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 ….

Lambda x: numpy.fft(x)

“Function pointer”

Stream Lambda Block

PDU Lambda Block
Stream Lambda Block

Signal Source
Sample Rate: 32k
Waveform: Cosine
Frequency: 1k
Amplitude: 1
Offset: 0

Throttle
Sample Rate: 32k

Stream Lambda Block
Function: lambda input_items, output_index: (input_items[output_index][:] * numpy.conjugate(input_items[output_index][:]))

QT GUI Time Sink
Number of Points: 1.024k
Sample Rate: 32k
AutoScale: Yes

---

Properties: Stream Lambda Block

General | Advanced | Documentation

ID: pyqt_stream_lambda_0
Function: lambda input_items, output_index: (input_items[output_index][:] * numpy.conjugate(input_items[output_index][:]))

Input Stream 1
Output Stream 1

---

Amplitude
Time (ms)

Amplitude
Time (ms)
Dynamic Stream Lambda Block

- Dynamic Lambda Demo
Message Based Modems

- Throwing together an FSK modulator with Lambda Blocks

\[
\text{lambda } x: \text{numpy.array}(x, \text{dtype=numpy.float32}) * 2 - 1 \\
\{0,1\} \rightarrow \{-1,+1\}
\]

\[
\text{lambda } x: \text{numpy.tile}(x, [\text{sps},1]).T\text{.reshape}([1,\text{len}(x)\times\text{sps}]) \\
\text{If sps==4 then } \{1\} \rightarrow \{-1,-1,-1,-1\}
\]

\[
\text{lambda } x: \text{numpy.array}(
\text{numpy.exp}(1j*2*\text{numpy.pi}*((\text{dev} \times x \times \text{numpy.arange(len(x))}) / \text{samp_rate})), \text{dtype=}'\text{complex64}')
\]
(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
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
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 doesn't 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 …

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

http://keras.io/
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 ($y$)

De-noising Auto-encoders

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

Naively de-noising structured signals in GNU Radio!

WTF??
De-noising Auto-encoders

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

Input

Output

Representation
Convolutional Auto-Encoders

- Weights tied together at shift offsets! Less parameters to fit than fully connected layer.

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

- **Unknows**
  - 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

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 ...

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

Packet Bits
Modulated Signal

Training Loss per Epoch (demod)

UQ Plot of the 0-th signal

PSD of the 0-th signal (clean)

[[ 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!

- Generative models for representing and denoising 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?