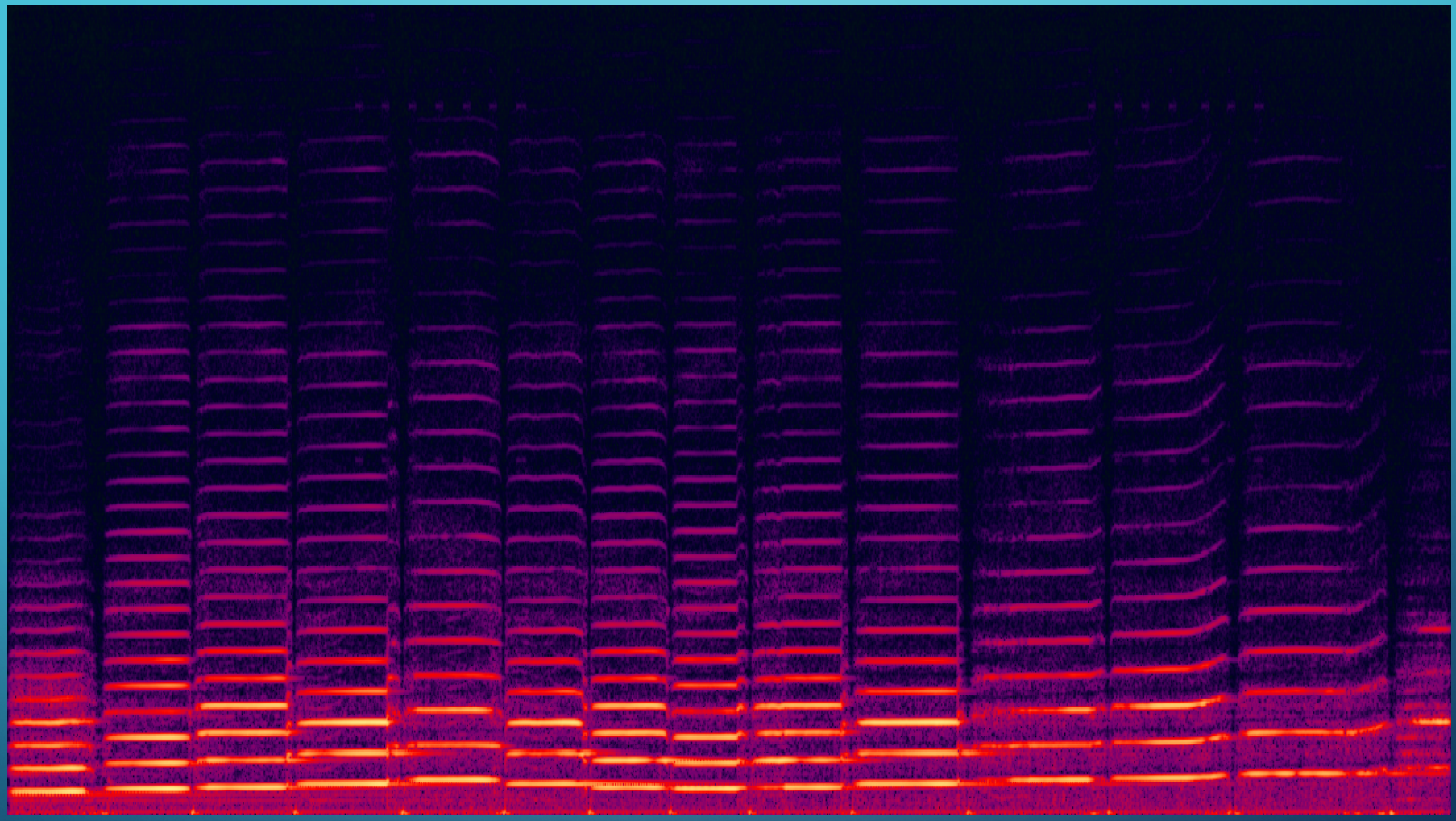
THE USE OF A GENERATIVE ADVERSARIAL NETWORK TO GENERATE INSTRUMENT TONES

JUSTIN ZHANG

GENERATING SOUNDS

- Audio is converted into images as spectrograms
- Uses generative adversarial networks (GANs), to generate images given a set of correct or ideal images
  - Unsupervised, meaning no classification labels are given with the training data
- Generated images are then converted back into audio which will (hopefully) be realistic



A spectrogram of a violin playing

Spectrograms are 3D, with time, frequency, and frequency intensity as the 3 dimensions

FUTURE WORK

- If somewhat or completely successful, this could be applied to other instruments, chords with either one instrument or different instruments, melodies, etc.
- If unsuccessful, different network architectures or different meta-pararmeters of the network could be tested

DIFFERENCE FROM PREVIOUS RESEARCH

- Previous research either generated MIDI melodies or used a different network architectures to generation realistic tones
- Using spectrograms in this project captures overtones and gives realistic (instead of purely synthesized) audio
- Previous research did not design networks with the intent of going from timbre analysis and/or timbre creation into chord generation etc.

VARIABLES AND HYPOTHESIS

- Independent variable: algorithm/network structure used
- Dependent variable: plausibility of audio generated
- Hypothesis: The algorithm will work similarly well to previous examples and can be expanded to chords, melodies, and/or combinations of instruments

CURRENT PROGRESS

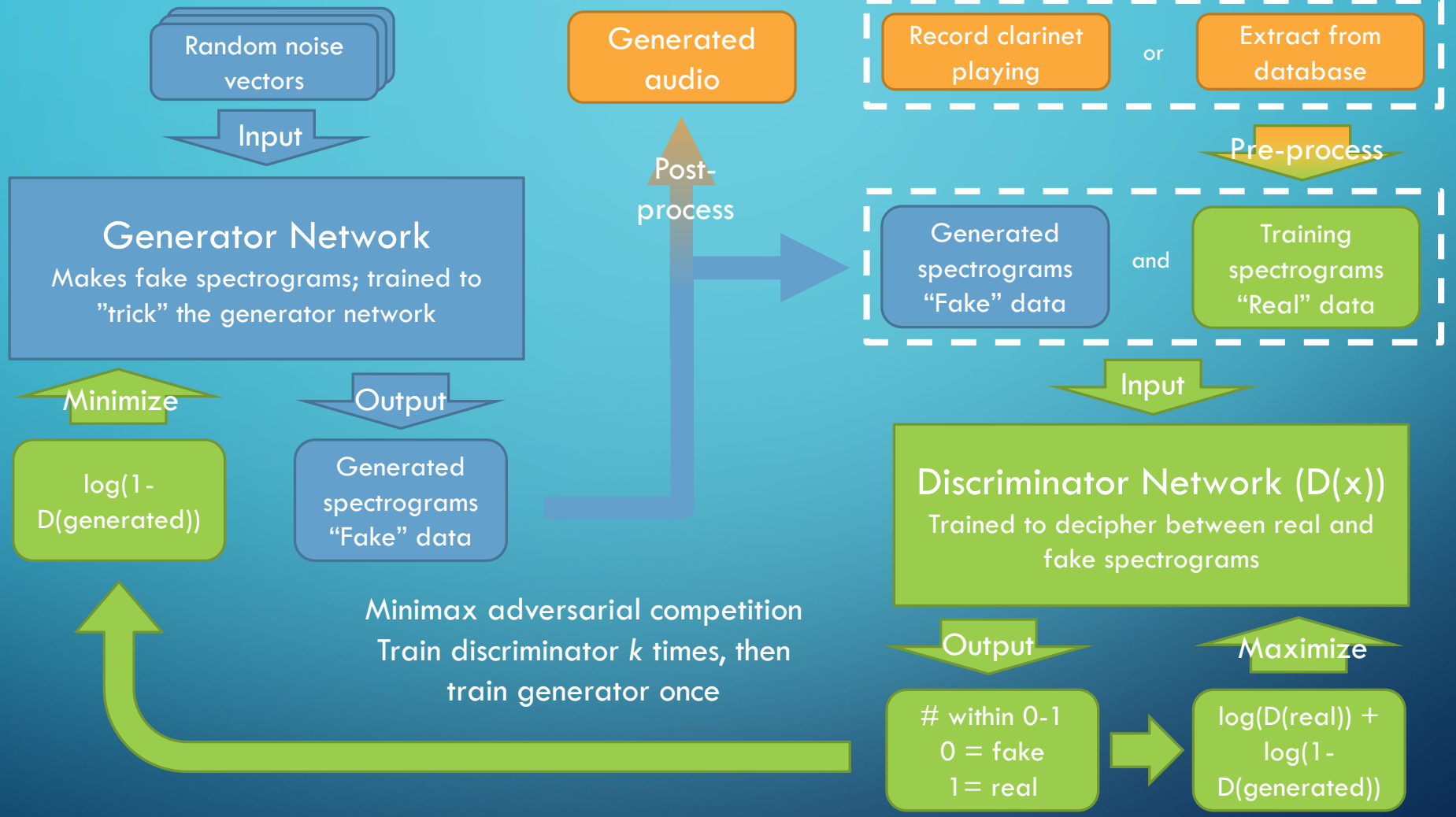
Math:

- Finished linear algebra course
- Currently working on multivariable course
- Learn how to code a neural network in Python (Jupyter notebooks)

Science:

- Found GANs and researched their architecture
- Designed a workflow diagram for easy interpretation
- Found previous machine learning tone generation attempts

To do: look at implementations of GANs, ex. deep convolutional GANs (DCGANs)
Continue the mathematics courses and Jupyter notebook Python exercises


Random noise vectors
Input
Generator Network
Makes fake spectrograms; trained to "trick" the generator network
Minimize
log(1-D(generated))
Output
Generated spectrograms "Fake" data
Post-process
Generated audio
Record clarinet playing
or
Extract from database
Pre-process
Generated spectrograms "Fake" data
and
Training spectrograms "Real" data
Input
Discriminator Network (D(x))
Trained to decipher between real and fake spectrograms
Output
# within 0-1
0 = fake
1= real
Maximize
log(D(real)) + log(1-D(generated))
Minimax adversarial competition
Train discriminator k times, then train generator once

REFERENCES

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

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