Distributed Neural Network Training and Data Flow Graph Construction

TensorFlow vs PyTorch

LSDPO (2017/2018)
Open-source project presentation: Ioana Bica (ib354)

Motivation

- Large number of applications nowadays use machine learning
- Large number of machine learning frameworks available

- How to choose the correct one for your application? Decision may depend on:
  - how easy can it be used to prototype new models?
  - does it have support for distributed training?

TensorFlow vs PyTorch

- Originally developed by Google Brain.

- Static Computational Graph.
- Client needs to define the entire computational graph before running it.

- Developed as a collaboration between companies and universities.

- Dynamic Computational Graph.
- Define nodes and execute them on the go.

Sequence tagging

Example: Named entity recognition

Mark lives in Cambridge.

person *location*

TensorFlow is a machine learning framework developed by Google.

misc *organization*

Sequence Tagging

TensorFlow

- Need to pad all sentences to have the same lengths.
- Wastes a large number of parameters.
- Output sequences also need to have same length.
- Produce unnecessary outputs.

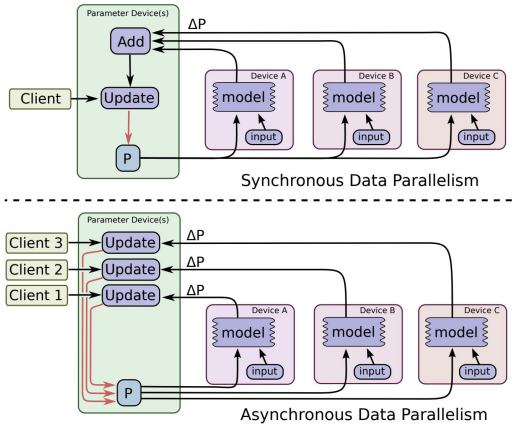
PyTorch

- Length of input and of the output depends on the length of the sentence being processed.

Distributed Neural Network Training

- Important for developing and training state-of-the-art neural network models.

- Explore data parallelism (replicated training):
  - Synchronous/Asynchronous Stochastic Gradient Descent
  - TensorFlow was designed to support distributed training
  - PyTorch has recently added this functionality



Abadi et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems

Evaluation

- Iteration time and training time.
- Overhead in TensorFlow in order to support variable size inputs.
- Performance on distributed training.

- Use Amazon EC2 for evaluating distributed training.

Steps

1. Become acquainted with PyTorch syntax.
2. Decide on example dataset to illustrate differences in data flow graph construction.
3. Build and evaluate models for each framework.
4. Set up Amazon EC2.
5. Distribute models and evaluate performance.
6. Extension: Explore TensorFlow Fold.

Thank you!
Questions?