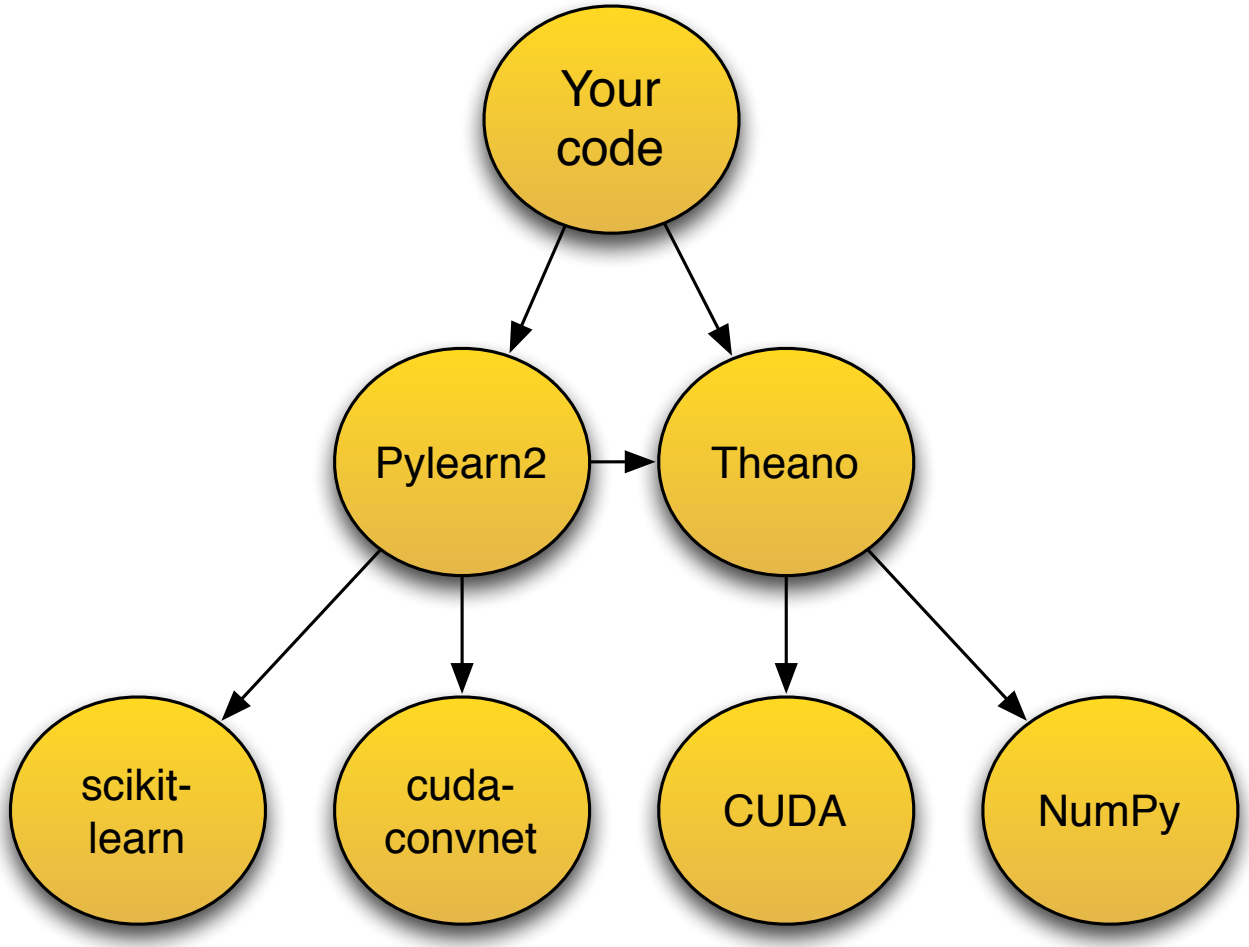
Theano and Pylearn2 for deep learning applied to computer vision

presented by Ian Goodfellow

Structure


Your code
Pylearn2
Theano
scikit-learn
cuda-convnet
CUDA
NumPy

NumPy

- Matrix computations
- Essentially Matlab as a python module
- Offers a C API
- C API requires Python reference counting, makes meaningful multitasking difficult

NumPy Example

```
>>> import numpy as np
>>> X = np.array([[1, 2],[3, 4]])
>>> Y = np.ones((2))
>>> print X
[[1 2]
 [3 4]]
>>> print Y
[ 1.  1.]
>>> np.dot(X, Y)
array([ 3.,  7.])
```

Theano

- Compiles mathematical expressions into C, CUDA, or OpenCL code
- Symbolic operations on expressions, like differentiation
- Numpy C API calls for many CPU operations
- CUBLAS or custom implementation of many GPU operations

Theano example

```
>>> from theano import tensor as T
>>> x = T.scalar()
>>> y = T.sqr(x)
>>> dydx = T.grad(y, x)
>>> dydx.eval({x: 2.})
array(4.0, dtype=float32)
>>> dydx.eval({x: 3.})
array(6.0, dtype=float32)
```

Advantages and disadvantages

- Advantages:
  - Python is easy to write and has lots of libraries
  - Optimizations (stability not just speed)
  - Symbolic manipulation
  - Easy to wrap C code (so can use Torch, cuda-convnet, etc)
- Disadvantages:
  - Poor support for arbitrary loops
  - Hard to be wrapped by C code
  - Very hard to do multithreading

Pylearn2

- Machine learning algorithms
- Interchangeable parts, most deep learning techniques decomposed into Cost, Model, and TrainingAlgorithm
- Designed to be extended. User is expected to subclass like crazy.
- Many deep learning reference implementations

Pylearn2 example

- Training a convolutional network on CIFAR-10 with maxout units:
- https://github.com/lisa-lab/pylearn2/blob/master/pylearn2/scripts/papers/maxout/cifar10_no_aug_valid.yaml

Advantages and disadvantages

- Advantages:
  - Combinatorial explosion of available methods (train RBM Model with denoising score matching Cost and nonlinear conjugate gradient TrainingAlgorithm)
  - Easy to change just one method to do your new research project
- Disadvantages:
  - Initial cost of learning to use it
  - Disadvantages inherited from Theano

Computer vision features

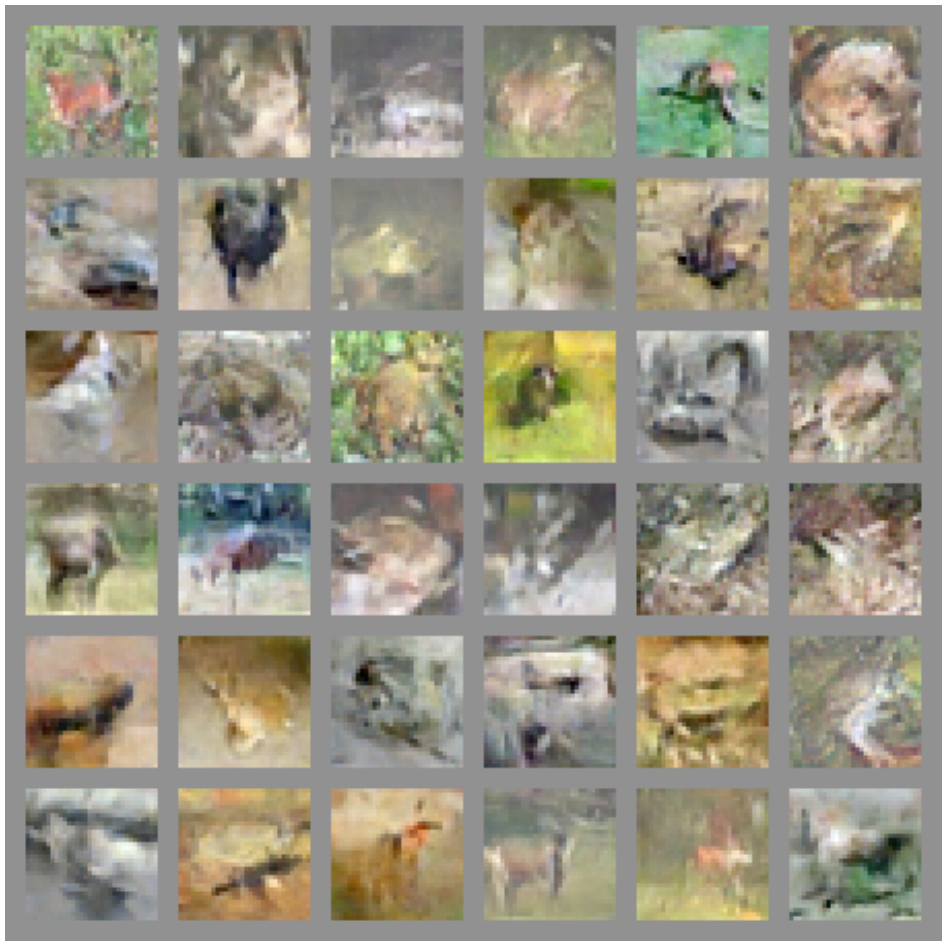
- Pylearn2 Space and LinearTransform classes allow polymorphism; instantly turn matrix multiplication on vectors into convolution on images
- pylearn2.datasets.preprocessing
  - Operations like whitening, local contrast normalization, patch extraction
- pylearn2.sandbox.cuda_convnet
  - Fast GPU convolution, "deconvolution", and max pooling
- Theano convolution
  - Slower but supports CPU and more input shapes

Computer vision achievements

- Maxout nets set the state of the art on MNIST, SVHN, CIFAR-10, and CIFAR-100 ("Maxout Networks", Goodfellow et al 2013)
- Won Emotiw face recognition challenge ("Combining Modality Specific Deep Neural Network Models for Emotion Recognition in Video", Ebrahimi et al 2013)
- Won Transfer Learning Challenge at NIPS workshops ("Scaling up spike-and-slab models for unsupervised feature learning", Goodfellow et al 2013)
- Won DARPA's Transfer Learning Challenge ("Unsupervised and Transfer Learning Challenge: A Deep Learning Approach", Mesnil et al 2012)
- State of the art face recognition on Toronto Face Database ("Disentangling Factors of Variation for Facial Expression Recognition," Rifai et al 2012)
- No ImageNet results yet

Natural image modeling

- Generative autoencoders, ssRBMs, adversarial nets, etc.



Community

- Get help:
  - theano-users@googlegroups.com
  - pylearn-users@googlegroups.com
- Report bugs, plan work on new features:
  - theano-dev@googlegroups.com
  - pylearn-dev@googlegroups.com
- These are community-developed libraries, pull requests are not just welcome but essential

Tutorials

- https://github.com/lisa-lab/pylearn2/tree/master/pylearn2/scripts/tutorials
- http://www.deeplearning.net/tutorial/
- https://github.com/goodfeli/theano_exercises