ML/Hardware Co-design: Overview, Preliminary Result, and Open Opportunities

Ce Zhang (ce.zhang@inf.ethz.ch)
Machine Learning: Why should we care?
HOW TO BE COOL

... plus some other (equally important) reasons!
Machine Learning Needs @ ETH (Some Samples)

Kevin Schawinski, Astrophysicist
“I need to classify different types of galaxies”

Dirk Helbing, Social Scientist
“I need to learn what whales are talking about”

Sabine Schilling, Biologist
“I need to segment images and find cells”
ONE SIMPLE REASON

If people who have 20 Science & Nature papers think machine learning is important to treat us when we are sick, we’d better help them with that.
Overview of This Lecture

1. **(Whiteboard - 20 mins)** Overview of Machine Learning from a System Perspective
   - How many of you know Linear Regression?
   - How many of you know Support Vector Machine?
   - How many of you know Gradient Descent?
   - How many of you know Stochastic Gradient Descent?

2. **(15 mins)** A sample of our previous work related to hardware & machine learning
   - NUMA
   - CPU vs. GPU

3. **(10 mins)** Ongoing
   - ...
   - Low Precision Arithmetic
A Short Tutorial: Linear Regression

Patient → Predict → Survival Time
A Short Tutorial: Linear Regression

Patient

Features
- Age
- Weight
- Stage

Predict

Survival Time
# A Short Tutorial: Linear Regression

## Features
- Age
- Weight
- Stage

## Data

<table>
<thead>
<tr>
<th>Age</th>
<th>Weight</th>
<th>Stage</th>
<th>Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>105.5</td>
<td>2</td>
<td>250</td>
</tr>
<tr>
<td>54</td>
<td>106.2</td>
<td>4</td>
<td>70</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>64</td>
<td>107.2</td>
<td>1</td>
<td>?</td>
</tr>
</tbody>
</table>

## Patient

## Survival Time

### Predict

### Training

### Testing
A Short Tutorial: Linear Regression

Data

<table>
<thead>
<tr>
<th>Age</th>
<th>Heig</th>
<th>Stag</th>
<th>Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>5.5</td>
<td>2</td>
<td>250</td>
</tr>
<tr>
<td>54</td>
<td>6.2</td>
<td>4</td>
<td>70</td>
</tr>
<tr>
<td>64</td>
<td>7.2</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

...  

Model

\[ A \cdot x \]

Assumption

The prediction of \( r \)’th patient is

\[ A_r \cdot x \]

\( r \)’th row of \( A \)

Goal of Training

\[
\min_x \frac{1}{2} \sum_r (A_r \cdot x - b_r)^2
\]

Loss

(Sum of prediction errors)
More General than Linear Regression

Many machine learning tasks can be written as

$$
\min_x \sum_{r=1}^{N} l(x, y_r)
$$

Billions!
1. Gradient Descent
2. Stochastic Gradient Descent
3. Analytical Comparison between SGD and GD
4. Other Smarter ways to design the gradient estimator
   1. Mini-batch
   2. Low-precision
5. A Taxonomy of System Bottlenecks (Single Worker)
6. Distributed Setting
Low Precision Arithmetic
Limited Precision

Master Thesis Proposal
Communication-Scalable Machine Learning

Dan Alistarh  Ce Zhang
{dan.alistarh, ce.zhang}@inf.ethz.ch

Fall Semester 2016

1 Introduction

Stochastic Gradient Descent (SGD) is one of the standard tools in machine learning and data science, and represents the backbone of many large-scale learning systems, such as training Deep Neural Networks, or Stanford’s DeepDive. When running SGD at scale, one of the main concerns is to reduce the amount of communication between the nodes. Recently, several communication-efficient variants of SGD have been proposed, including lower-precision quantization (e.g., Google TensorFlow), one-bit quantization (e.g., Microsoft CNTK), or randomized quantization. The goal of this project is to investigate the trade-offs between reduced communication and precision of SGD training of state-of-the-art deep neural networks.

With the industrial success in building beefy machines such as NVIDIA’s DGX-1 (170 TFLOPs in a single box), our first hypothesis is that, on NVIDIA DGX-1, even with fast network such as NVLINK, we need network communications with limited precision to achieve reasonable scalability, for both synchronous and asynchronous scheduling. To validate this hypothesis, we need to first implement existing approaches on quantization, optimize their performance, and benchmark on standard tasks and datasets.

If this first hypothesis is true, one step moving forward is to build a low-precision version for each of the popular distributed Deep Learning systems. This goal is out of the scope of this master thesis proposal, but it represents one direction that the result of this master thesis is going to enable down the road. The hypothesis there is that, instead of revising each tool one by one, it is possible to build a new network protocol that has limited-precision primitives that all these distributed deep learning engines can rely on—In the ideal case, to implement a low-precision distributed deep learning system, we would simply change its network communication code to this new protocol, and need to do nothing else.

Single Bit Gradient?
NVLINK, DGX-1

Can we do low-precision data (not gradient)?
More difficult.
(Can you see why?)

Can we do near memory computation here?!
Linear Regression: 32-bit Data => 1-bit Data

Training Loss

# Iterations

32/1.5 = 21x Bandwidth Reduction?!

1.5bit “Double Sampling” (We get this curve today @5am :)

1-bit Random Rounding

32-bit

Better
Deep Learning: CPU vs. GPU
Single-Machine: CPU vs. GPU

On which machine should we run? CPU or GPU?

I have a GPU Cluster
I have 5000 CPU cores
I have $100K to spend on the cloud

Microsoft

EC2: c4.4xlarge
8 cores@2.90GHz
0.7TFlops

EC2: g2.2xlarge
1.5K cores@800MHz
1.2TFlops

Not a 10x gap? Can we close this gap?
Caffe con Troll

A prototype system to study the CPU/GPU tradeoff.
Same-input-same-output as Caffe.

http://github.com/HazyResearch/CaffeConTroll
What we found...

Relative Speed

Caffe CPU
CcT CPU
Caffe GPU

Caffe GPU
g2.2x_large
($0.47/h)
What we found...

Relative Speed

Caffe CPU
c4.4x_large
($0.68/h)

CcT CPU

Caffe GPU
g2.2x_large
($0.47/h)
What we found...

<table>
<thead>
<tr>
<th>Model</th>
<th>Speed</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caffe CPU c4.4x_large</td>
<td>0.28</td>
<td>$0.68/h</td>
</tr>
<tr>
<td>Caffe GPU g2.2x_large</td>
<td>1.1</td>
<td>$0.47/h</td>
</tr>
</tbody>
</table>
What we found...

Relative Speed

Caffe CPU: c4.4x_large ($0.68/h)
CcT CPU: c4.4x_large ($0.68/h)
Caffe GPU: g2.2x_large ($0.47/h)
Caffe CPU: c4.8x_large ($1.37/h)
CcT CPU: c4.8x_large ($1.37/h)

Proportional to FLOPs!
Four Shallow Ideas Described in Four Pages…
One of the four shallow ideas...

If the amount of data is too small for each core, the process might not be CPU bound.

For AlexNet over Haswell CPUs, Strategy 2 is 3-4x faster.
Ongoing Work
Where to run Machine Learning?
How fast can I train an SVM over 100GB data?

PCIe Gen2 4x 6sec = 1Epoch

How fast can I train SVM+LR over 100GB?

Jetson TX1
FPGA
Phone

Hypothesis 1: 2min on Jetson TX1

Hypothesis 2: Same as 1 SVM (Bandwidth Bound)

Go Beyond Deep Learning

Data Compression?

Model Compression (Hashing Trick for Training)?

Batching?

Which operations should happen on host/device?

Energy Efficiency?

Big Data in Small Pockets: A Vision
Precision-aware Network Protocol
How Should Limited Precision Primitives be Supported?

Low-precision Network Primitives? TCP/IP for ML?

- TensorFlow
- CNTK
- Caffe
- MxNET
- SINGA

Precision Aware Network Protocol

Network

Need FPGA to be fast?
How to Program Machine Learning?
Three Worlds: One System?

Machine Learning
- Tensor, Tensor, Tensor
- Real Algebra

Database
- Relation
- Relational Algebra

Spark
- Tuples
- Map Reduce

Logically, not different, all DataCube

Can we bridge the gap of physical representation?

Matrix Factorization
let matrix be float[]

let result be (P, Q) where
  P is float[] and
  R is float[]

COPY matrix from “matrix.tsv”

COPY result from `zero()`

v_{i,j} = P_{i,-} \times Q_{j,-}

loss = \sum_{i,j}(\sqrt(v_{i,j} – matrix_{i,j}))

min loss over P and Q

Real-Algebra Query

Partial Schema
(Static Analysis to figure out the rest?)

SQL/Spark

LaTeX-compilant
(Math should look like Math)
How to Guide user to use Machine Learning?
Are you serious?

What does it mean to be a good Cheat sheet? — HCI

=> I want a quantitative measure

If X then Y:
- How to give Premise?
- How to give Consequence?
How to take advantage Machine Learning?
More Intelligent/Robust Systems

What would happen if you ask today’s DB System a query that it was not programmed to answer?

DB System
Given q, Find d:
min |q ∧ dl|/|q ∨ dl|

WTF!@#$%#@ I don’t understand!!

SMT Solver to automatically figure out how to answer new queries

Simple, but we made the production system of a leading security company 100x faster

KEEP CALM AND TRY HARDER
Even Crazier

Can we beat human in generating Assembly?

Not a new dream “Super optimizer” (Henry Massalin 1987)

BUT we are at the right time to making it come true!
How to change the world with Machine Learning?
Applications

• **Cyber security**: FireEye & IBM Watson
• **Astrophysics**: ETH
• **Patient Stratification**: UZH Hospital & ETH
• **Genomics**: ETH
• **Aneurysm**: ZHAW
• **Social Science**: ETH
• **Dialogue System**: BMW

More to come!

How to support 100 applications with sublinear cost (development time, machine time, management time)?

Our final goal in the next K years.