TensorFlow Tutorial

Western New York Image and Signal Processing Workshop 2017

Syed Ahmed
KGCOE Computer Engineering Department
Rochester Institute of Technology

syed-ahmed.github.io/tensorflow-tutorial

Agenda

TensorFlow Concepts

TensorFlow 101

Software Development Workflow

Profiling your code

Multi-GPU execution

Inference Pipeline

How do I learn TensorFlow?

- Pick a paper without code
- Implement the code
- Publish in GitHub
- Tweet/Email to the Author

TensorFlow Concepts

TensorFlow 101

Software Development Workflow

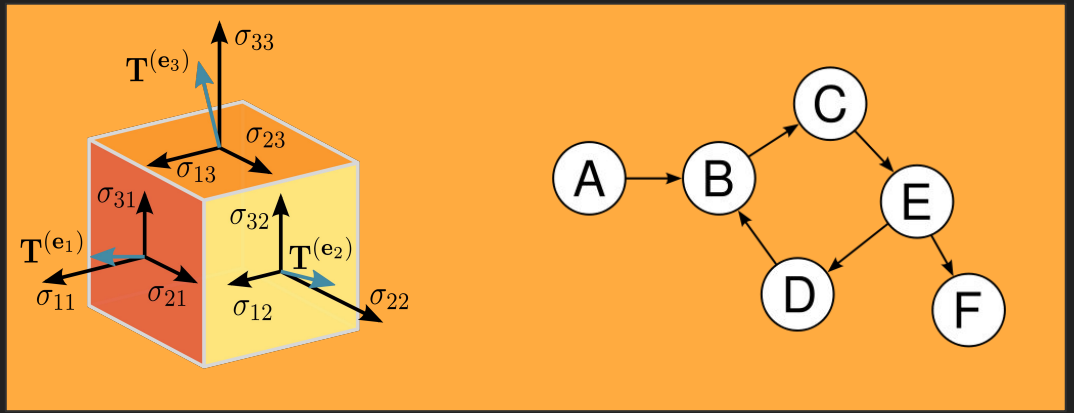
Profiling your code

Multi-GPU execution

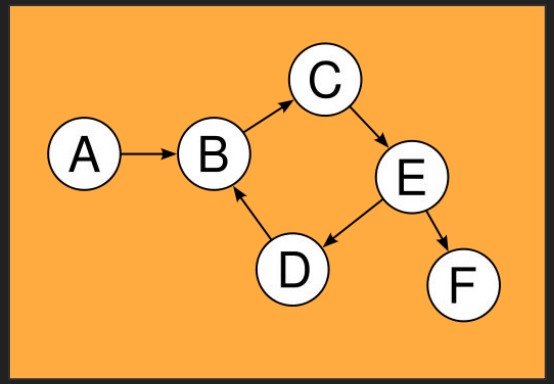
Inference Pipeline

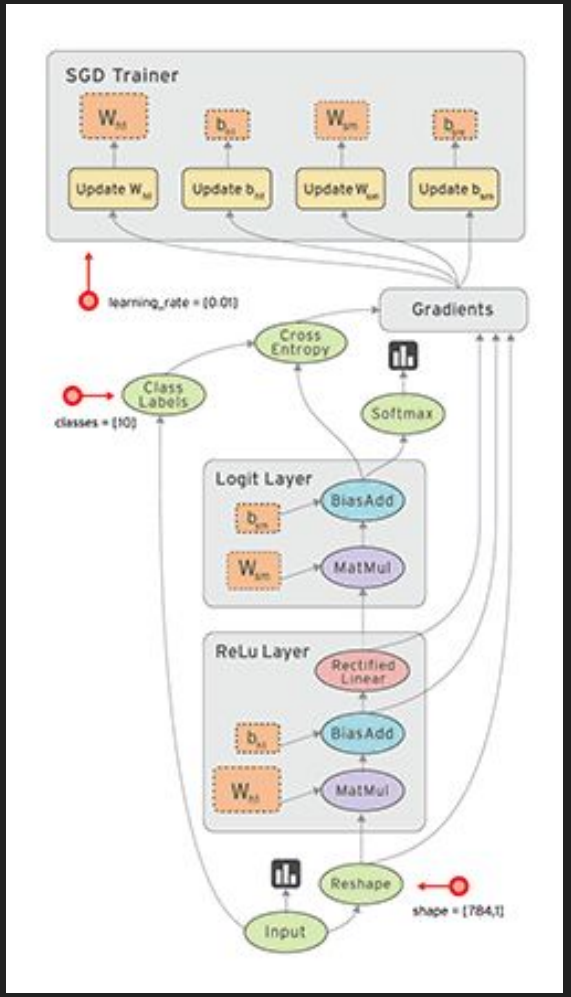
What's in the name?

{Tensor}{Flow}



Data Flow


A
B
C
D
E
F


SGD Trainer
Update W
Update b
Update W
Update b
learning_rate = (0.01)
Gradients
Cross Entropy
Class Labels
classes = (10)
Softmax
Logit Layer
BiasAdd
MatMul
ReLu Layer
Rectified Linear
BiasAdd
MatMul
Reshape
shape = [784,1]
Input

Data Flow Programming and Architecture

Dataflow programming

From Wikipedia, the free encyclopedia

In computer programming, **dataflow programming** is a programming paradigm that models a program as a directed graph of the data flowing between operations, thus implementing dataflow principles and architecture. Dataflow programming languages share some features of functional languages, and were generally developed in order to bring some functional concepts to a language more suitable for numeric processing. Some authors use the term Datastream instead of Dataflow to avoid confusion with Dataflow Computing or Dataflow architecture, based on an indeterministic machine paradigm. Dataflow programming was pioneered by Jack Dennis and his graduate students at MIT in the 1960s.

Contents [hide]

Dataflow Machine Architecture

ARTHUR H. VEEN

Center for Mathematics and Computer Science, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Dataflow machines are programmable computers of which the hardware is optimized for fine-grain data-driven parallel computation. The principles and complications of data-driven execution are explained, as well as the advantages and costs of fine-grain parallelism. A general model for a dataflow machine is presented and the major design options are discussed.

Most dataflow machines described in the literature are surveyed on the basis of this model and its associated technology. For general-purpose computing the most promising dataflow machines are those that employ packet-switching communication and support general recursion. Such a recursion mechanism requires an extremely fast mechanism to map a sparsely occupied virtual space to a physical space of realistic size. No solution has yet proved fully satisfactory.

A working prototype of one processing element is described in detail. On the basis of experience with this prototype, some of the objections raised against the dataflow approach are discussed. It appears that the overhead due to fine-grain parallelism can be made acceptable by sophisticated compiling and employing special hardware for the storage of data structures. Many computing-intensive programs show sufficient parallelism. In fact, a major problem is to restrain parallelism when machine resources tend to get overloaded. Another issue that requires further investigation is the distribution of computation and data structures over the processing elements.

Categories and Subject Descriptors: A.1 [**General Literature**]: Introductory and Survey; C.1.2 [**Processor Architectures**]: Multiple Data Stream Architectures—*multiple-instruction-stream, multiple-data-stream processors (MIMD)*; C.1.3 [**Processor Architectures**]: Other Architecture Styles—*data-flow architectures*; C.4 [**Computer Systems Organization**]: Performance of Systems—*design studies*

General Terms: Design, Performance

Additional Key Words and Phrases: Data-driven architectures, dataflow machines, data structure storage

TensorFlow Concepts

TensorFlow 101

Software Development Workflow

Profiling your code

Multi-GPU execution

Inference Pipeline

TensorFlow Concepts

TensorFlow 101

Software Development Workflow

Profiling your code

Multi-GPU execution

Inference Pipeline

TensorFlow 101

Follow Jupyter Notebook

TensorFlow Concepts

TensorFlow 101

Software Development Workflow

Profiling your code

Multi-GPU execution

Inference Pipeline

TensorFlow Concepts

TensorFlow 101

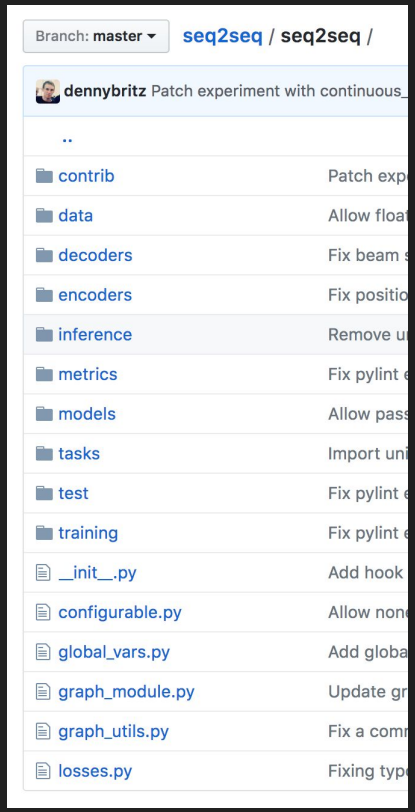
Software Development Workflow

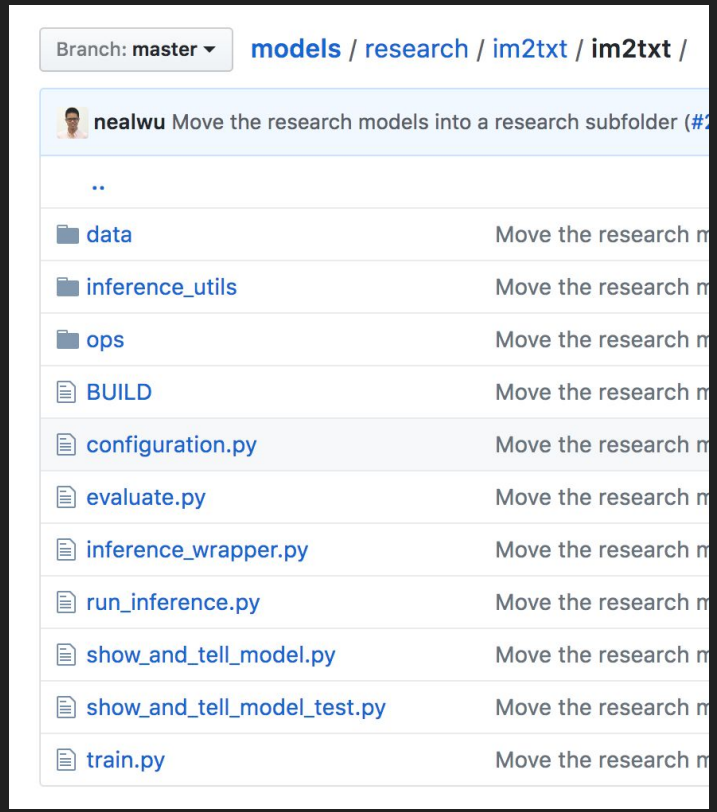
Profiling your code

Multi-GPU execution

Inference Pipeline

Structuring your project





https://github.com/tensorflow/models

Docker Containers for Reproducibility

Follow link in Jupyter Notebook

Debugging

https://wookayin.github.io/tensorflow-talk-debugging/#1

TensorFlow Concepts

TensorFlow 101

Software Development Workflow

Profiling your code

Multi-GPU execution

Inference Pipeline

TensorFlow Concepts

TensorFlow 101

Software Development Workflow

Profiling your code

Multi-GPU execution

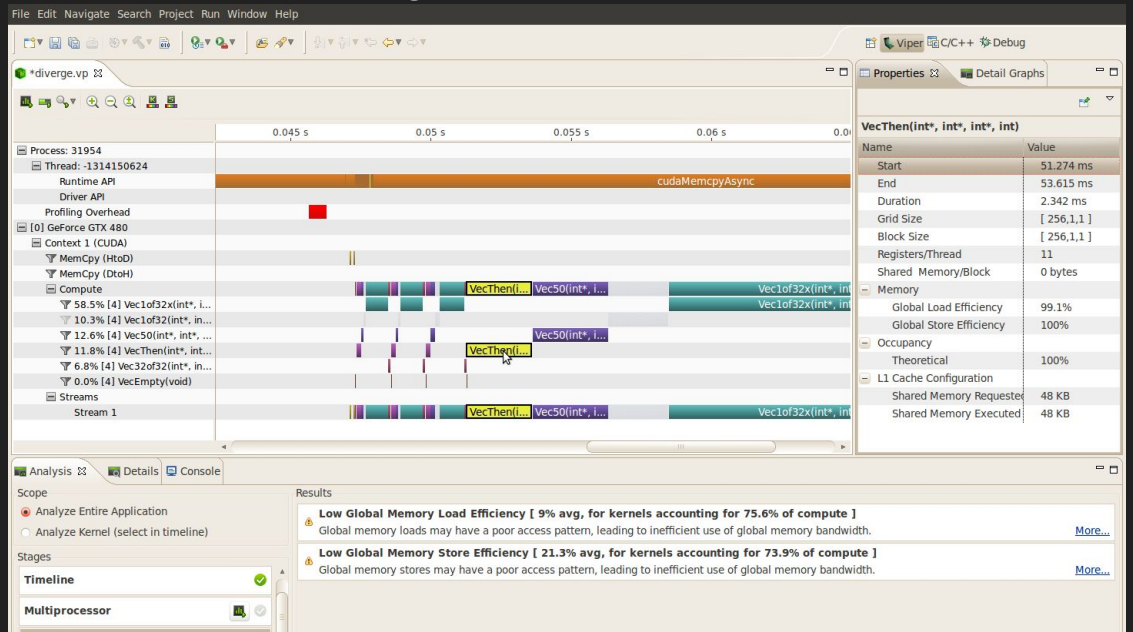
Inference Pipeline

TF Profiler

Follow on Jupyter Notebook

NVIDIA Visual Profiler

1. nvprof -o out.nvvp python mnist_deep.py
2. Open out.nvvp in NVIDIA Visual Profiler
3. Optimize code to minimize gap in compute



TensorFlow Concepts

TensorFlow 101

Software Development Workflow

Profiling your code

Multi-GPU execution

Inference Pipeline

TensorFlow Concepts

TensorFlow 101

Software Development Workflow

Profiling your code

Multi-GPU execution

Inference Pipeline

Multi-GPU Training

1. Documentation

TensorFlow Concepts

TensorFlow 101

Software Development Workflow

Profiling your code

Multi-GPU execution

Inference Pipeline

TensorFlow Concepts

TensorFlow 101

Software Development Workflow

Profiling your code

Multi-GPU execution

Inference Pipeline

Inference on Android

1. Google Codelabs
2. TensorFlow Lite

Thank You!

sxa1056@rit.edu

Questions?