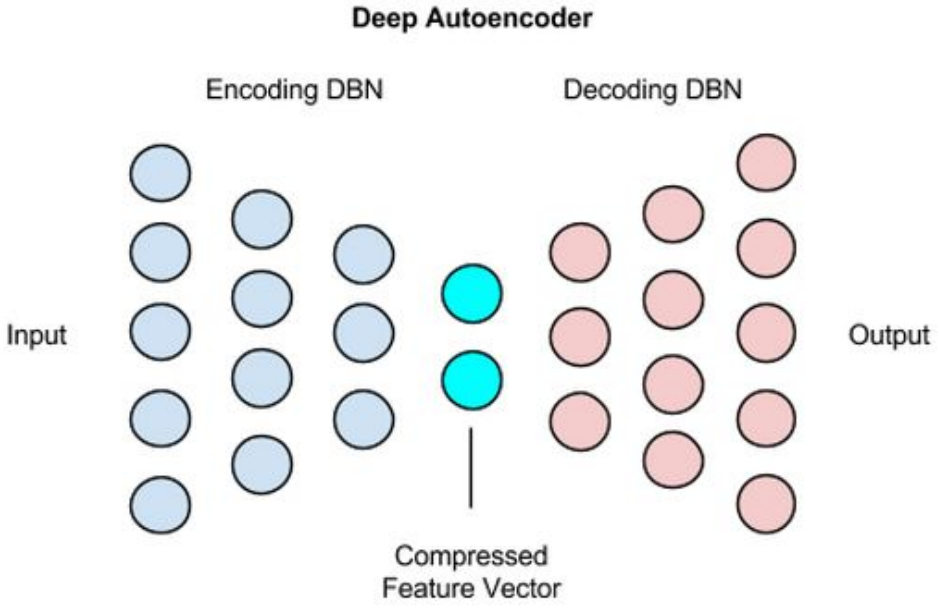


Figure 1

Defined autoencoder has two components: an encoder *f* and a decoder *g*.

$$f : \mathbb{R}^N \to \mathbb{R}^M, \quad g : \mathbb{R}^M \to \mathbb{R}^N$$

Our primary goal is to minimize the distortion, mean square error, of input and reconstructed input.

$$\underbrace{d\left(\mathbf{x}, g([f(\mathbf{x})])\right)}_{\text{Distortion}}$$

The quantized code, which represents audio, is stored losslessly. The main source of information loss is the quantization. Furthermore, some information can be lost by the encoder, and not perfect decoder can result in increasing distortion.

Our distortion cannot be optimized using well-known gradient-based techniques, because $[\bullet]$ is not differentiable. The derivative of $[\bullet]$, the rounding function, is zero everywhere except at integers, even at integers it is undefined. The derivative can be replaced with the derivative of a smooth approximation, *r*, in the backward pass of backpropagation [11]. Hence, the derivative can be defined in this way,

$$\frac{d}{dy}[y] := \frac{d}{dy} r(y).$$

We should notice that we replace only the derivative of the rounding function with a smooth approximation. Rounding r(y) = y works well as other more sophisticated options, such as stochastic form of binarization [12].

3. Network architecture

Our deep neural network architecture (Fig. 2) was inspired by the work of sub-pixel convolutional neural network [13], residual networks [14], and autoencoders. For the encoder and decoder of the autoencoder we use convolutional neural network [15]. Sub-pixel convolutional neural network shows that super-resolution can be done much more efficiently by operating in the low-resolution space, in other words, by convolving vectors and then upsampling instead of upsampling and then convolving a vector.

The one-dimensional vector is first convolved twice by increasing the number of channels from 1 to 128. Afterwards, there are three residual blocks [14], where each block has two supplementary convolutional layers. It follows by additional convolutional layer. Sub-pixel convolutional layer is the sequence of convolution and the reorganization, where last one turns a tensor with many channels into a tensor with fewer channels but same dimensionality. After rounding coefficients to the nearest integer, sub-pixel convolutional layer follows. Following three residual networks, two sub-pixel convolutional layers, final convolutional layer upsample the audio to the resolution of the input. All convolutions are followed by rectification units.

The quantized output of the encoder is the code which is stored in storage losslessly. To achieve it we used arithmetic coding. Arithmetic coding is a form of entropy coding used in lossless data compression and it is superior in most respects to the better-known Huffman method [16]. Compare to other types of entropy coding it differs in that rather than separating the input into component symbols and replacing each with a code, arithmetic coding encodes the entire message into a single number.

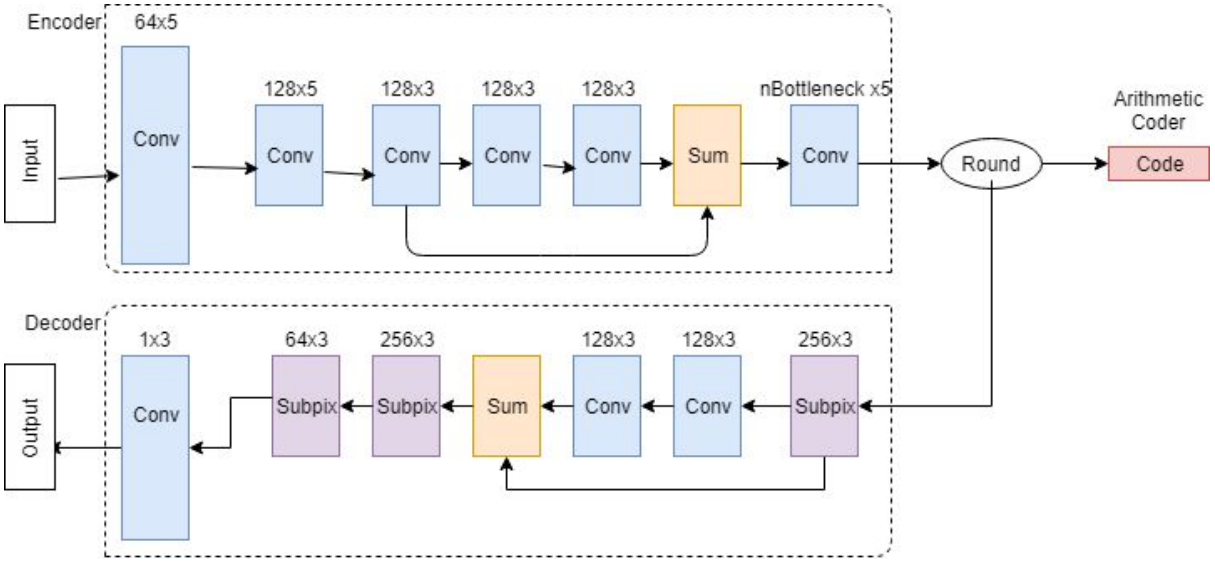


Figure 2. To reduce size, we showed only one residual block of the encoder and the decoder. The notations C x K refers to K convolutions with C channels.

4. Experiments and Results

We trained our network with VCTK dataset [19,] which is available in PyTorch. This dataset includes speech data uttered by 109 native speakers of English with various accents. All speech data was recorded using an identical recording setup: an omni-directional head-mounted microphone (DPA 4035), 96kHz sampling frequency at 24 bits and in a hemi-anechoic chamber of the University of Edinburgh. All recordings were converted into 16 bits, were downsampled to 48 kHz based on STPK, and were manually end-pointed. We manually divided VCTK dataset into testing, and training sets with relative ratio 14 : 80. Notably, this dataset in PyTorch had some unlabeked items, so additional data manipulation is required.

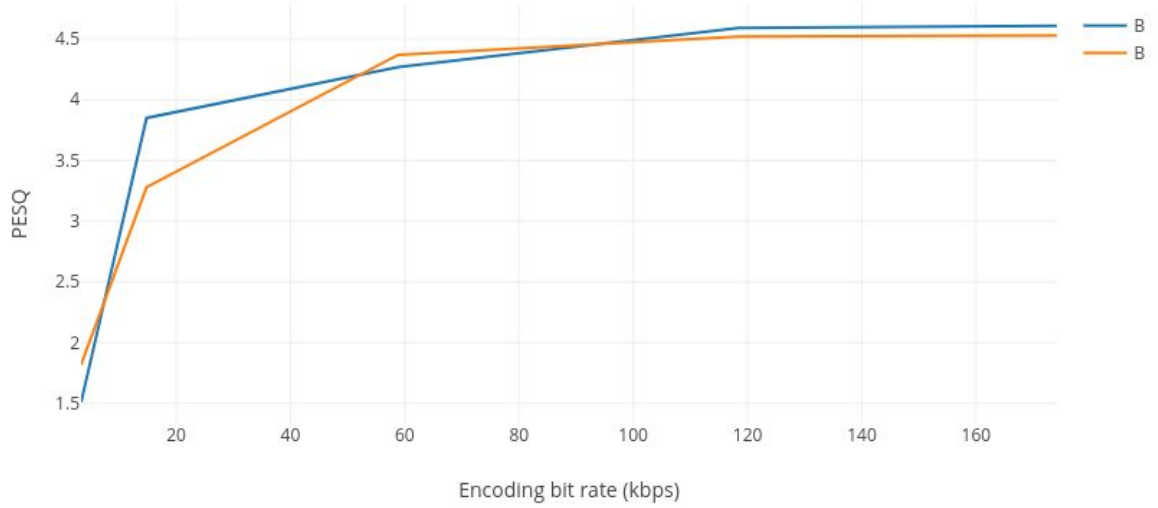
The power of lossy audio compression and the reason why is it so popular in industry is that it reduces size of file significantly, but human ear cannot distinguish between original and compressed audio signals. Lossy compression algorithm discards frequencies that are indistinguishable for human ear, therefore psychoacoustics is essential. Psychoacoustics is the scientific study of sound perception and audiology.. For example, the human can hear sounds in the range 20 Hz to 20000 Hz and with age this interval shrinks. One way to measure sound perception is Perceptual Evaluation of Speech Quality (PESQ). PESQ is an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs.

We compared the results of our method with Opus audio format (Table 1), one of the most recent lossy audio coding format. It designed to efficiently code speech and general audio in a single format and it replaces Vorbis and Speex.
We evaluated the average PESQ of our speech coder versus the Opus standard around 5 different target bitrates.

	DNN	Opus
Bitrate (kbps)	PESQ	PESQ
3.36	1.51	1.82
14.81	3.85	3.28
58.86	4.27	4.37
118.47	4.59	4.52
174.14	4.61	4.53

Table 1

On the graph orange line represents Opus and blue line represents our neural network.



We can clearly see that our model shows slightly better result than Opus encoding audio format.

5. Conclusion

We have shown that our adaptive algorithm achieves similar performance with hand-designed Opus coding audio format. Opus is based on two initially separate standard proposals from Xiph.Org Foundations and Skype Technologies S.A. (now Microsoft). We proved that simple but effective approach of dealing with non-differentiability in training autoencoders for lossy compression, simple objective function can compete with modern standard codecs. Such result was achieved using efficient convolutional architecture and simple rounding-based quantization.

In future work we would like to advance our objective function by including entropy and optimizing rate-distortion trade-off. Additionally, we can include concepts such as adversarial loss [6], perceptual loss, and etc. Improving network architecture design can also lead improvement of performance.

Until now there is not many work on audio compression using neural networks. We can explore lossy and lossless compression, speech and music compression, and etc. This work also proved that deep neural networks have power to solve very different engineering problems.

6. References

[1] Opus (homepage) http://opus-codec.org/

[2] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine, 29(6), 82-97.

[3] Graves, A., Mohamed, A. R., & Hinton, G. (2013, May). Speech recognition with deep recurrent neural networks. In Acoustics, speech and signal processing (icassp), 2013 ieee international conference on (pp. 6645-6649). IEEE.

[4] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P. A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 11(Dec), 3371-3408

[5] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105)

[6] Santurkar, Shibani, David Budden, and Nir Shavit. "Generative compression." arXiv preprint arXiv:1703.01467 (2017).

[7] Toderici, George, et al. "Full resolution image compression with recurrent neural networks." arXiv preprint (2016).

[8] Del Testa, Davide, and Michele Rossi. "Lightweight lossy compression of biometric patterns via denoising autoencoders." IEEE Signal Processing Letters 22.12 (2015): 2304-2308.

[9] Theis, Lucas, et al. "Lossy image compression with compressive autoencoders." arXiv preprint arXiv:1703.00395 (2017).

[10] Goodfellow, Ian, et al. Deep learning. Vol. 1. Cambridge: MIT press, 2016

[11] Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." nature 323.6088 (1986): 533.

[12] Williams, Ronald J. "Simple statistical gradient-following algorithms for connectionist reinforcement learning." Reinforcement Learning. Springer, Boston, MA, 1992. 5-32.

[13] Shi, Wenzhe, et al. "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[14] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[15] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

[16] Moffat, Alistair, Radford M. Neal, and Ian H. Witten. "Arithmetic coding revisited." ACM Transactions on Information Systems (TOIS) 16.3 (1998): 256-294.

[17] Kankanahalli “End-to-end optimized speech coding with deep neural networks”, ICAASP 2018.

[18] Morishima, Shigeo, H. Harashima, and Y. Katayama. "Speech coding based on a multi-layer neural network." Communications, 1990. ICC'90, Including Supercomm Technical Sessions. SUPERCOMM/ICC'90. Conference Record., IEEE International Conference on. IEEE, 1990.

[19] VCTK dataset homepage: http://datashare.is.ed.ac.uk/handle/10283/2651