Module 35

Tensorflow/Pytorch Tutorials

Slides by:

Team DSS

Amanda Ma, Kevin Miao, Sandeep Sainath, Youngli Hong

Week

8

Introduction to TF/Torch

After this module, you should independently be able to:

1. get an intuition of the differences between the two libraries.
2. transition between programming, math and theory, specifically, the different optimizations and gradients.
3. load and augment data.
4. independently use any of the libraries to implement neural models of choice.
5. be able to effectively debug and read through documentation.
6. get some sense of residual nets are implemented on a basic level.

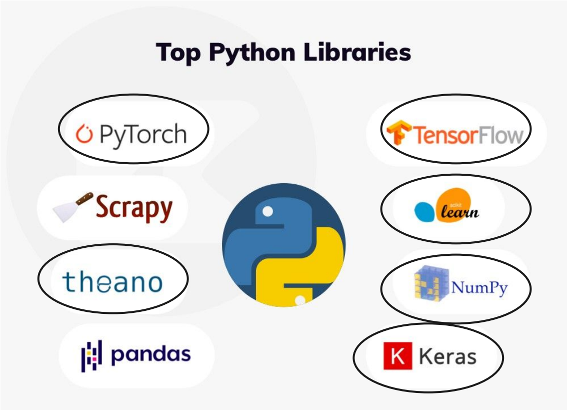
Prerequisite knowledge:

1. EE16A
2. CS61A/B or equivalent programming experience
3. Module 21 - Backpropagation
4. Module 23 - Hyperparameter Tuning
5. Module 34 - CNNs
6. Module 35 - Transfer Learning
7. EE16B *(Concurrently)*

History of ML Libraries

History of ML Libraries

- **1991:** Python invented
- **1995:** Numeric, predecessor NumPy, launched
- **2000s:** SciPy
- **2006:** NumPy
- **2007:** Theano, Scikit-Learn
- **2010:** Deep Learning skyrocketing due to cheap and many GPUs
- **2015:** Keras, TensorFlow
- **2016:** PyTorch



Objective: Faster, more-intuitive to create scientific computing tool through smart data structures and architecture

Current Deep Learning Frameworks

Current Deep Learning Frameworks

Library/ Framework		theano		
Developer/Year	Google Brain (2015)	University of Montreal (2007)	Facebook AI (2016)	Francois Chollet (2015)
High/Low Level	High/Low	Low	Low	High (Based on TF)
Strength	Large-Scale Employment	Fast, Prototyping	Dynamic	High Level

Tensorflow

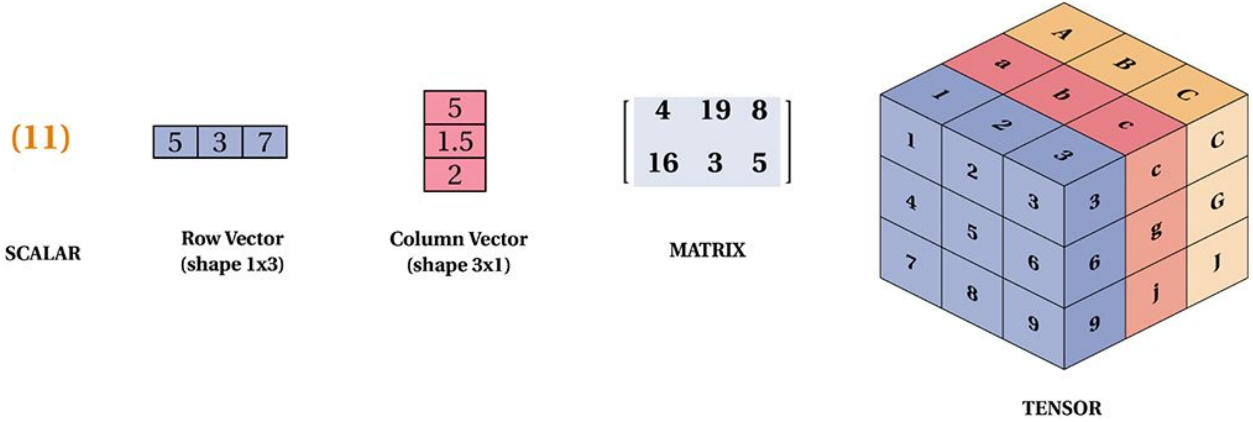
What is and Why Tensorflow?

Deep Learning library open-sourced by Google

- Deep Learning -- neural networks!
- Distributed Computing & Scalability
- GPU/TPU Support & Pipelining
- Flexibility

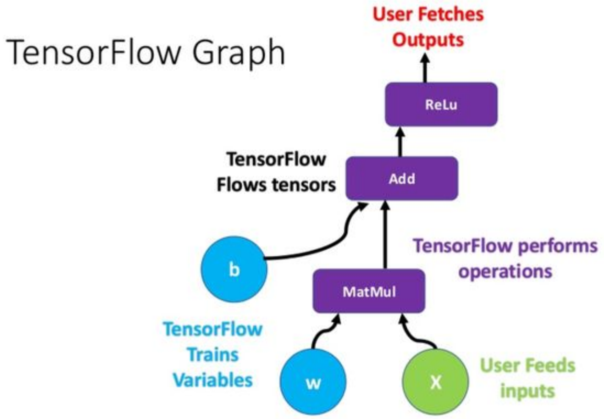
Tensors

N-dimensional array of data

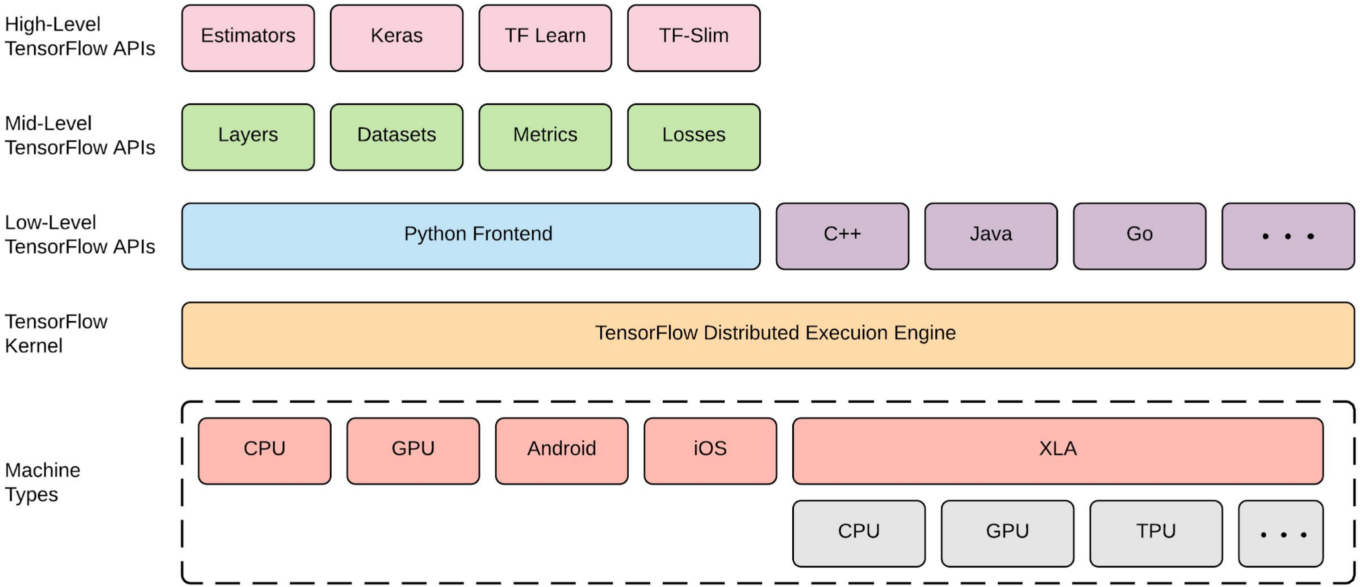


Graphs

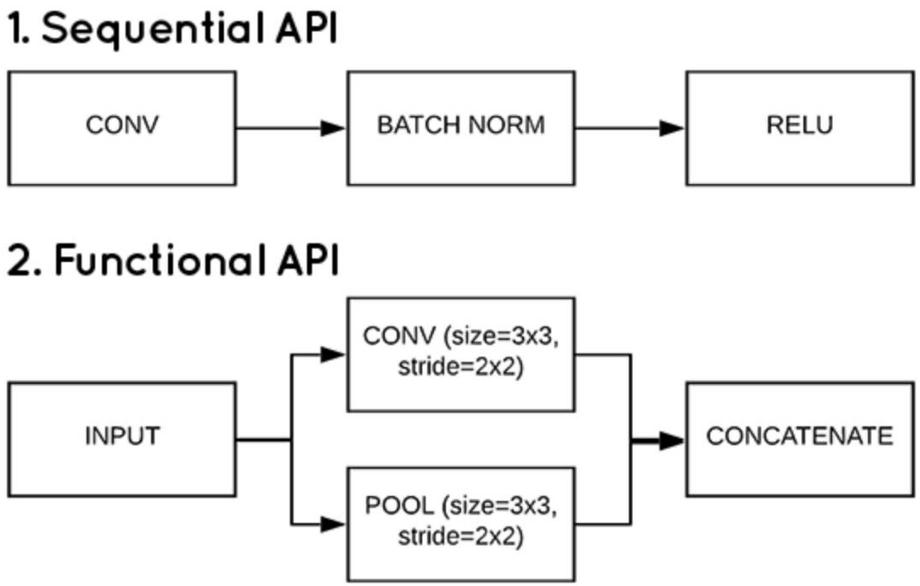
Represent computations as dependencies between various operations



Abstraction Layers



Sequential API vs Functional API



Work on
Assignment 1A

Work on

Assignment 2A

Pytorch

PyTorch

- PyTorch is **open-source machine learning framework**
- Created by Facebook AI in **2016**
- More dynamic and pythonian
- Data parallelism
- Easier navigation
- Majority of researchers, **Tesla’s autopilot** and **UBER pyro**

PyTorch Tensors, Dimensionality and NumPy

- PyTorch is extremely similar to **NumPy**
- Tensors, unlike in TF, are used in a similar fashion as NumPy’s Arrays
  - Arithmetics are the same
  - Mutations are the same
- Slightly different syntax

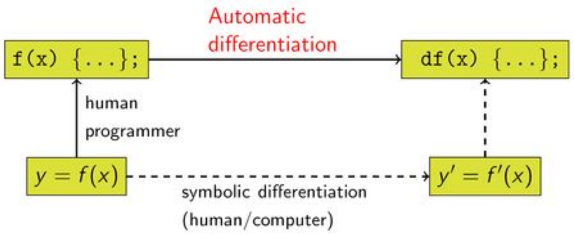
Data Utilities

- Datasets are also custom classes
- These classes need to contain the following structure:
    - **Initializer(data, labels, transforms):** need to create class variables storing data and labels or alternatively file paths
    - **__get__(idx):** Return one sample located at x
    - **len:** the length of your dataset
- Training can happen through using a one-liner **data loading**

Neural Networks

- Neural Networks and (generally DAGs) encapsulated in **classes** in PyTorch
- **Library:** `torch.nn`
- These classes need to contain the following structure:
    - **Initializer:** inherit from super class
    - **Forward(input):** calculations when neural network is called on input
    - **num_flat_features() [OPTIONAL]:** sometimes you need to flatten something and it is nice to have.

Autograd



- PyTorch tensors allow for the fast computation of **derivatives** through **autograd,** automatic differentiation framework
- Automatic Differentiation vs Symbolic Differentation
- To use **autograd,** initialize tensor with `requires_grad=True` and take gradient with `.backward()` and `tensor.grad`
- Detach using `tensor.detach()

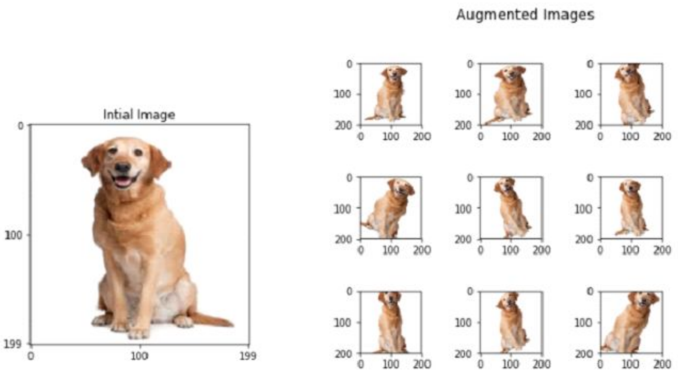
Optimizers

- **Optimizers,** a smart optimization to do **backprop** for you!
- **`torch.optim`**
- Workflow:
  - Initialize **optimizer** with **coefficients, learning rate**, **momentum factor**
    - **Momentum factor:** degree of using previous gradients for directionality
  - Define **loss function (also called criterion)**
  - Run model on **input**
  - Calculate **loss**
  - Call `.backward()` on **loss**

Work on

Assignment 1B

Data Augmentation



- How do we prevent **overfitting** and **increase our dataset?**
  - Data Augmentation
  - i.e. from one picture we can create 10 pictures for training
- **Transforms** in your **dataset** class allow you to do that.
  - Pre-defined functions: https://pytorch.org/docs/stable/torchvision/transforms.html
- **Disclaimer:** If you crop or translate, make sure that the region of interest is still in the picture.

CUDA

- Parallel Computing Platform to allow GPUs for general processing
- To use GPUs, we need to send to **DEVICE** first
  - Model
  - Data/Labels
- Define a **device**
- Sending data through
  - `tensor.to(device)`
- If you want to extract the values, you simply append `.data`

ResNet

- We will not go too much in-depth about the mechanics behind Residual Nets, but you are encouraged to use this in your assignments.
- State-of-the-art
- Deep Learning Networks became deeper and deeper (yet also shrunk in filter sizes)
- Researchers found out that the residual block in a CNN is useful
- Not entirely sure (still being researched)
- **Different flavors:** ResNet 18, 34, 50, …., 1202

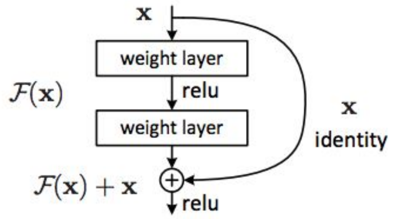


Figure 2. Residual learning: a building block.

Work on

Assignment 2B

Project Overview

Capstone Project: MURA

- **Goal:**
  - Create a classifier given **bone x-rays** to classify whether a x-ray is **normal** or **abnormal**
- Pick either PyTorch or TensorFlow to create a classifier from start to end.
- Create a model and optimize through hyper-parameter tuning.
- Include a small write-up about the medical implications of **machine learning** in **medicine**

Work on **Project**

Ethics

Ethics: Bias

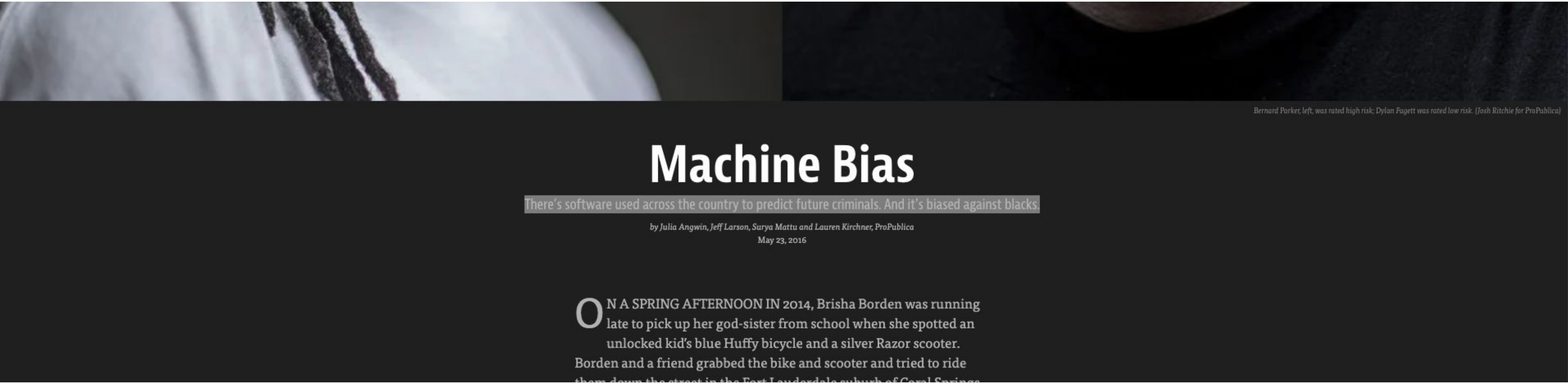
- Take the moral machine test
- How biased are you?
- Observe:
  - Bias is culturally dependent
  - Bias embedded in technical systems have far more reach than one person has ⇒ *Machine Bias*
- Let’s talk about Algorithmic Bias
  - Watch this Ted talk by MIT Graduate Student Joy Buolamwini who talks about how she has experienced machine bias in her life.
  - **Video:** https://www.youtube.com/watch?v=UG_X_7g63rY

Ethics: Why study bias?

- As observed, human bias is present within data and the frameworks being built.
- Algorithmic bias, as shown by Joy Buolamwini, can systematically disadvantage large groups of people disproportionately
- How do we solve it?
  - There is no one answer. It is extremely context-dependent.
  - Diversity in technology
  - Thorough assessment of equality of algorithms
  - Unbiased data collection

Case 1: Algorithmic Sentencing

- Read the following article:
  https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing


Bernard Parker, left, was rated high risk; Dylan Fugett was rated low risk. (Josh Ritchie for ProPublica)

Machine Bias

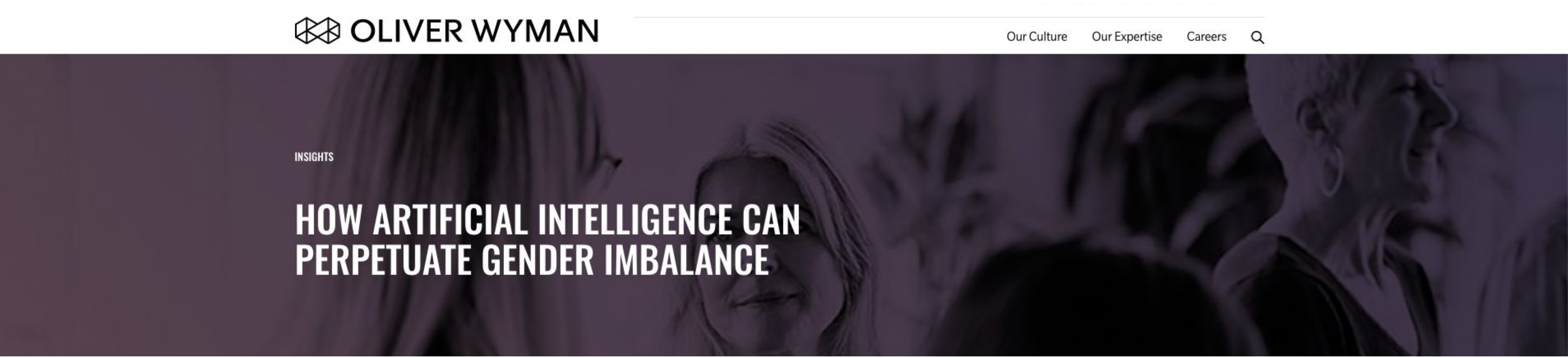
There's software used across the country to predict future criminals. And it's biased against blacks.

by Julia Angwin, Jeff Larson, Surya Mattu and Lauren Kirchner, ProPublica
May 23, 2016

ON A SPRING AFTERNOON IN 2014, Brisha Borden was running late to pick up her god-sister from school when she spotted an unlocked kid's blue Huffy bicycle and a silver Razor scooter. Borden and a friend grabbed the bike and scooter and tried to ride

Case 2: Gender Bias

- Read the following article:
  https://www.oliverwyman.com/our-expertise/insights/2020/mar/gender-bias-in-artificial-intelligence.html



Ethics

Capstone Project

- **After reading and learning about instances of ethics in A.I.**
    - **Think about ways of making your model more ethical**
    - **To get you started, some ideas:**
        - Meta analysis
            - See whether you see any patterns amongst misclassified x-rays.
            - Check whether there are any signs of overfitting.
        - Transparency
            - Visualize the learned filters. Do they make sense?
        - Create other metrics to capture the performance of your model
            - Think about ROCAUC/PRC

Take the

Quiz

End of Module 35 - Introduction to TF/Torch

Since you have completed the module successfully, you can:

1. get an intuition of the differences between the two libraries.
2. transition between programming, math and theory, specifically, the different optimizations and gradients.
3. load and augment data.
4. independently use any of the libraries to implement neural models of choice.
5. be able to effectively debug and read through documentation.
6. get some sense of residual nets are implemented on a basic level.