Neural Networks For Computer Vision

Kyle Kastner
INRIA Parietal
Université de Montréal

Neural Networks For Computer Vision

Kyle Kastner
INRIA Parietal
Université de Montréal

The Most Dangerous Game: Sloth Hunting 101

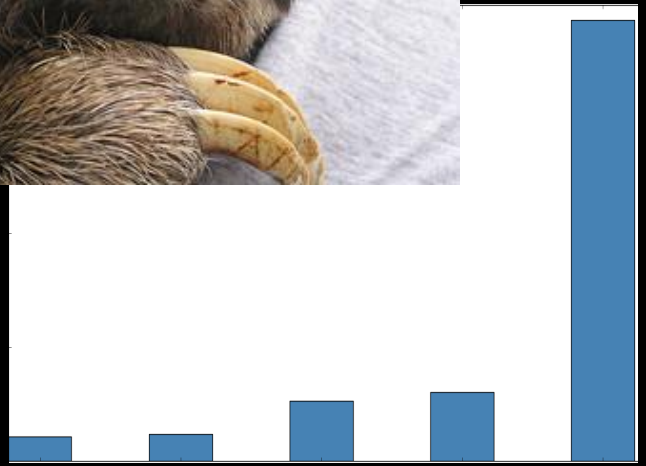
Kyle Kastner
INRIA Parietal
Université de Montréal

* No animals were harmed in making this talk
** *Please* don't hunt sloths

The Most Dangerous Game: Sloth Hunting 101

Kyle Kastner
INRIA Parietal
Université de Montréal

The Plan



Classify
Locate
???
Profit

Computer Vision

Fixed Feature Extractors

SIFT

HOG

SURF

skimage

Simple Classifier

linear models

SVM

sklearn

“Deep” Computer Vision

neural network

Trained together

neural network

Trainable Feature Extractor

Trainable Classifier

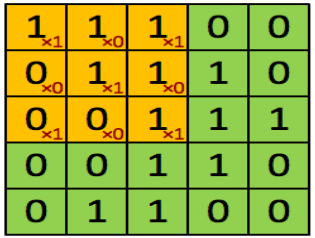
theano
pylearn2

Not pictured:
1.2 TB of images (ImageNet, 1.2E6 images)
compute cluster / monster GPU(s)

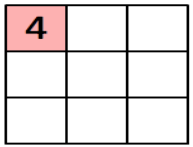
Concepts

- Neural networks are ***universal function approximators***
- Can train these to do (almost) whatever we want
  - Will it generalize to new data?
- Computer Vision: image descriptors
- Machine Learning: features

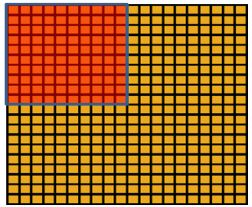
Exploiting Structure



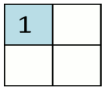
Image



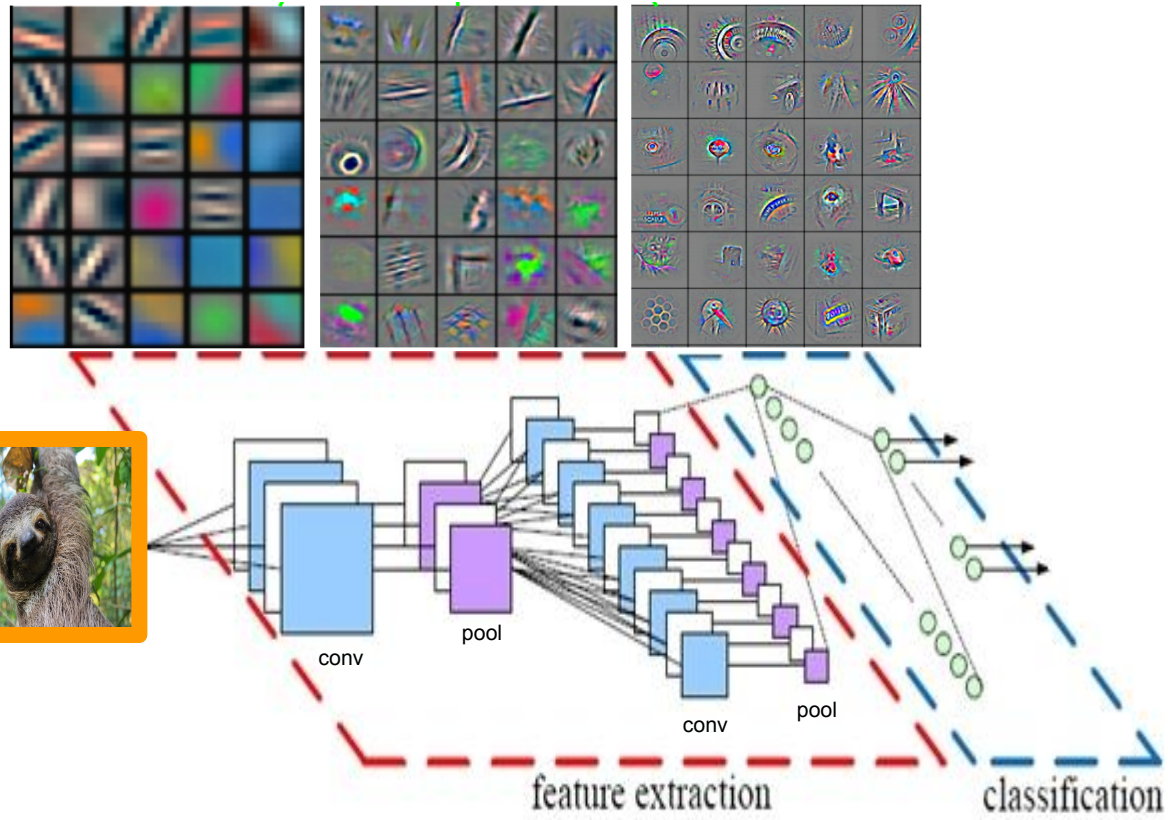
Convolved Feature



Convolved feature



Pooled feature



[1],[2],[3]

Technology

- Theano
- Optimizing compiler
- Lots of mathematical tools
- Many operations from numpy “just work”
- Automatic differentiation
- GPU or CPU set by single flag
  - THEANO_FLAGS=mode=FAST_RUN,device=gpu,floatX=float32

Cherry On Top

- pylearn2
- Built with Theano
- Used for neural network research
- Great guides to YAML interface
- Can code in pure python

https://github.com/kastnerkyle/pylearn2-practice

```
Monitoring step:
        Epochs seen: 33
        Batches seen: 16500
        Examples seen: 1650000
        learning_rate: 0.00999999046326
        momentum: 0.989998817444
        test_h2_kernel_norms_max: 1.87118041515
        test_h2_kernel_norms_mean: 0.804925084114
        test_h2_kernel_norms_min: 0.109122686088
        test_h3_kernel_norms_max: 1.90657293797
        test_h3_kernel_norms_mean: 1.3938267231
        test_h3_kernel_norms_min: 0.348352521658
        test_objective: 0.0377874299884
        test_term_0: 0.0261260885745
        test_term_1_weight_decay: 0.0116613674909
        test_y_max_max_class: 0.999999344349
        test_y_mean_max_class: 0.99490904808
        test_y_min_max_class: 0.76411986351
        test_y_misclass: 0.00859999842942
        valid_h2_kernel_norms_max: 1.87118041515
        valid_h2_kernel_norms_mean: 0.804925084114
        valid_h2_kernel_norms_min: 0.109122686088
        valid_h3_kernel_norms_max: 1.90657293797
        valid_h3_kernel_norms_mean: 1.3938267231
        valid_h3_kernel_norms_min: 0.348352521658
        valid_objective: 0.0444900207222
        valid_term_0: 0.0328286737204
        valid_term_1_weight_decay: 0.0116613674909
        valid_y_max_max_class: 0.999999344349
        valid_y_mean_max_class: 0.995154678822
        valid_y_min_max_class: 0.737455904484
        valid_y_misclass: 0.00819999724627
```

sklearn-theano

- Michael Eickenberg and myself
- Easy interface to trained networks
- Other Theano based models
- (Mostly) scikit-learn compatible
- Includes examples from this talk

https://github.com/sklearn-theano/sklearn-theano

sklearn-theano

OverFeat (NYU)[4]

pylearn2 models

Trained Feature Extractor

linear models

SVM

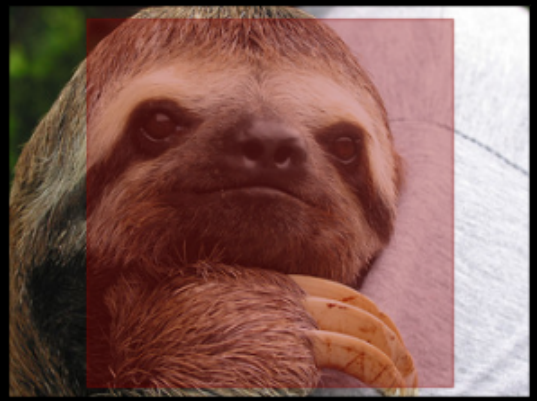
Trainable Classifier

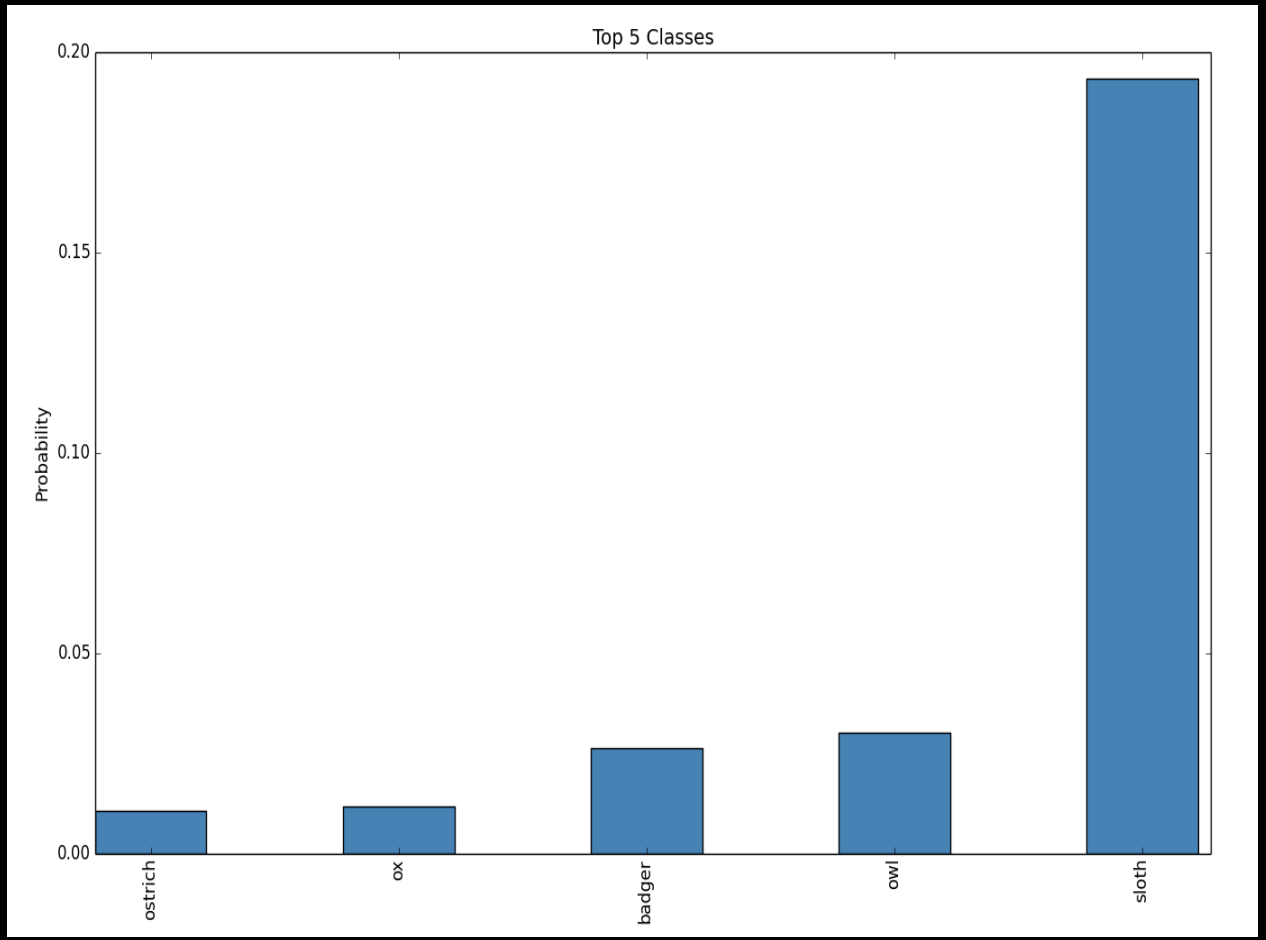
sklearn

Not pictured:

Weeks of expert time and compute time

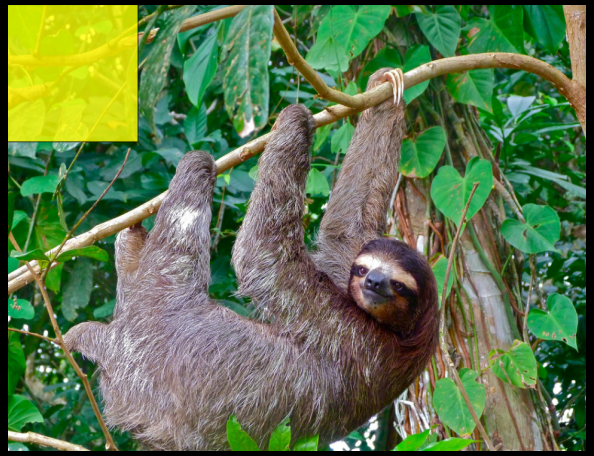
Cross-validation of entire networks!

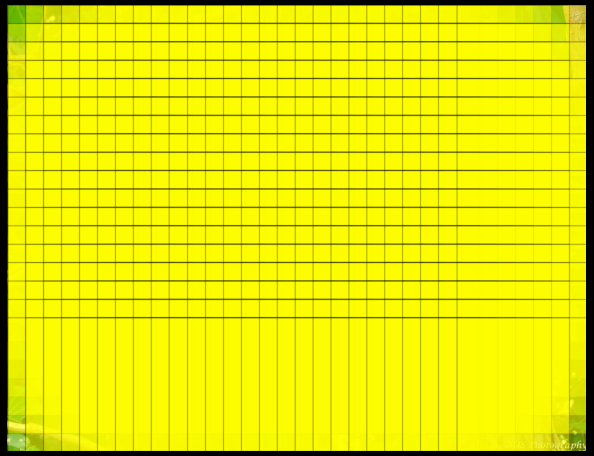




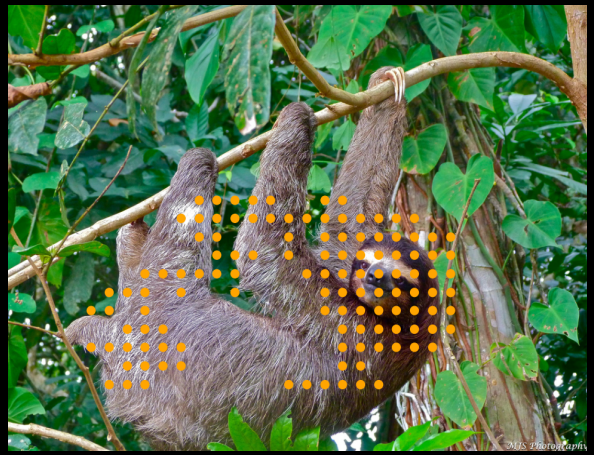
Classification

sklearn_theano/examples/plot_classification.py





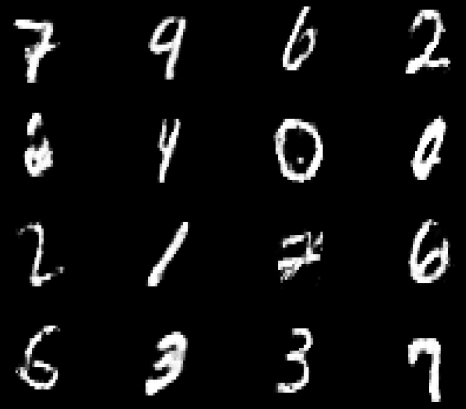
Localization

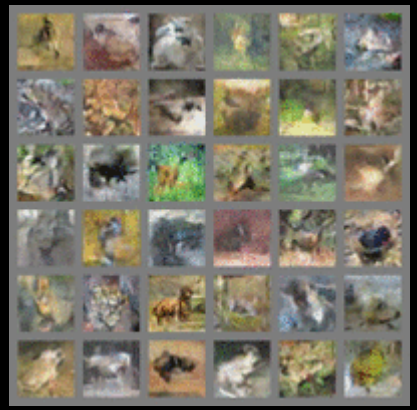


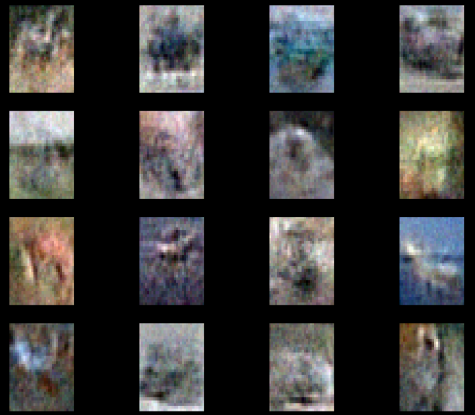
sklearn_theano/examples/plot_localization_tutorial.py

Generative Adversarial Networks

- Ian Goodfellow and a slew of others, 2014 [5]
- Trains two networks
  - Generate examples from random input
  - Discriminate whether generated or real







Generation sklearn_theano/examples/plot_mnist_generator.py

Takeaways

- Neural networks are good for many vision tasks!
- Theano is a great package for {C,G}PU math in Python
- pylearn2 is an awesome neural net package!
- ***Use*** large neural nets easily in sklearn-theano!
- Not every problem needs a massive neural network!
- But some do!

Slides:

https://speakerdeck.com/kastnerkyle/euroscipy2014
https://github.com/kastnerkyle/EuroScipy2014

Code:

https://github.com/sklearn-theano/sklearn-theano

Thank You!

Learning Materials: Courses

Coursera ML course, Andrew Ng

https://www.coursera.org/course/ml

This course is phenomenal, and will force you to *learn* machine learning including simple neural nets. Yes, it is MATLAB, sorry. Still worth it.

Coursera Neural Networks Course, Geoff Hinton

https://www.coursera.org/course/neuralnets

This course is a detailed look at modern neural networks by one of the core researchers in the field. Well worth it, and goes up to the edge of research.

Deep Learning Tutorial, LISA Lab, Yoshua Bengio

http://deeplearning.net/tutorial/

If you want to understand Theano and how to build neural nets, this is an excellent tutorial. After this you will understand the whole of my talk!

Neural Networks Course, Hugo Larochelle

https://www.youtube.com/playlist?list=PL6Xpj9I5qXYEcOhn7TqghAJ6NAPrNmUBH

This course will lock down every single piece of neural network approaches, and covers many important topics.

Learning Materials: Papers

Representation Learning, A Review and New Perspectives, Yoshua Bengio, Aaron Courville, Pascal Vincent

http://arxiv.org/abs/1206.5538

This paper covers much of the “nitty gritty” details. Weight initialization, optimization schemes, tradeoffs, and implementation. Fantastic!

Learning Features from Tiny Images, Alex Krizhevsky

http://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

The greedy layerwise training of this thesis is out of date now, but the rest of the insights into image statistics and intuition are very, very good.

ImageNet Classification with Deep Convolutional Neural Networks, Alex Krizhevsky, Ilya Sutskever, Geoff Hinton

http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks

This paper basically lays out the “modern” approach to most neural networks for images (also audio, and some NLP). OverFeat is basically an implementation of this architecture, with a number of custom tweaks to get better performance.

OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks, P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun

http://arxiv.org/abs/1312.6229

I like this paper (and associated code: https://github.com/sermanet/OverFeat) a whole lot. It inspired sklearn-theano!

Learning Materials: Blogs

Kaggle Galaxy Challenge Solution, Sander Dieleman

http://benanne.github.io/2014/04/05/galaxy-zoo.html

This blog post will show you how to build a neural network architecture from scratch for a custom dataset. His post on audio for Spotify is also spectacular: http://benanne.github.io/2014/08/05/spotify-cnns.html

Deep Learning in One File, Gabriel Synnaeve

http://snippyhollow.github.io/

This blog post has great, straightforward insight into deep learning, and provides a *single file* solution to do deep learning using Theano!

Neural Networks, Manifolds, and Topology, Chris Olah

http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

A whole blog full of remarkable discussions of neural networks, but this post gives some neat intuition about how and why neural networks work. This post on NLP is also great: http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/

Using Deep Learning To Listen For Whales, Daniel Nouri

http://danielnouri.org/notes/2014/01/10/using-deep-learning-to-listen-for-whales/

This model did not win, but the level of effort the winner put into feature engineering was immense. This approach did very well, very simply. Awesome!

References

[1] http://deeplearning.stanford.edu/wiki/index.php/UFLDL_Tutorial

[2] http://www.cs.nyu.edu/~yann/talks/lecun-ranzato-icml2013.pdf

[3] http://parse.ele.tue.nl/education/cluster2

[4] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun. *OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks*, International Conference on Learning Representations (ICLR 2014), April 2014. http://arxiv.org/abs/1406.2661

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. *Generative Adversarial Networks*, arXiV, June 2014. http://arxiv.org/abs/1406.2661

BONUS SLIDES




Preprocessing

sklearn-theano/examples/asirra_dataset_example.py

Code

```
tf = OverfeatTransformer(output_layers=[-3])
clf = LogisticRegression()
pipe = make_pipeline(tf, clf)
pipe.fit(X_train, y_train)
y_pred = pipe.predict(X_test)
```

class	precision	recall	f1-score	support
0.0	0.89	0.89	0.89	9
1.0	0.91	0.91	0.91	11
avg / total	0.90	0.90	0.90	20

accuracy score
0.9

Data Werewolves...

- But no silver bullets!
- Primarily for automation (WHAT)
- Other models can be better for
  - Scientific inference (WHY)
  - Simplicity of tuning (HOW)
  - Underpowered hardware (WHO)