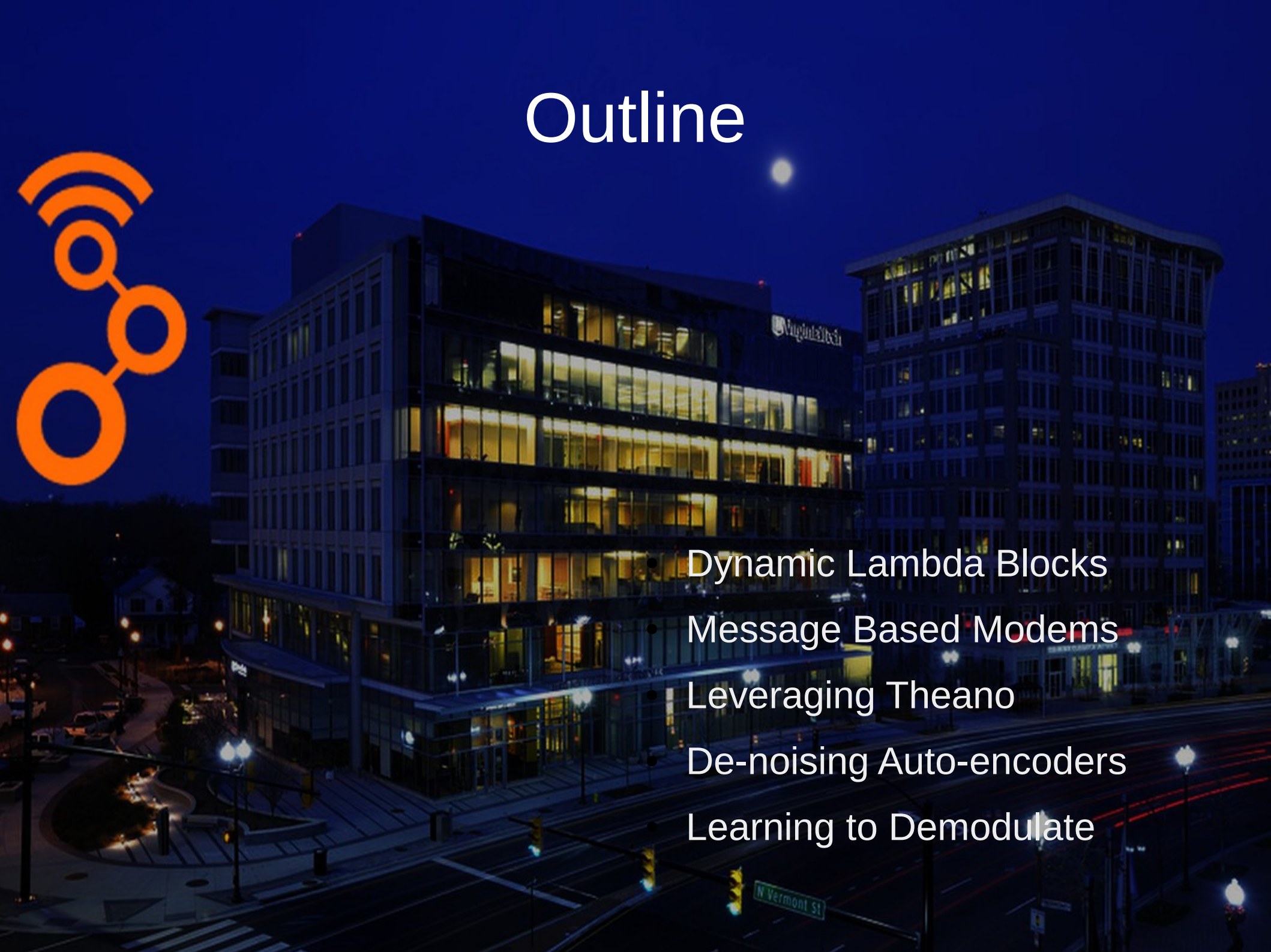

GNURadio
THE FREE & OPEN SOFTWARE RADIO ECOSYSTEM
Bursty Signals & Radio Learning
Tim O'Shea, Virginia Tech
www.oshearesearch.com
N Vermont St

# Deep Dreams from VT Arlington

# Deep Dreams from VT Arlington

View from Ballston World of Beer after #Cyberspectrum Meetup last night?

# Outline

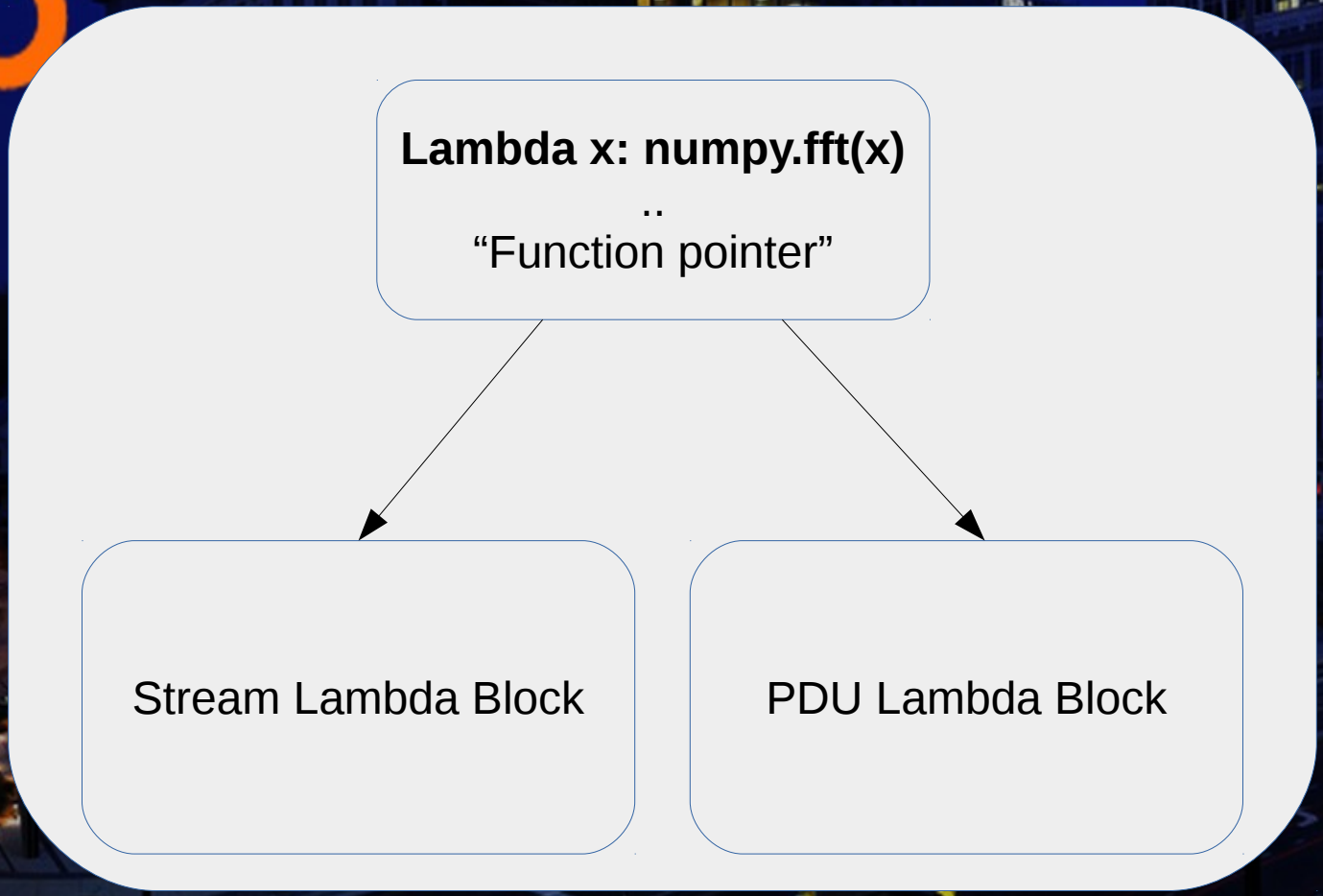



- Dynamic Lambda Blocks
- Message Based Modems
- Leveraging Theano
- De-noising Auto-encoders
- Learning to Demodulate

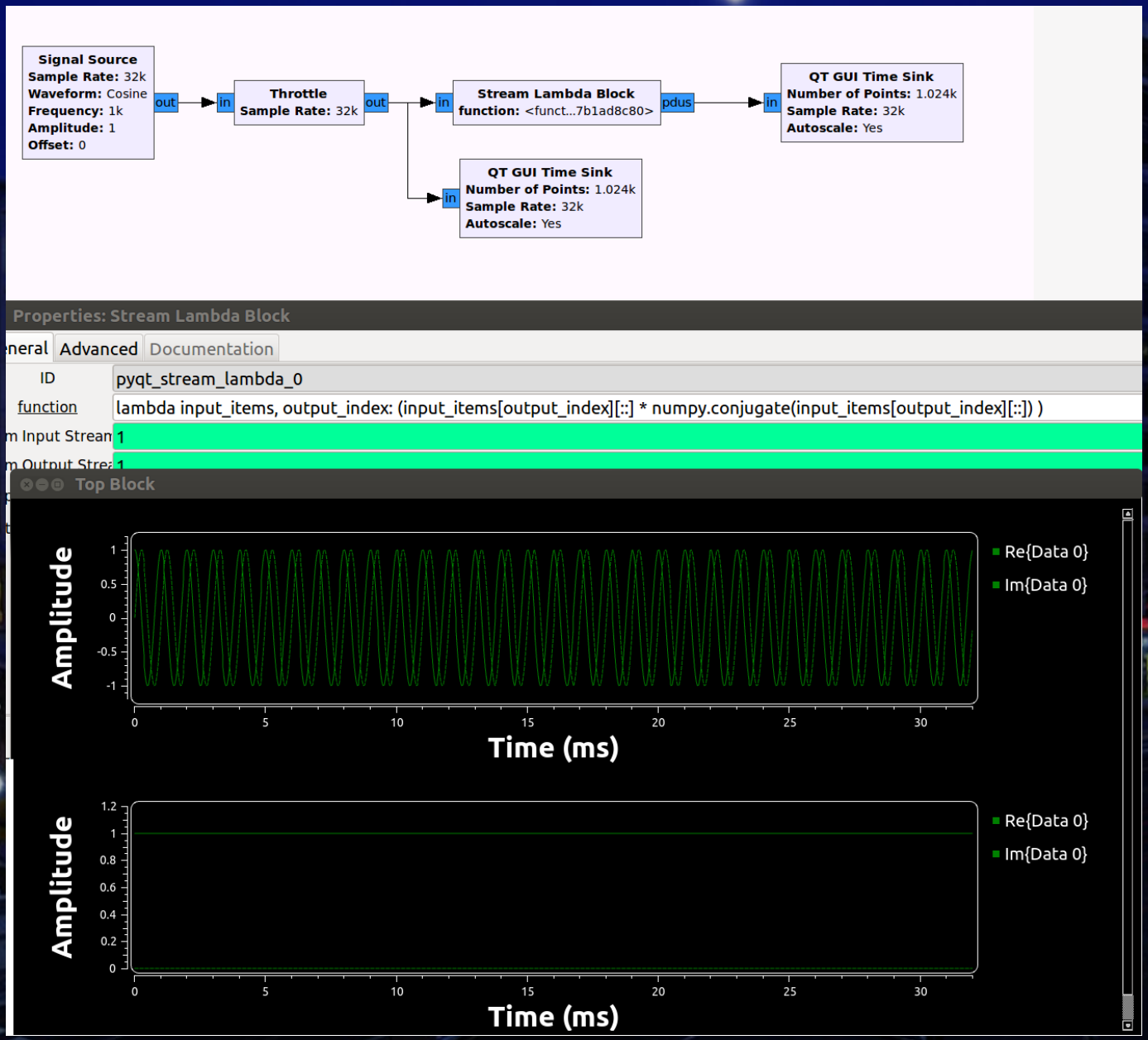
# Lambda Blocks

- In-Line anonymous python/numpy algorithms from GRC

Fastest Blocks Ever ….



# Stream Lambda Block



# Dynamic Stream Lambda Block

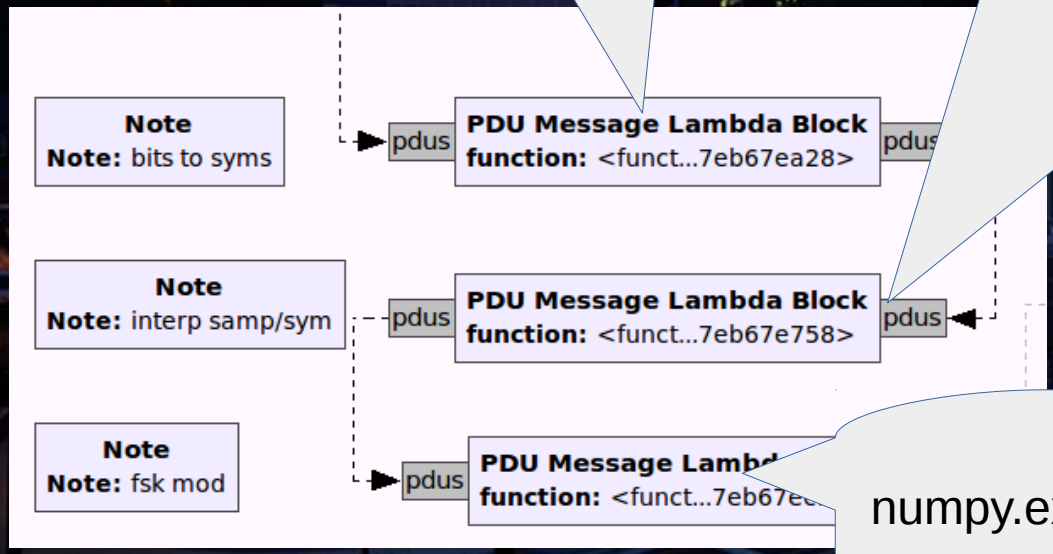

- **Dynamic Lambda Demo**

# Message Based Modems

- Throwing together an FSK modulator with Lambda Blocks

lambda x: numpy.array(x, dtype=numpy.float32)*2-1
**{0,1} → {-1,+1}**

lambda x: numpy.tile(x,[sps,1]).T.reshape([1,len(x)*sps])
Iff sps==4 then **{-1} → {-1,-1,-1,-1}**




lambda x: numpy.array(
numpy.exp(1j*2*numpy.pi*((dev*x*numpy.arange(len(x)))
/samp_rate)),dtype="complex64")
(mix with a **carrier** at FSK deviation frequency)

# Message Based Modems


- **gr-fsk-burst repo**
- **gr-psk-burst repo**

- Kiran has a great write up on the PSK modem!

http://tinyurl.com/ngu2bgf

# Message Based Modems


- **FSK Tx Demo**

# gr-theano

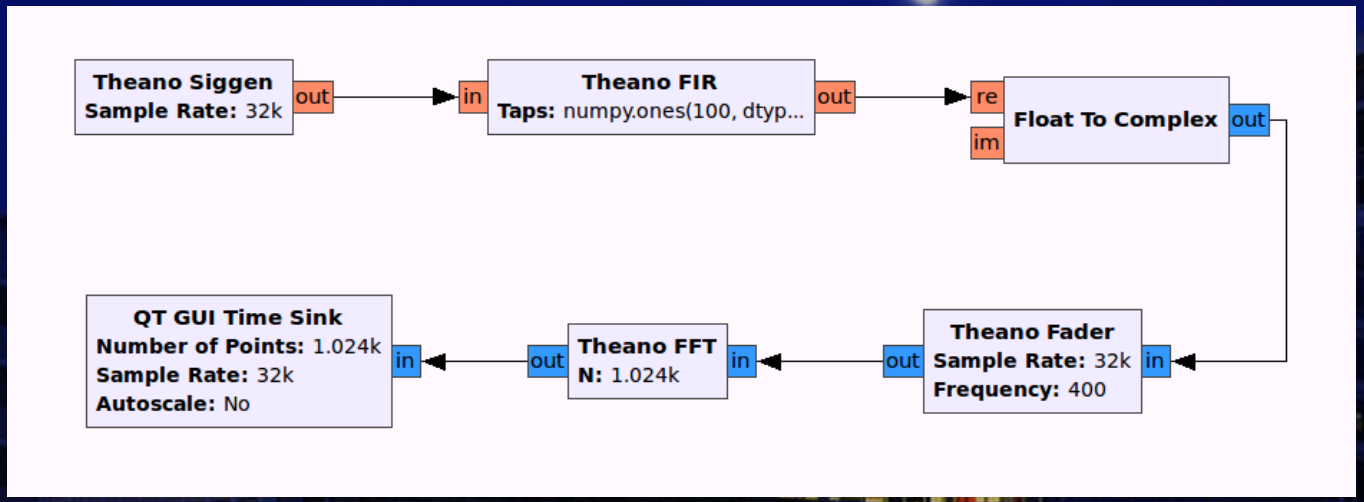
**Theano is a Python library that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently. It can use GPUs and perform efficient symbolic differentiation.http://www.deeplearning.net/software/theano**

**i.e. Write Numpy → Run Algorithms on GPUs**

Built by University of Montreal for Scaling Machine Learning

Instead of putting a python lambda expression in a block, it compiles the expression and returns a “Wrapped” function call to compiled version

# gr-theano




Demoed this at FOSDEM '15

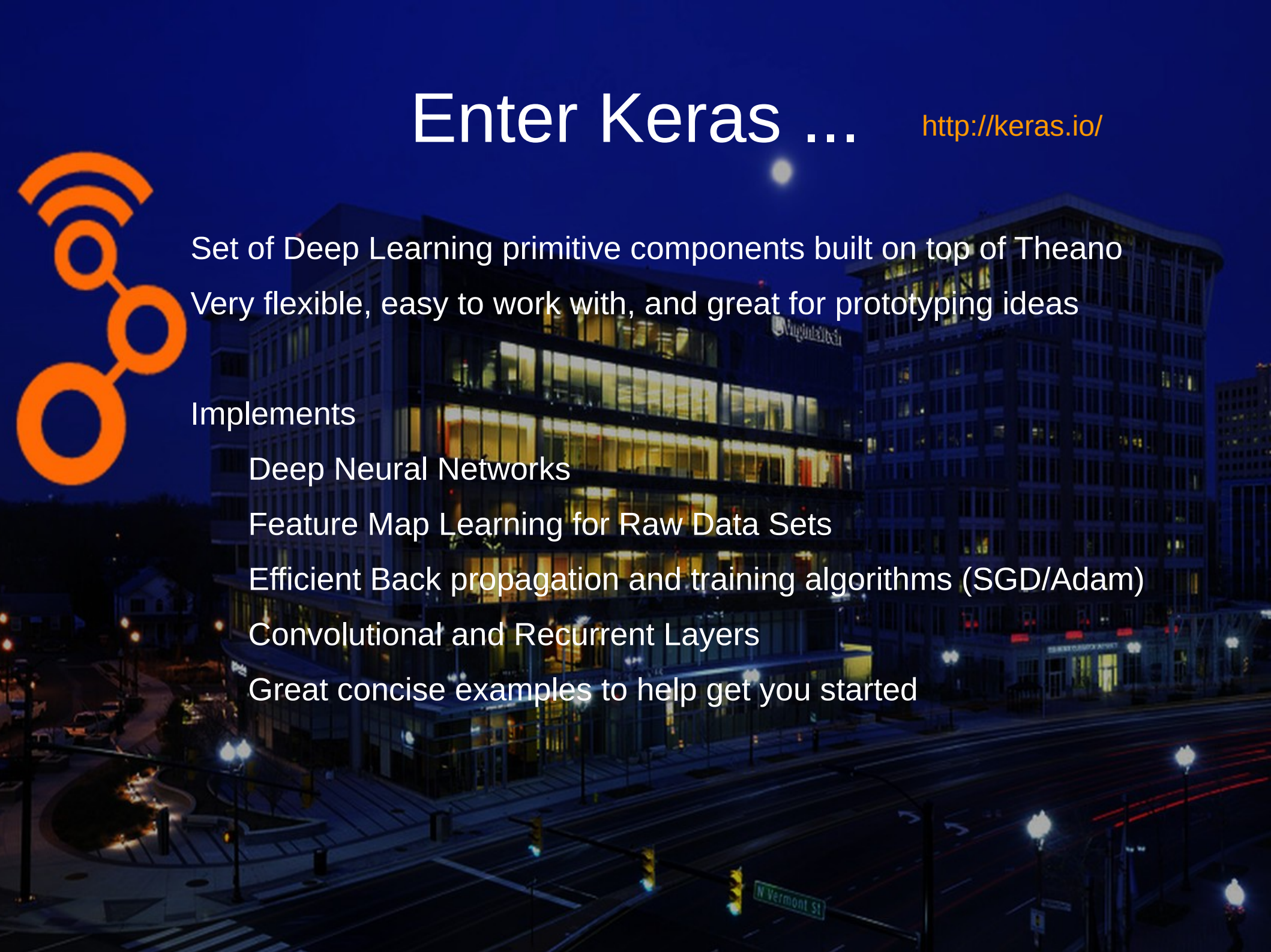
This works and is pretty cool, except:

- We go back and forth from host to/from GPU each time
- Theano doesnt really support numpy.complex64 types for now :-(

Best uses would be for a very expensive algorithm offload in one block

i.e. Monolithic CAF search block or something

# Enter Keras ... http://keras.io/




Set of Deep Learning primitive components built on top of Theano

Very flexible, easy to work with, and great for prototyping ideas

Implements

- Deep Neural Networks
- Feature Map Learning for Raw Data Sets
- Efficient Back propagation and training algorithms (SGD/Adam)
- Convolutional and Recurrent Layers
- Great concise examples to help get you started

# De-noising Auto-encoders

**Reconstruction Cost (Loss Term for backprop)**

- Output of the network is equal to the input of the network

**Introduce Noisy Input**

- Dropout or corruption on input layer randomly during training
- Still target a reconstruction of the clean signal

**Narrowing Hidden Layer**

- To force a dimensionality reduction in the hidden layer (y)

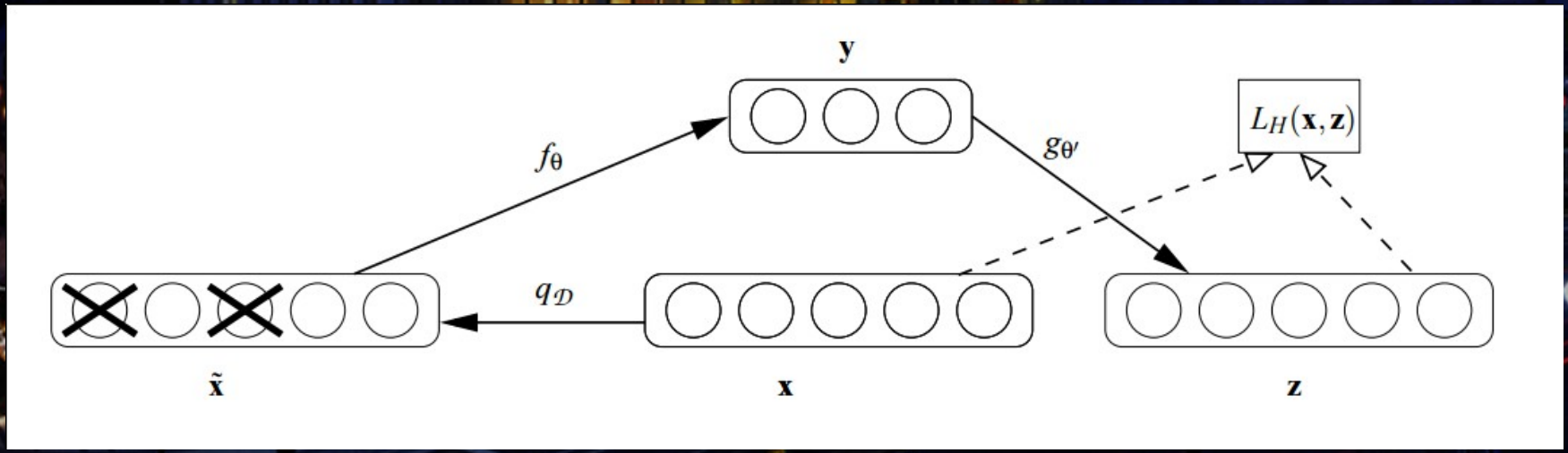
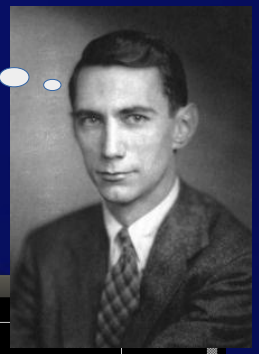


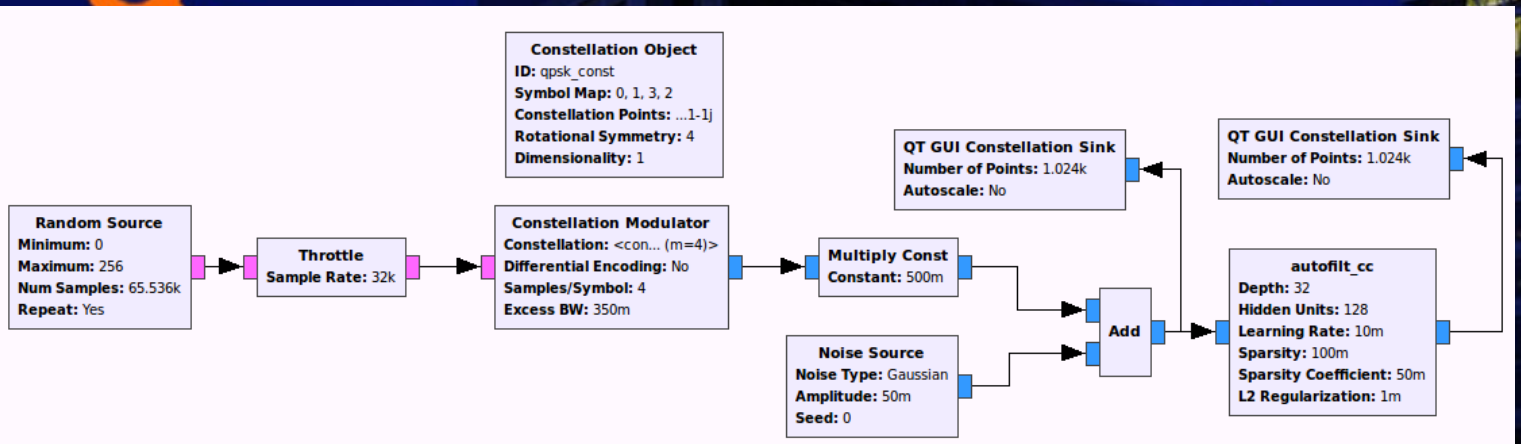

Image From … Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." The Journal of Machine Learning Research 11 (2010): 3371-3408.

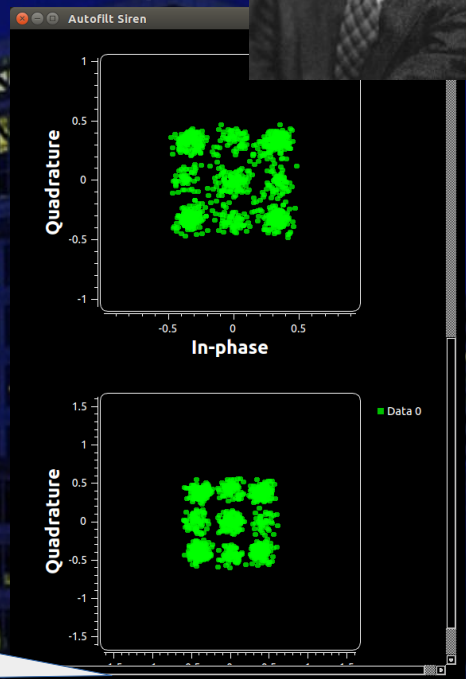
# De-noising Auto-encoders

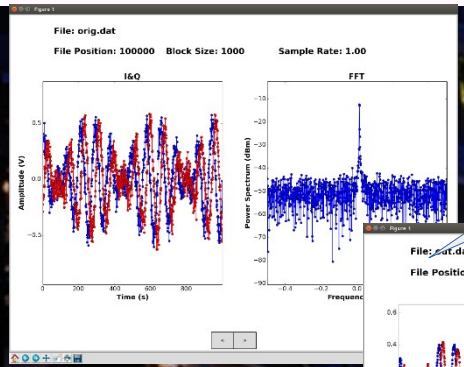



Ongoing collaboration with Jon Corgan (http://corganlabs.com/)

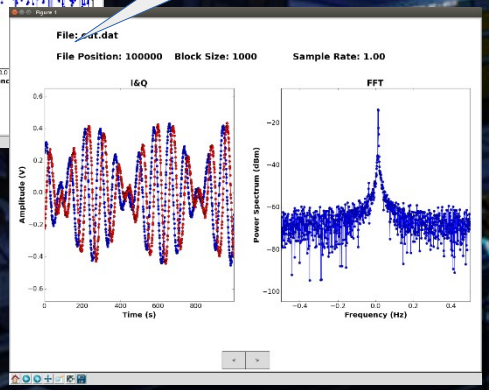
Naively de-noising structured signals in GNU Radio!



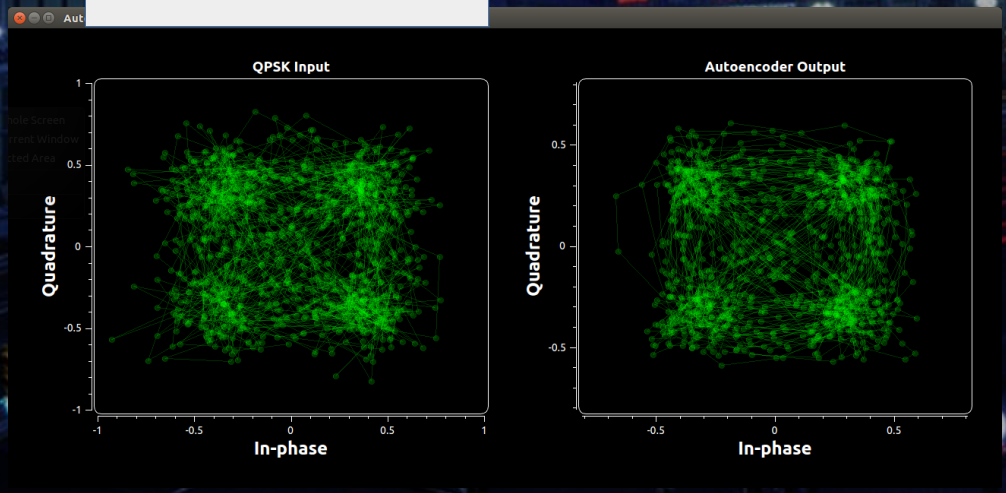




Dial Tone Denoising

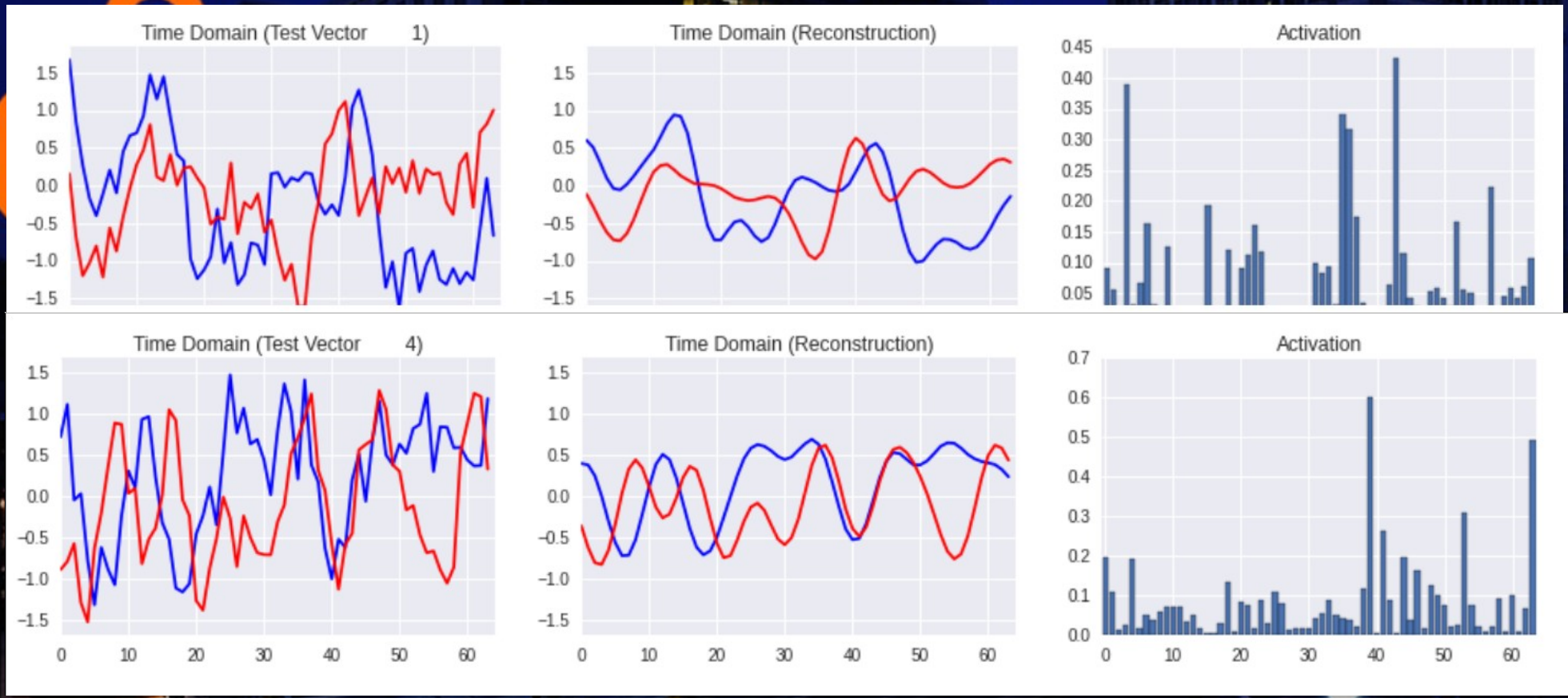



PSK De-noising



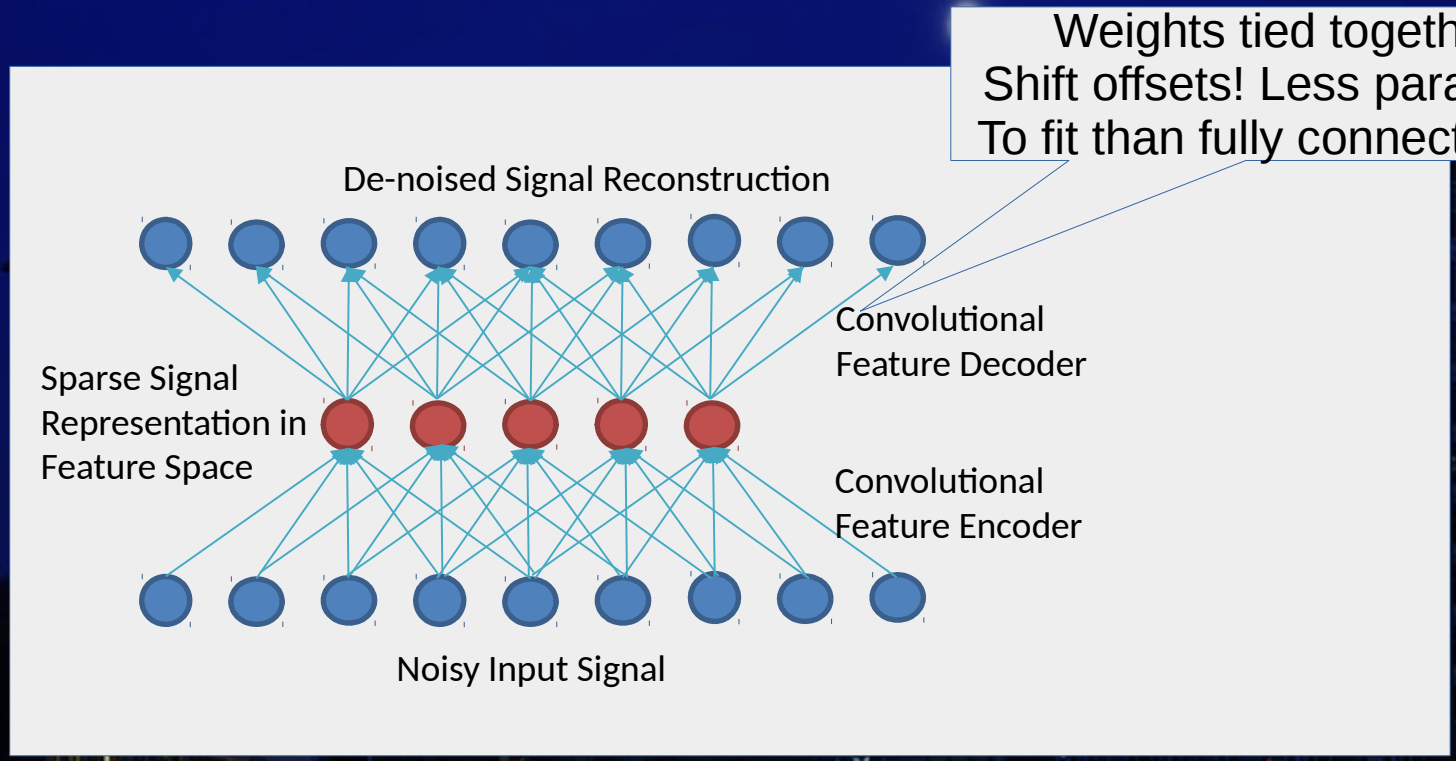
# De-noising Auto-encoders

- Naive PSK de-noising in time (iPython, Corgan)
- - Using strong regularizers




- Input
- Output
- Representation

# Convolutional Auto-Encoders



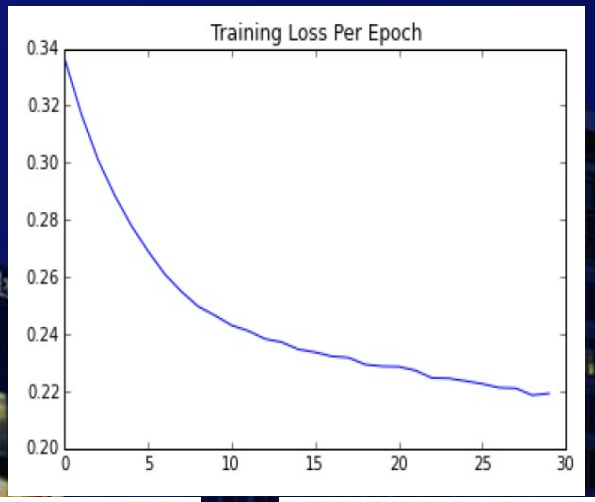

Improved version of the auto-encoder when we care about

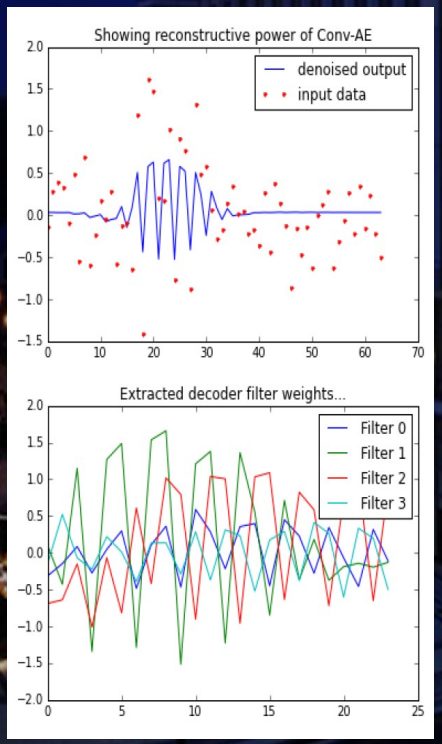
- Forcing the network to learn shift invariance!
- Reducing the number of parameters to fit

Mostly used on Images Right now! Also some really cool time series work on audio @ Google! → *Towards End-To-End Speech Recognition Using Deep Neural Networks, Invited Talk at ICML Deep Learning Workshop, July 2015.*

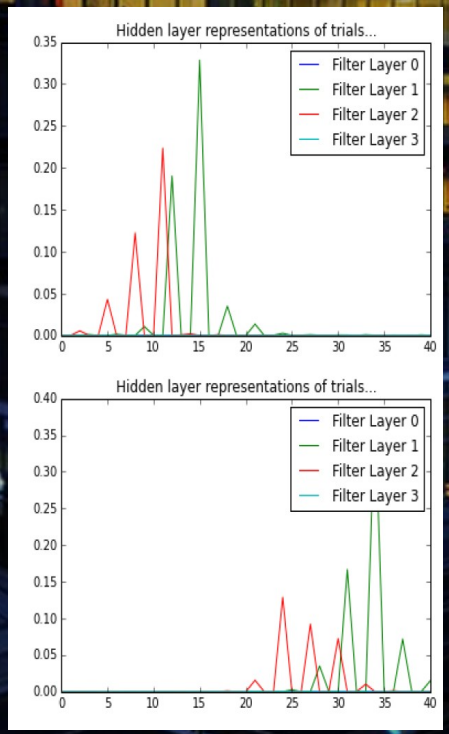
# Learning Filters with Convolutional Auto-Encoders

- Unknowns
  - 12 bit preamble values
  - Noise & Times of arrival
- Demonstrating Generative Nature
  - Convolutional AE /w Dropout Regularizer

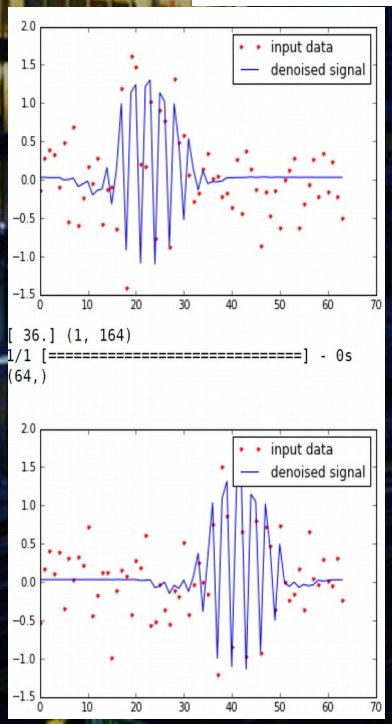


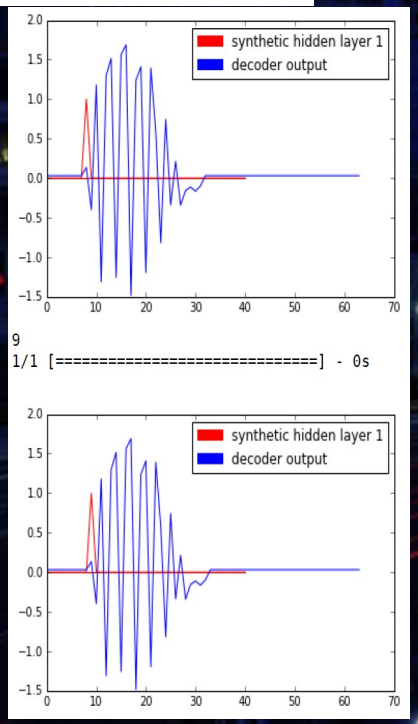

Learning




Compressing

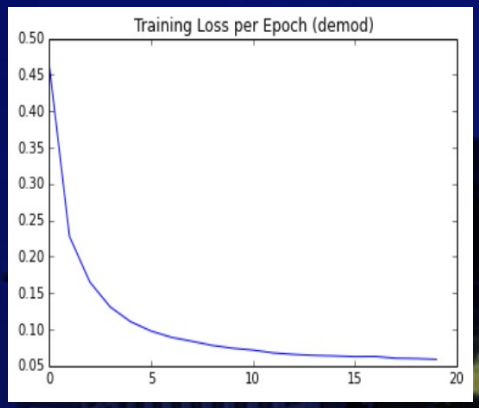



De-noising




Modulating

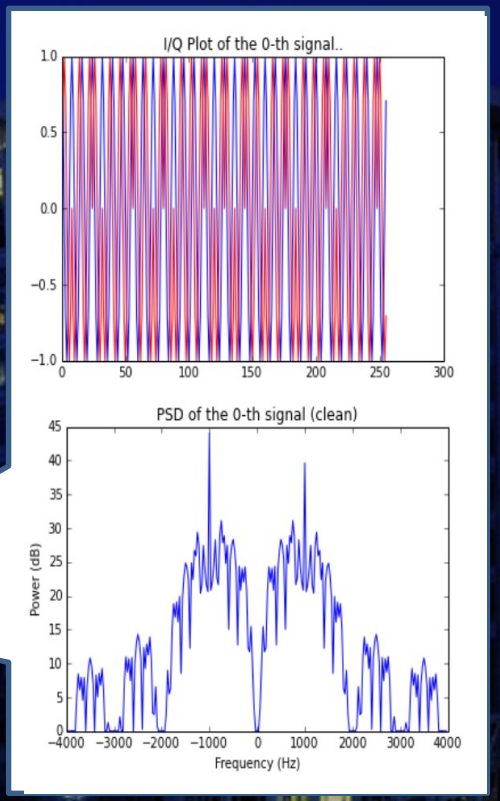
# Training an FSK Demodulator




Bits
Dense Layer
Dense Layer
Tan-H Non-linearity
Dense Layer
Dense Layer
Dropout Layer (0.5)
Clean Waveform

Packet Bits

Modulated Signal




**Training an "FSK Slicer"**

Supervised learning task:

Map noisy Modulated waveform to Bits

- Produces 0 BER at reasonably high SNR …
- More characterization needed to compare performance to classical methods …
- Does not yet address timing recovery and equalization …

```
[[ 0.  0.  0. ...,  0.  0.  0.]
 [ 0.  0.  0. ...,  0.  0.  0.]
 [ 0.  0.  0. ...,  0.  0.  0.]
 ...,
 [ 0.  0.  0. ...,  0.  0.  0.]
 [ 0.  0.  0. ...,  0.  0.  0.]
 [ 0.  0.  0. ...,  0.  0.  0.]]
NN-Demod BER: 0.000000
```

# Future Work

**LOTS** more work to come in this area!

- A
- - Generative models for representing and de-noising and sequence modeling all kinds of structures signals and signal processing algorithms
- - Real emergent learned radio behavior!
- - Learned features, less expert knowledge

**DEEP COGNITIVE RADIO! :-0**

Questions?