Tensorflow Overview

CS294-112: Deep Reinforcement Learning
Goal of this course

Train an agent to perform useful tasks
Goal of this course

Data → Agent

Train model

Agent → Data

Get more data
Goal of this course

Data → Train model → Agent → Get more data

This lecture
Machine learning

How to do this?

\[ \theta^* = \arg \min_{\theta} \sum_{(x,y) \in D} \| f_\theta(x) - y \| \]

gradient descent  neural networks
What is Tensorflow?

- Library for
  - Defining computation graphs
  - Calculating gradients
Tensorflow: defining computation graphs

\[ h_1 = \sigma(W_1x) \]
\[ h_2 = \sigma(W_2h_1) \]
\[ y = \sigma(W_3h_2) \]
Tensorflow: calculating gradients

\[ h_1 = \sigma(W_1x) \]
\[ h_2 = \sigma(W_2h_1) \]
\[ y = \sigma(W_3h_2) \]

\[ \frac{\partial y}{\partial W_1} = \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial W_1} \]
\[ \frac{\partial y}{\partial W_2} = \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial W_2} \]
\[ \frac{\partial y}{\partial W_3} \]
Tensorflow: calculating gradients

You define

\[ h_1 = \sigma(W_1x) \]
\[ h_2 = \sigma(W_2h_1) \]
\[ y = \sigma(W_3h_2) \]

TF calculates

\[ \frac{\partial y}{\partial W_1} = \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial W_1} \]
\[ \frac{\partial y}{\partial W_2} = \frac{\partial y}{\partial h_2} \frac{\partial h_2}{\partial W_2} \]
\[ \frac{\partial y}{\partial W_3} \]
Alternatives to Tensorflow

PyTorch, Caffe, Theano, MXNet, Chainer, ..... 

fundamentally the same implementation differences
Hands-on with Tensorflow

<switch to Jupyter notebook>