Using a generative adversarial network in instrument audio synthesis

Justin Zhang 5/9/18

*Background*

As technology progresses, new and novel ways are developed to solve problems. Many problems cannot easily be solved via traditional programming; one such example is the classification of images. An interesting solution to these types of abstract problems is techniques to make machines learn the solution to the problems themselves; this is consequently called machine learning. Neural nets are a common form of machine learning, modeled after neurons. This research focuses on using deep convolutional generative adversarial networks, a type of neural network, to generate original audio of the quality of a certain instrument, which is, in this case, a clarinet.

Audio can be processed in many ways, of which one of the most common is a conversion into spectrograms. Spectrograms have 4 dimensions: time, frequency, intensity, and phase. They are most commonly represented as images, and can be treated as such. They are calculated using consecutive Fourier transforms over small samples of overlapping time signals. It has been chosen in this research to preprocess audio into a spectrogram before being processed by the neural net, as many previous neural nets have successfully processed spectrograms; for example, in Pons, Slizovskaia, Gong, Gómez, & Serra, 2017, audio was successfully classified using spectrograms as input into their neural net. Since an instrument is characterized by its patterns of higher frequencies above the main frequency (timbre), a spectrogram is useful in capturing those said frequencies and making it easier for the neural net to find patterns in the audio.

The network is to be trained with unlabeled spectrograms of samples of single notes played on a clarinet. Therefore, this network is unsupervised. Currently, there are two types of unsupervised generative networks: variational autoencoders, first proposed in Kingma & Welling, 2013, and generative adversarial networks, first proposed in Goodfellow et al., 2014.

Variational autoencoders (VAEs) work by training an autoencoder. An autoencoder consists of two parts: an encoder and a decoder. The encoder takes an image and compresses into a small vector with few values, and the decoder reconstructs the image using the small vector outputted by the encoder. The error is calculated using the dissimilarity of the reconstructed image as compared to the original. The autoencoder is trained as one large network with the encoder and decoder combines. A set of images is needed to be used as training data for the network to learn how to encode and decode. This autoencoder can then generate images by creating a random small vector and using that vector as an input to the decoder. The random vector introduces variation that provides uniqueness to the generated image; hence, it is called a variational autoencoder (Kingma & Welling, 2013).

Generative adversarial networks (GANs) work using two parts: a generator and a discriminator. The generator takes an input of a random vector of a selected size and outputs an image. The discriminator takes the generated images and training images and attempts to discriminate between them. The error is simply the measure of how wrong the discriminator was in distinguishing between the generated images and the training (real) images. The discriminator is trained to minimize the classification error, while the generator is trained to maximize the classification error. The generated images are simply the output of the generator.

The original paper proposing GANs only specified a novel way to train networks to generate images (Goodfellow et al., 2014). The two primary concrete implementations of GANs are the deep convolutional GAN (DCGAN) (Radford, Metz, & Chintala, 2015) and the Wasserstein GAN (Arjovsky, Chintala, & Bottou, 2017). The Wasserstein GAN (WGAN) uses the Wasserstein distance as a measure of error, and heavily involves complicated maths which are outside the scope of this research (Arjovsky, Chintala, & Bottou, 2017). Radford, Metz, & Chintala, 2015 outlined architectural guidelines for DCGANs: pooling layers that would traditionally be used for convolutional neural networks (CNN) are replaced by strided (discriminator) or fractionally-strided (generator) convolutions; batchnorm is used for both the discriminator and generator; no fully connected layers are used; the generator uses a ReLU activation for all layers except the output, where tanh is used; and the discriminator use leaky ReLU for all layers. A DCGAN also trains similarly to other convolutional neural networks (Radford, Metz, & Chintala, 2015). A DCGAN is used in this research.

Currently, much research has been done to classify audio (examples: Kingma & Welling, 2013; Pons, Slizovskaia, Gong, Gómez, & Serra, 2017), but relatively little has been done to synthesize audio, with the only previous group to be found to do so being Engel et al., 2017. Engel et al., 2017, also used a VAE instead of the DCGAN proposed to be used in this research. VAEs and GANs have advantages and disadvantages, with one being more suited than the other for certain applications. This research explores the feasibility and possibly advantages of using a GAN (more specifically, a DCGAN) to generate audio instead of a VAE. The DCGAN will generate images that can then be converted back to audio. The feasibility of using a GAN to generate audio can be used to extrapolate the feasibility of using a GAN in other applications, such as chord generation. These areas (chord generation etc.) are also possible future topics for research if this research is successful.

It should be noted that previous research on the classification of audio can be used to help design the discriminator network in the DCGAN. Meta-parameters were found (ex. frame size of the convolutional layers) to maximize classification success in Pons, Slizovskaia, Gong, Gómez, & Serra, 2017. A stronger discriminator network means that the generator network would have to learn more aggressively, resulting in better images and audio. However, these previous findings may not scale to DCGANs, because of the different natures of the networks. This research can further examine the optimal architectures and metaparameters required for generation success. The results can also add to the base of knowledge required to gain insights to the mysterious and commonly researched “black box” of neural nets, i.e. what the network is doing on the inside to extract patterns, etc. (Shwartz-Ziv & Tishby, 2017).

*Works cited*

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. *ArXiv E-Prints*.

Engel, J., Resnick, C., Roberts, A., Dieleman, S., Eck, D., Simonyan, K., & Norouzi, M. (2017). Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders. *CoRR*, *abs/1704.0*. Retrieved from http://arxiv.org/abs/1704.01279

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., … Bengio, Y. (2014). Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 27*(pp. 2672–2680). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

Kingma, D. ~., & Welling, M. (2013). Auto-Encoding Variational Bayes. *ArXiv E-Prints*.

Pons, J., Slizovskaia, O., Gong, R., Gómez, E., & Serra, X. (2017). Timbre Analysis of Music Audio Signals with Convolutional Neural Networks. *CoRR*,*abs/1703.0*. Retrieved from http://arxiv.org/abs/1703.06697

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *ArXiv E-Prints*.

Shwartz-Ziv, R., & Tishby, N. (2017). Opening the Black Box of Deep Neural Networks via Information. *CoRR*, *abs/1703.00810*. Retrieved from http://arxiv.org/abs/1703.00810