Introduction to TensorFlow and PyTorch

Kendall Chuang and David Clark

February 16, 2017

Acknowledgements

Thank you to Tubular Labs for hosting this workshop!

Who We Are

- David Clark
  - Data Science Consultant
  - http://www.dmclark5.com/

- Kendall Chuang
  - Senior Python Engineer at Ayasdi

Objectives

- Learn basic concepts of neural networks
- Work through tutorial on Tensor Flow
- Work through tutorial on PyTorch
- Compare/contrast each deep learning framework

What is Tensor Flow?

- Developed by the Google Brain Team and open-sourced in November 2015
    - Graph based
    - Nodes are operations
    - Edges are multi-dimensional arrays called tensors

- All operations are done outside of Python

- Inputs are stored in a **`placeholder()`** or **`Variable()`**
    - **`placeholder()`**: fixed input
    - **`Variable()`**: variable input

- Inputs are populated during a Tensor Flow session

What is a tensor?

- Vector: One-dimension
- Matrix: Two-dimensions
- Tensor: n-dimensions

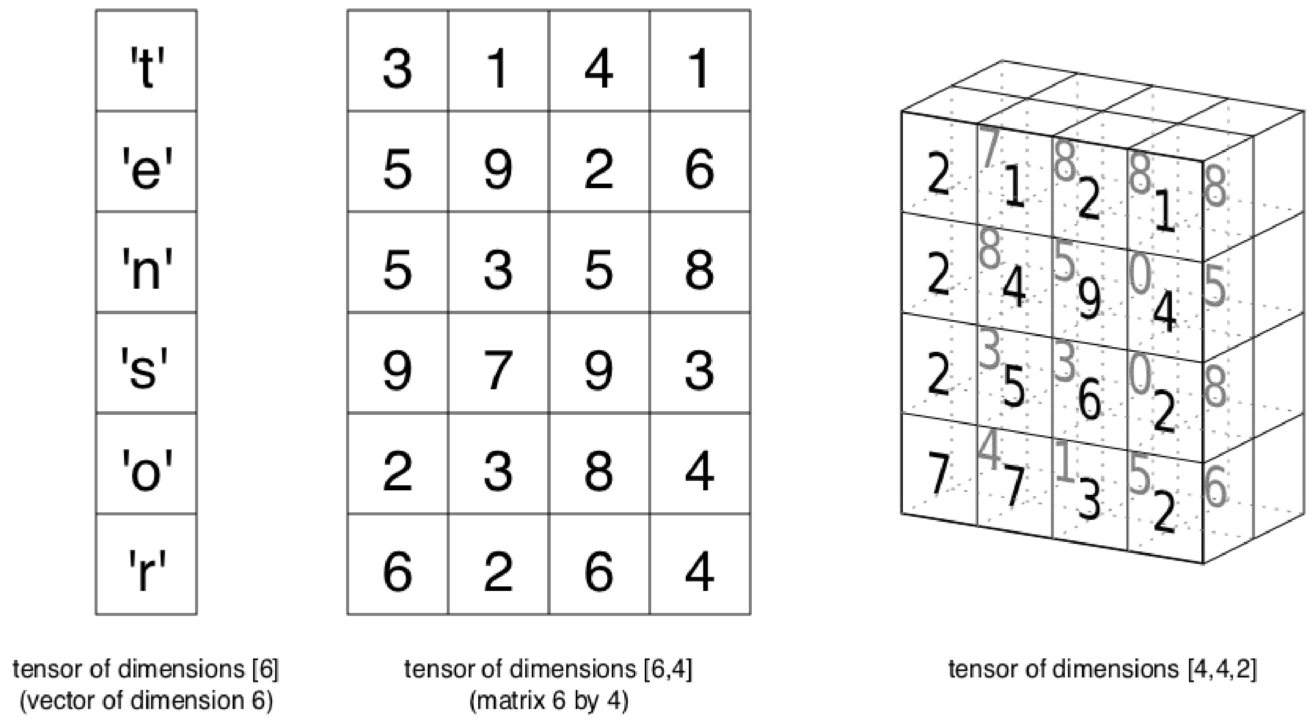


Image from http://noaxiom.org/tensor.

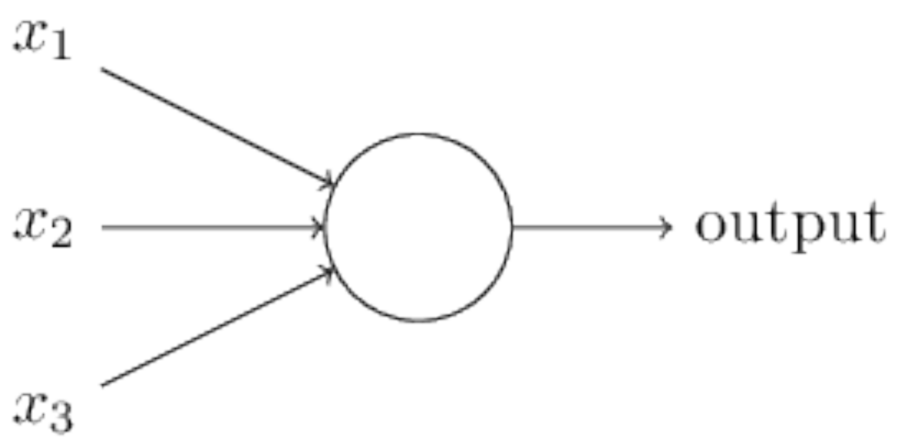
- Tensor Flow stores data in tensors

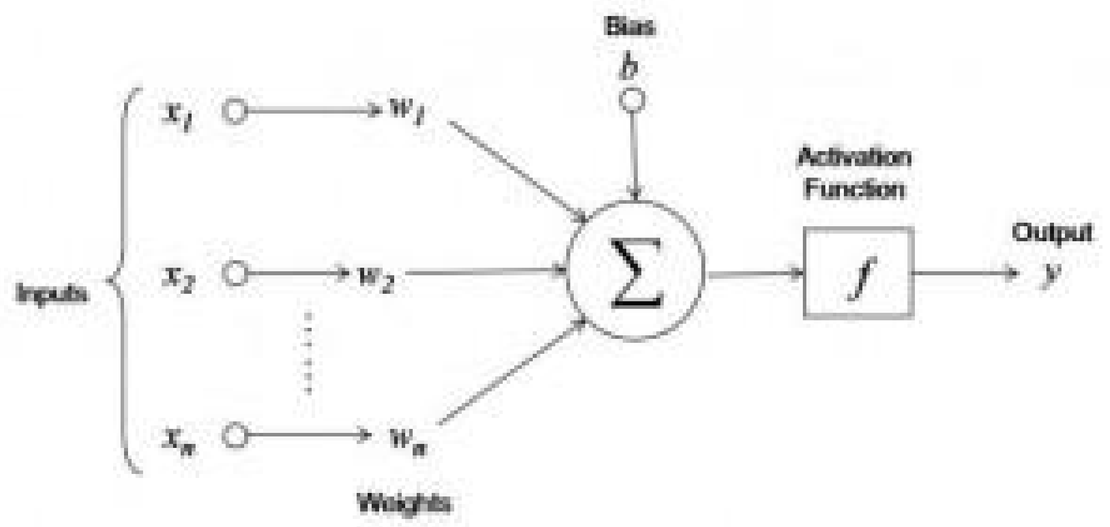
What is PyTorch?

- Python wrapper over Torch (C++ Library) released in Jan 2017
- Also stores data and variables as Tensors
- Developed by Facebook and other companies
  - Graph based
  - Nodes are operations
  - Edges are multi-dimensional arrays called tensors
- Deep Learning operations are done outside of Python
- **`Tensor`** and **`Variable`** objects
  - **`Tensor():`** inputs/outputs
  - **`Variable()`**: variable
- Does not have the concept of a session

What is a neural network?

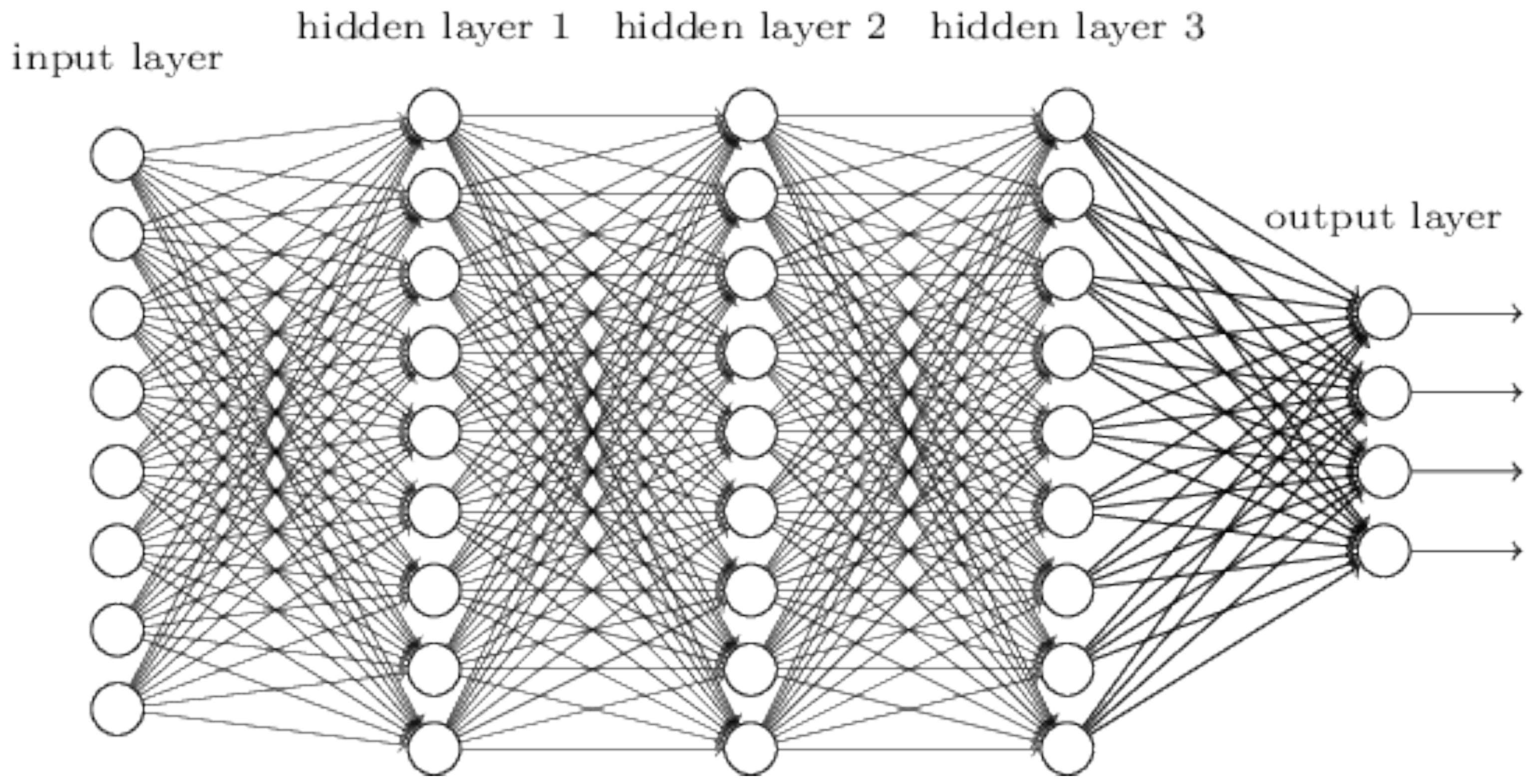
- Machine learning based on how the brain works
- Input is passed through neurons.
- Weights and biases applied.
- Decisions made using activation function.





Neural Networks and Deep Learning

What is a neural network?



- Many inputs can pass through a single neuron
- Hidden layers
  - Layer of neurons between inputs and outputs
  - Increased modeling complexity
- Inputs, hidden layers, outputs => neural network
- Deep learning
  - Many hidden layers
  - Can make very complex decisions

Neural Networks and Deep Learning

Softmax Regression

- Regression for j = 1,... K
- Algorithm for categorical data

- Evidence: Weighted sum of chemical properties.
  - Negative weight: is not of that quality.
  - Positive weight: is of that quality.

$$E_i = \sum_j W_{i,j} x_j + b_i$$

- Convert evidence to predicted probabilities using softmax equation

$$\sigma(E)_j = \frac{e^{E_j}}{\sum_{j=i}^{K}} \quad for\ j = 1, \ldots K$$

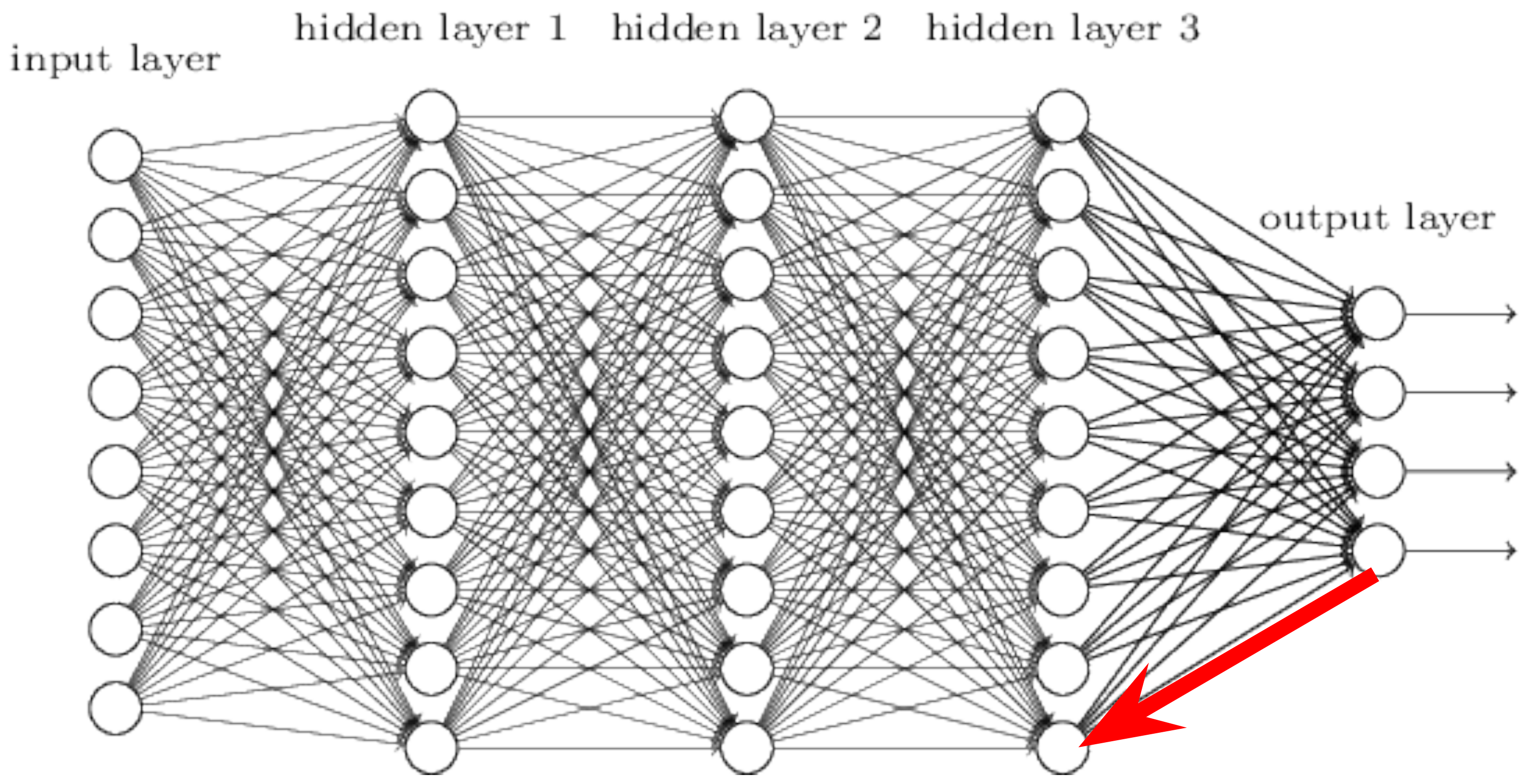
Cost/Loss Function

- Model trained by minimizing the cost function
- Cross-entropy cost function:

$$H_{y'}(y) = -\sum_i y'_i log(y_i)$$

- y = softmax(E)
- y’ = labels
- More Info: https://rdipietro.github.io/friendly-intro-to-cross-entropy-loss/

Back Propagation



- Backpropagation is used to evaluate how variables affect the minimization of the cost function

Training Loop

1. Forward Propagation (feed in variables forward, calculate the results)

2. Calculate Loss at the Output

3. Back Propagation (propagate loss backwards, estimate error)

4. Optimization with Gradient Descent Algorithm

5. Repeat until done!