Things to know while choosing a Deep-Learning Library

Saurabh Agarwal
Machine Learning Engineer
MAD Street Den

MAD™
MAD STREET DEN

Why DL libraries ?

- Quick Prototyping
- Reducing Engineering and Focus on building algorithms.

MAD™
MAD STREET DEN

What can I choose from ? (there are many more)

MAD™
MAD STREET DEN

Confused ?

- Library - Library everywhere not a clue which one to use.
- Everybody claims theirs is faster or easier to use.

MAD™
MAD STREET DEN

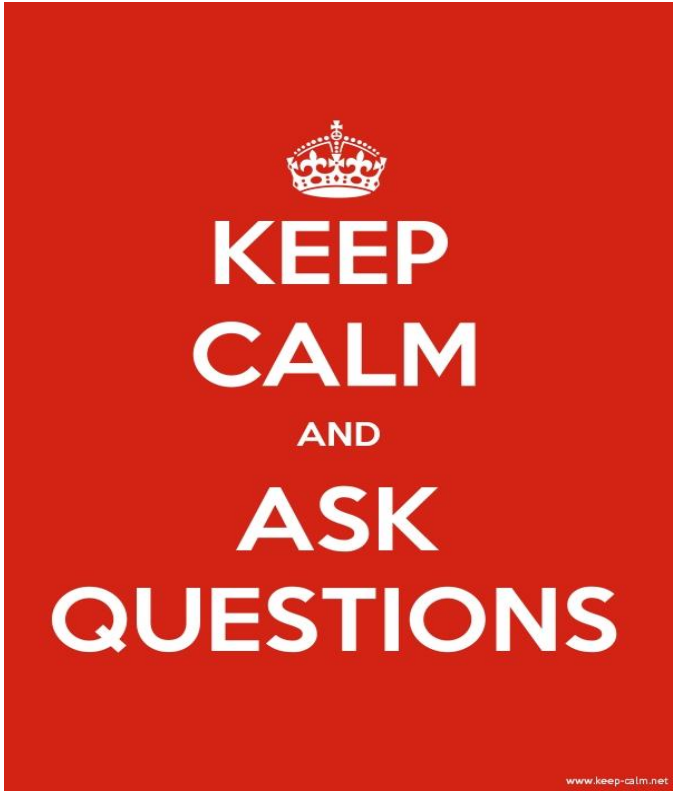
Structure of the talk

1. Differences and what those differences mean ?
2. State of some popular ML libraries
3. Example scenarios with intended use cases.

MAD™
MAD STREET DEN

Take a step back ?

- Fundamental Questions -
  - How do they differ ?
  - What benefits do I get because of these differences ?



MAD™
MAD STREET DEN

Language the framework is written

- Python - (Theano)
- C++ - (Caffe)
- Lua - (Torch)

1. Frameworks written in Python usually are slower. But easy to modify for a custom change.
2. Most of them have Python-API. So you get the speed of low-level language and ease of coding in python

Community Support and Documentation

- How active is the community ?
- Is the project under active development ?
- Is the project stable ?
- Are there any big companies supporting it ?
- Documentation ?

Graph ?

- What is a graph ?
  - Typically the frameworks maintain a computational graph.
  - It specifies the sequence of operations to perform.

- Types of Graph:
  - Static - Define and Run
  - Dynamic - Define by Run

More on Static Graph

- First a graph is defined in host language (ex- Python) .
- A second interpreter that executes the graph.
- Pros:
  - Optimization
  - Utilization of SIMD (single instruction , multiple data)
  - Overall-Faster at runtime
- Cons:
  - Slow compile time
  - Difficult to debug, because of two interpreter structure
  - Almost impossible to build certain recursive neural net architecture

MAD™
MAD STREET DEN

More on Dynamic Graphs

- Every iteration a new graph is built
- The states of older variables is saved in the graph for next iteration
- Pros:
  - Complex recursive architectures
  - Easy Debugging - standard python tools like pdb work
- Cons:
  - Slower at runtime
  - High memory consumption

MAD™
MAD STREET DEN

Using a High-Level API

- Toolkits like Keras, Lasange
- Provide simplified API's to allow quick prototyping
- Use other libraries in the backend .
- Pros:
  - Helper functions to handle grunt work
- Cons:
  - Another level of Abstraction.

MAD™
MAD STREET DEN

Inference Speed on GPU

- Speed usually is almost same because of CuDNN
- The delta can come because of graph creation overhead

State of Some Popular Libraries

Tensorflow

- The Good things:
  - Excellent Community support
  - Stable Release
  - Under Active Development
  - Google's Support
  - Lot pretrained Networks and Open Source implementation of Networks
- The not so Good things:
  - Static Graph Library
  - Overwhelming

MAD™
MAD STREET DEN

Caffe

- The Good things
  - Very Fast
  - Lot of Pretrained Networks
  - Great Community Support
- The not so Good things
  - C++, very difficult to modify

Theano

- Good Things
  - Good Community Support
  - Python mostly
  - Allows you to do a lot of crazy stuff
- The not so Good things
  - Slow at inference
  - Single GPU
  - Graph is very low level

Chainer

- The Good things
  - First to support Dynamic Graphs
  - All benefits of Dynamic Graphs
  - Highly scalable multi-gpu version
- The not so Good things
  - Slower than other dynamic libraries
  - Small Community

Dynet

- The Good things
  - Almost as fast as static libraries on CPU
  - Python bindings
  - Very Pythonic API Structure
- The not so Good things
  - Small user group
  - Not backed by any major company

PyTorch

- The Good things:
  - Torch without Lua
  - Dynamic Graph
  - Promises Numpy on GPU
  - Amazing Community
- The not so Good things
  - Still in Beta
  - Can be slow at times

MAD™
MAD STREET DEN

Keras

- Good things
  - Amazing Community Support
  - Lot of built in functionalities
  - Simple API
- Not so Good things
  - Depends on the library used in the backend

MXNet

- Good things
  - Apache Supported Project
  - Fast
  - Under Active Development
  - Great Documentation
  - AWS Supports

MAD™
MAD STREET DEN

What works for one might not work for another

Intended Usage

- Research vs Production
  - For Research :
    - Ability for rapid prototyping
    - Ease of debugging
    - Speed takes a backseat
  - For Production
    - Speed takes prime importance
    - Ability of Library to Scale

MAD™
MAD STREET DEN

Example Scenario

Beginner

- Easy API
- Lot of support for quick saving and loading of models
- Good Documentation
- Community Support
- Better off using toolkits like KERAS

Researcher

- Likes to prototype quickly to test new and crazy hypothesis
- Ease of debugging
- Doesn’t care much about speed
- Dynamic Graph libraries like Chainer and Pytorch make more sense

MAD™
MAD STREET DEN

Practitioner in Industry

- Cares about speed
- Availability of pre-trained models
- Adoption by other companies.
- Static Graph libraries like Caffe, Tensorflow and Caffe2 makes more sense
- Also if you do deployments on a GPU then the differences can be negligible, CUDNN at work
- A hybrid approach is usually a good idea. Development in a dynamic graph library and deployment in static libraries

MAD™
MAD STREET DEN

Multi Platform Deployment

- Tensorflow shines over everybody.

MAD™
MAD STREET DEN

General Advice

- Get started with something you feel comfortable.
- Move around as and when situation demands
- Community Support
- You get hang of one type everything else makes sense

MAD™
MAD STREET DEN

THANK YOU!!

MAD™
MAD STREET DEN