TOWARDS DEEP LEARNING TECHNIQUES ON DISTRIBUTED SYSTEM

Presented by: Xiu Zhang
20161201
Outlines

• Significance
• Image Classification Task
• Tensorflow Introduction
• My Experience
• Experiments
SIGNIFICANCE
Deep Learning
- very good at extracting features
Distributed Computation
Is dnn fit for distributed learning?

• Data parallelism is obvious.
• Model Parallelism?

IMAGE CLASSIFICATION TASK
Image Classification Task
Image Classification Framework

Traditional recognition: “Shallow” architecture

Image → Hand-designed feature extraction → Trainable classifier → Object Class

Deep learning: “Deep” architecture

Image → Layer 1 → … → Layer N → Simple classifier → Object Class
Deep Learning for Image Classification

Image size large (Alexnet256*256[1], GoogleNet 256*256[2])

Image size small: don’t need many layers


Different Systems for Image Classification

- **Alexnet**: (f7 layer)

- **GoogleNet**: (p5 layer)
TENSORFLOW
INTRODUCTION
Tensorflow

• What’s Tensor?
  – a scalar, vector, matrix, Multidimensional array of numbers

• Graph-base Algorithm: Easy to distribute

• Session:
  – A session object encapsulates the environment in which Tensor objects are evaluated
  – Context, maintaining the states
Tensorflow

• Computation is a dataflow graph
Tensorflow

Edges are N-dimensional arrays: Tensors

- biases
- weights
- examples
- labels

Add → Relu → Xent

MatMul
Tensorflow (Device A : CPU, Device B: GPU)

http://www.slideshare.net/JenAman/large-scale-deep-learning-with-tensorflow
Single Process Configuration

![Diagram](image-url)

- **Client** → **Master**
  - Session
  - Run
  - Execute subgraph

- **Worker**
  - GPU\(_0\)
  - GPU\(_1\)
  - ...
  - CPU\(_0\)
Tensorflow Distributed System

Data Parallelism

\[ p'' = p' + \Delta p \]

Parameter Servers

\[ \Delta p' \]

\[ p' \]

\[ p'' \]

Model Replicas

Data

...
Distributed Configurations
MY EXPERIENCE
Why don’t use CaffeonSpark
CaffeonSpark Installation

• Apache Maven needs to be installed
• Environment Setting
  – Be careful for the rectification of the system files
  – etc/profile bashrc file
• Protobuf version 2.6
• Not enough support
Why choose Tensorflow?

• Very good support from google
• Relatively easy to implement:
  – Only to care about the configurations, not going into models like torch.
    ▪ Loop over GPUs
      ▪ Run mode[gpuX]:forward + criterion[gpuX]:forward
          +criterion[gpuX]:backward+mode[gpuX]:backward
    ▪ Accumulate GPUx’s gradParameters to GPU1’s gradParameters
    ▪ Do SGD on GPU1
    ▪ Copy back GPU1’s parameters to GPUx
Experiments

• Image Classification using Pretrained models: (Caffe)
  
• Distributed Deep Learning (Tensorflow)

• Self-designed DNN model for image classification (Torch)
Experiments

• Dataset: PASCAL VOC 2007 Dataset
  – 5011 Training/Validation
  – 4902 Testing
  – 20 object classes

• Classifier:
  – SVM : one-vs-all
Experiments

• Evaluation Protocol
  – Time & Precision Comparison
  – Precision
    • MAP (Mean Average Precision)
    • Confusion Matrix to show inter-class correlations
## Results:

<table>
<thead>
<tr>
<th>Index</th>
<th>Type of solver</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>L2-regularized logistic regression</td>
</tr>
<tr>
<td>1</td>
<td>L2-regularized L2-loss support vector classification</td>
</tr>
<tr>
<td>2</td>
<td>L2-regularized L2-loss support vector classification</td>
</tr>
<tr>
<td>3</td>
<td>L2-regularized L1-loss support vector classification</td>
</tr>
<tr>
<td>4</td>
<td>L1-regularized L2-loss support vector classification</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index</th>
<th>Mean Average Precision</th>
<th>AlexNet_fc7</th>
<th>Google_p5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>58.9</td>
<td>67.3</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>59.3</td>
<td>68.2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>61.7</td>
<td>72.5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>58.6</td>
<td>69.7</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>59.2</td>
<td>70.1</td>
<td></td>
</tr>
</tbody>
</table>
Results:
THANK YOU
Any QUESTIONS?