Deep Learning Summer Workshop

Ver. 0.6

Boris Ginzburg (boris.ginburg@intel.com)

1 Introduction

This is practical, hands-on workshop on convolutional neural networks for undergraduate and graduate students. Course will combine a theory (30%) and practical work (70%). Course consists of two parts:

1. Introduction to CNN, and Caffe internals. In this part we will learn:
    a. how CNN works;
    b. how to use *caffe* for training of large convolutional networks;
    c. how to add to caffe new algorithms.
2. *Code Acceleration on CPU and GPU.* We will learn:
    a. how to accelerate code on GPU with CUDA,
    b. How to profile CPU code
    c. How to accelerate code with OpenMP and Intel MKL (Math kernel library).

The workshop will also have projects: development of visual classification applications, adding new algorithms to *caffe* etc.

1.1 Pre-requests

1. Good programming skills on C/C++ under Linux are required.
2. Preliminary knowledge of classical neural networks is plus, but not required

2 Part 1: Introduction to Convolutional NNs. Caffe Internals.

2.1 Introduction & Forward Propagation

9:00 – 10:00 Introduction to Convolutional NN
10:00 – 12:00 Caffe

- Installation & setup
- First example: MNIST-10

12:30 – 16:00 Code walk through forward-propagation (CPU)

- Data layer, Convolutional layer, Non-linear layer ReLU, Pooling, Fully-connected layer and Soft-max.
- Details of convolutional layer implementation:
    - convolution unfolding
    - groups

Homework:

1. Prepare overview of non-linear layers: logistic, tanh,…
2. Prepare overview of normalization layer
3. Study how MNIST accuracy depends on net topologies.
4. Port to caffe one of datasets http://deeplearning.net/datasets (NORB, SVHN,…)

Hands on tutorials:

1. https://code.google.com/p/cuda-convnet/
2. http://code.cogbits.com/wiki/doku.php
3. http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial

Links:

1. http://caffe.berkeleyvision.org/
2. Introduction: http://cs.nyu.edu/~fergus/presentations/nips2013_final.pdf
3. Krizhevsky et all, “ImageNet Classification with Deep Convolutional Neural Networks http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf
4. MNIST: http://deeplearning.net/tutorial/lenet.html, http://yann.lecun.com/exdb/lenet/
5. Chellapilla et all, “High Performance Convolutional Neural Networks for Document Processing”, http://hal.archives-ouvertes.fr/docs/00/11/26/31/PDF/p1038112283956.pdf

2.2 Backward Propagation

9:00 – 12:00 Introduction to CNN learning:

- Gradient-based learning for Multi-layer perceptron
- Back-propagation in Convolutional NN

13:00 – 16:00 Code walk through back-propagation (CPU)

- Data layer, Convolutional layer, Non-linear layer (Relu), Pooling, Fully-connected layer, Soft-max

Homework:

1. Train CIFAR-10 with different topologies.
2. Implement new non-linear layer from cuda-convnet: soft-relu

Links:

1. Back-propagation: http://ufldl.stanford.edu/wiki/index.php/Backpropagation_Algorithm, http://www.iro.umontreal.ca/~pift6266/H10/notes/mlp.html#the-back-propagation-algorithm
2. CIFAR-10: http://www.cs.toronto.edu/~kriz/cifar.html

2.3 Optimization Methods for CNN

9:00 - 12:00 Introduction to Gradient method

- Stochastic Gradient Descent (SGD)
  - adaptive learning rate
  - momentum
- SGD with line search
- Adagrad & AdaDelta
- Conjugate Gradient Descent
- Other methods
  - Limited memory BFGS
  - Levenberg-Marquardt
  - Nesterov accelerated gradient

13:00 – 16:00: Caffe: playing with SGD parameters for CIFAR-10

Exercise:

1. Experiment with SGD parameters for CIFAR-10 and Imagenet

Projects:

1. Implement following optimization methods:
   a. SGD with line search
   b. Conjugate gradient
   c. Adagrad/Adadelta

Links:

1. http://cseweb.ucsd.edu/classes/wi08/cse253/Handouts/lecun-98b.pdf
2. http://ufldl.stanford.edu/wiki/index.php/Gradient_checking_and_advanced_optimization
3. https://www.cs.toronto.edu/~hinton/csc2515/notes/lec6tutorial.pdf
4. http://videolectures.net/site/normal_dl/tag=12209/eml07_bengio_ssg_01.pdf
5. http://www.mscand.dk/index.php/daimipb/article/viewFile/6570/5693
6. http://www.stanford.edu/~acoates/papers/LeNgiCoaLahProNg11.pdf
7. http://www.ark.cs.cmu.edu/cdyer/adagrad.pdf
8. http://www.matthewzeiler.com/pubs/googleTR2012/googleTR2012.pdf
9. http://research.microsoft.com/pubs/192769/tricks-2012.pdf
10. R. Pascanu, "On the saddle point problem for non-convex Optimization", http://arxiv.org/abs/1405.4604
11. Dauphin, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization", http://arxiv.org/pdf/1406.2572v1.pdf

2.4 Regularization

9:00 - 12:00: Introduction to Regularization:

- Dropout
- Stochastic pooling
- Maxout

13:00 – 16:00: Caffe: playing with dropout layer

Projects:

1. Implement Stochastic Pooling & Maxout layer

Links:

1. Dropout www.cs.toronto.edu/~fritz/absps/imagenet.pdf
2. Stochastic pooling http://arxiv.org/pdf/1301.3557v1.pdf
3. Maxout http://jmlr.org/proceedings/papers/v28/goodfellow13.pdf

2.5 Unified Classification and Localization using Conv NN

9:00-16:00 Classification and Localization

- ILSCVRC – Classification and Localization challenge
- Overfeat
- Regions with CNN (R-CNN)

Exercise:

1. Implement Overfeat ”fast” net and train it

Projects:

1. Install R-CNN and try it: https://github.com/rbgirshick/rcnn (Requires Matlab!)
2. Build R-CNN detector in pure Pyhthon/C++.

Links:

1. http://cilvr.nyu.edu/doku.php?id=software:overfeat:start
2. www.cs.berkeley.edu/~rbg/slides/rcnn-cvpr14-slides.pdf
3. http://arxiv.org/abs/1312.6229

3 Part 2: Code acceleration on CPU and GPU

This is second, optional part of course, focused on SW optimization for CPU and GPU

3.1 Caffe: GPU implementation

9:00 – 12:00 Caffe GPU implementation

- Introduction to CUDA
- Café: CUDA internals

13:00 – 16:00 practical exercise on using CUDA:

- Implement 2D convolution
- implement convolutional layer

Homework:

1. Read Alex implementation of cuda-convnet2

Projects:

1. Re-implement caffe_gpu based on CuBLASXT / NVBLAS (CUDA 6.0).
2. Direct implementation of caffe convolutional layer using CUDA 6.0

Links:

1. http://www.nvidia.com/object/cuda_home_new.html
2. https://code.google.com/p/cuda-convnet2/

3.2 Caffe: CPU Optimization

9:00 – 12:00 Introduction to CPU optimization

- caffe performance analysis with Vtune
- Caffe and BLAS: ATLAS, OpenMP, and MKL
- OpenMP: introduction

13:00 – 16:00 practical exercises

- Vtune analysis of caffe
- OpenMP: add openmp to ReLU and Pooling layer

Homework:

1. Download openmp branch of caffe,
    a. Study convolutional layer with openmp
    b. analyze with Vtune
2. Identify additional layers where OpenMP makes sense

Links:

1. http://www.openblas.net/
2. https://software.intel.com/en-us/non-commercial-software-development
3. https://computing.llnl.gov/tutorials/openMP/
4. hattps://www.youtube.com/playlist?list=PLLX-Q6B8xqZ8n8bwjGdzBJ25X2utwnoEG
5. http://research.google.com/pubs/archive/37631.pdf

4 List of Big Projects

1. Grand challenge: http://image-net.org/challenges/LSVRC/2014/
2. Implement classification layers (e.g. SVM, kNN, ShareBoost)
3. Build detector and localizer based on caffe.
4. Extend Matlab and python wrapper for training.
5. Re-implement caffe_gpu based on CuBLASXT / NVBLAS (CUDA 6.0).
6. Direct implementation of caffe convolutional layer using CUDA 6.0
7. Projects for known datasets (http://www.csc.kth.se/cvap/cvg/DL/ots/):
    a. CIFAR-100 http://www.cs.toronto.edu/~kriz/cifar.html
    b. SVHN …
    c. Microsoft Coco: http://mscoco.org/
    d. Yahoo Flickr dataset http://yahoolabs.tumblr.com/post/89783581601/one-hundred-million-creative-commons-flickr-images-for