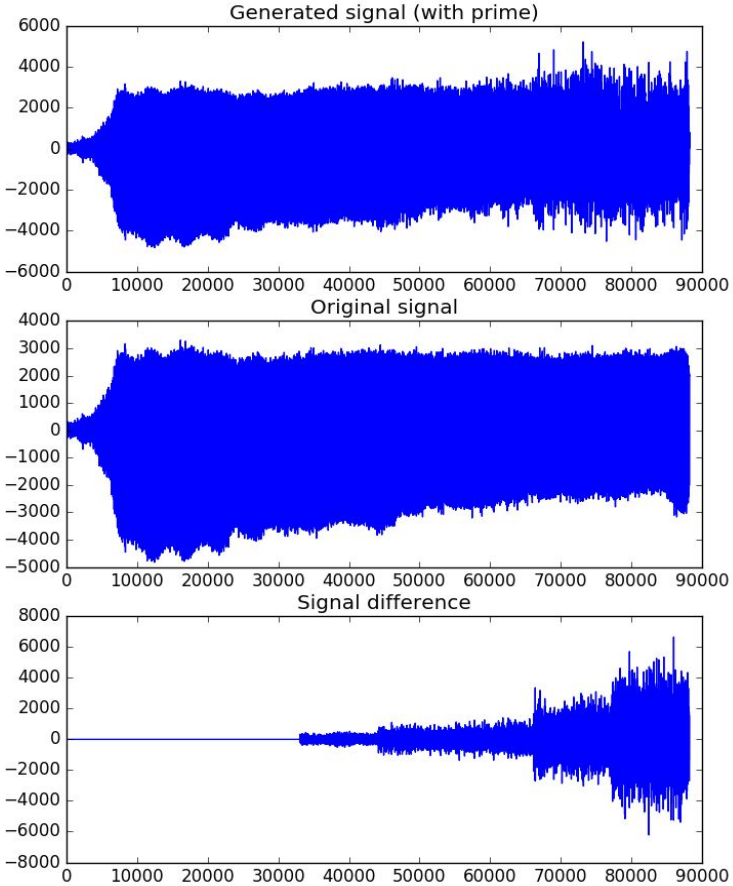
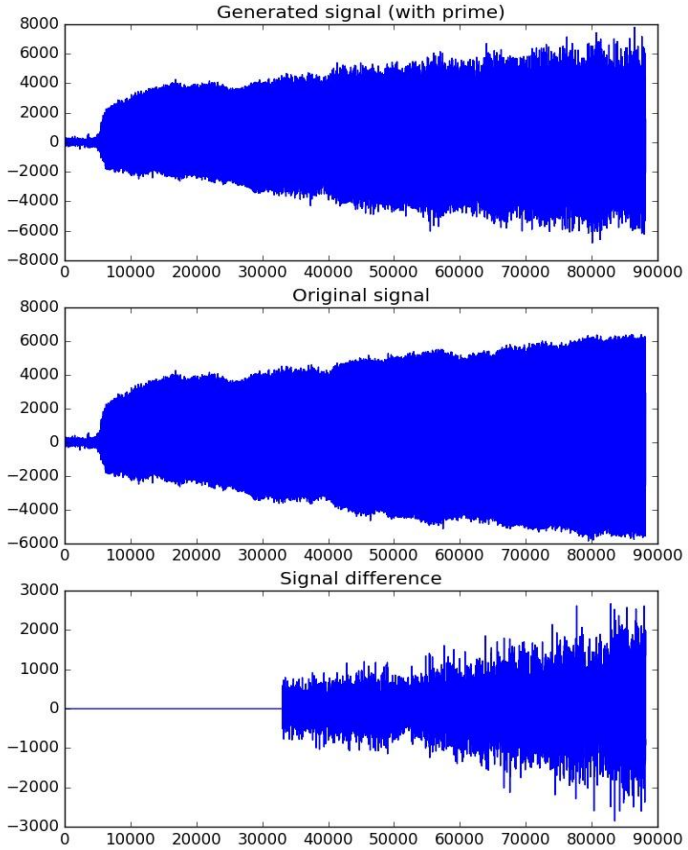
What have we done since Tuesday

- Finished building our data pipeline
- Tried to understand the DFT and FFT
- Tried
- Build a first version an RNN/LSTM
- Train model on 39 Woodwinds flute sounds (raw audio signals without downsampling)
- Generate flute sounds based on prime sequence
- Added DFT representation to pipeline


Generated signal (with prime)
Original signal
Signal difference


Generated signal (with prime)
Original signal
Signal difference

Generating process

	Prime sequence	RNN output	Generated sequence
1	[x1, x2, x3]	[x2', x3', x4']	[x1, x2, x3, x4']
2	[x1, x2, x3, x4']	[x2', x3', x4', x5']	[x1, x2, x3, x4', x5']
3	[x1, x2, x3, x4', x5']	…	

Next steps

- How to determine what is a “good result” (goal was, learn prediction of sound decay)
- Use DFT representation
- Use smaller time slices in representation
- Vary size of hidden layer
- Use GRU units (instead of LSTM units)
- Use an instrument that produces a *significantly decaying* sound (e.g. string instrument)
- Compare generating sequences by means of *mean squared error*