CS230: Experimental Neural Net Framework (ENNF)

Video presentation: http://www.youtube.com/watch?v=ldgj17GqxcO
Student: Konstantin Burichenko (kburc@stanford.edu, burichenko@gmail.com)
Mentor: Steven Jizhu Chen (stevenj@stanford.edu)

Motivation

There are a list of materials describes why Deep Learning is very powerful hammer in various engineering application. Original motivation of Deep Learning community was to solve important problems more than theoretical justification.

Unfortunately all this bring us to theory-practice gap in this area. One of them lies in understanding more of phenomenon of non-convex optimization. I talked with Dong-Pyo Park (world expert in math optimization) > 1 year ago and he said for me that unfortunately non-convex optimization in terms of fast methods has only reach situation with a lot of confusion when method works or not.

Personally I hope for future we will have more powerful tool developed in area of interaction of optimization community and machine learning community.

And I want to be part of it in future.

From other side there is a situation that prevalent programming language in Machine Learning is Python, and not C/C++ which is de-facto language in which almost all things in this planet have been written when we talk about speed.

Both this thing bring me to conclusion to create own framework in C++ which I hope will evolve in future.

I want give ability to use AI/ML tools not only for prototyping languages like Python or Matlab, but have ability to use them from C/C++ production code.

Software Implementation goal was:
1. General framework for people who is interesting in creating their own math optimization solvers
2. Various need things to organize work - load data from the disk to dense matrices and vectors, generate data with various p.d.f
3. All is written in compilable languages C/C++ and optionally exploits SSE2
4. Demonstrate how possible to do things outside TensorFlow.

Relate to math

Solvers used during learning/validate various well-known first order methods including batched version of SGD, ADAM, RMSProp, Polyak heavy ball method.

Reduction of Convex

Parameter Optimization and Convex Optimization Work with datastructures like SSD

Convex Linear Algebra

Basic Vector and Matrix

Applier methodologies allow use work with hardware with some microacceler.

Microelectronics

Quantum Physics

Inside different hierarch different things have been implemented:
- optimization algorithms
- data loading
- forward and backward pass
- own BLAS library
- different weight initialization schemas,
- initiation network with specific topology,
- network communication with plotter tool.

Experiments

Experimental Light Neural Net Framework consist of several parts:

In fact there is a stack of technologies and very high level libraries will really not allow tools development. Even interfaces is how Computer Science is evolving during last 10 years, but interfaces bring to misunderstanding limitations and possibility core un-obvious experiments.

Deep Neural Net

Parameter Optimization and Convex Optimization Work with datastructures like SSD

Convex Linear Algebra

Basic Vector and Matrix

Applier methodologies allow use work with hardware with some microacceler.

Microelectronics

Quantum Physics

Another question is possibility to have more fine time and References

1. Append support of connection neural nets
2. Append support of showing feature maps in plotter tool
3. Append support of visualizing configured Neural Net via graphlab
4. Try improve YOLO: Real-Time Object Detection
5. Append NVIDIA CUDA support.

(1) CS230 class from Stanford University
http://cs230.stanford.edu/ syllabus/
(2) Notes about Stochastic Gradient Descent from S.P.Bayl and J.C.Duchi
http://web.stanford.edu/class/ee368b/lectures/stoch_gradient_notes.pdf