Neural Networks and Google TensorFlow with R!

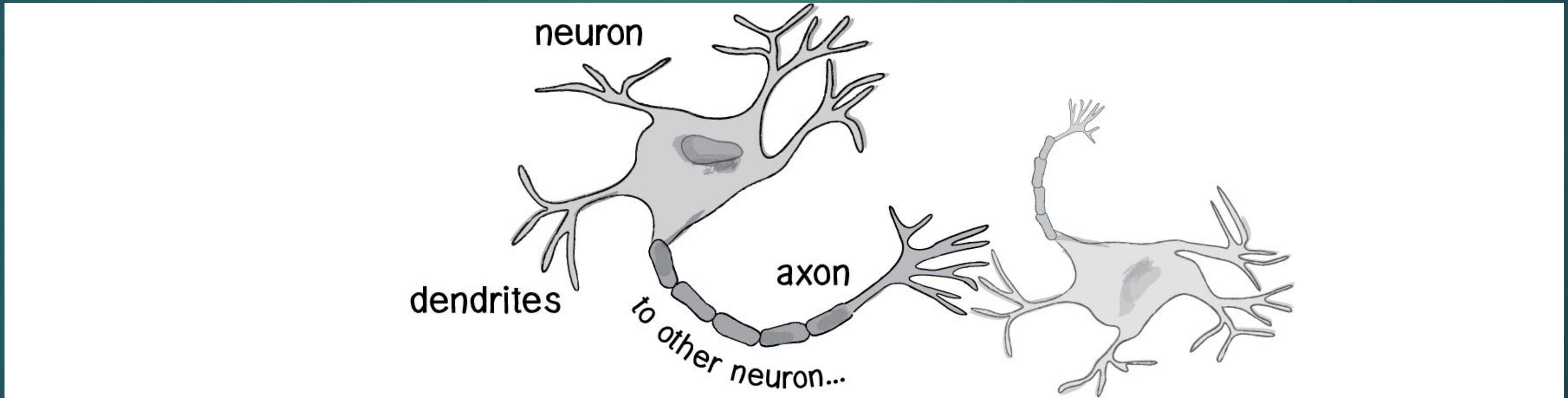
SHANNON MCCORMICK

Outline

- Intro to Neural Networks
  - Inspiration
  - Basic explanation
  - Practical applications
- Google TensorFlow
  - What is it?
  - R Package
  - Basic Usage
  - Examples

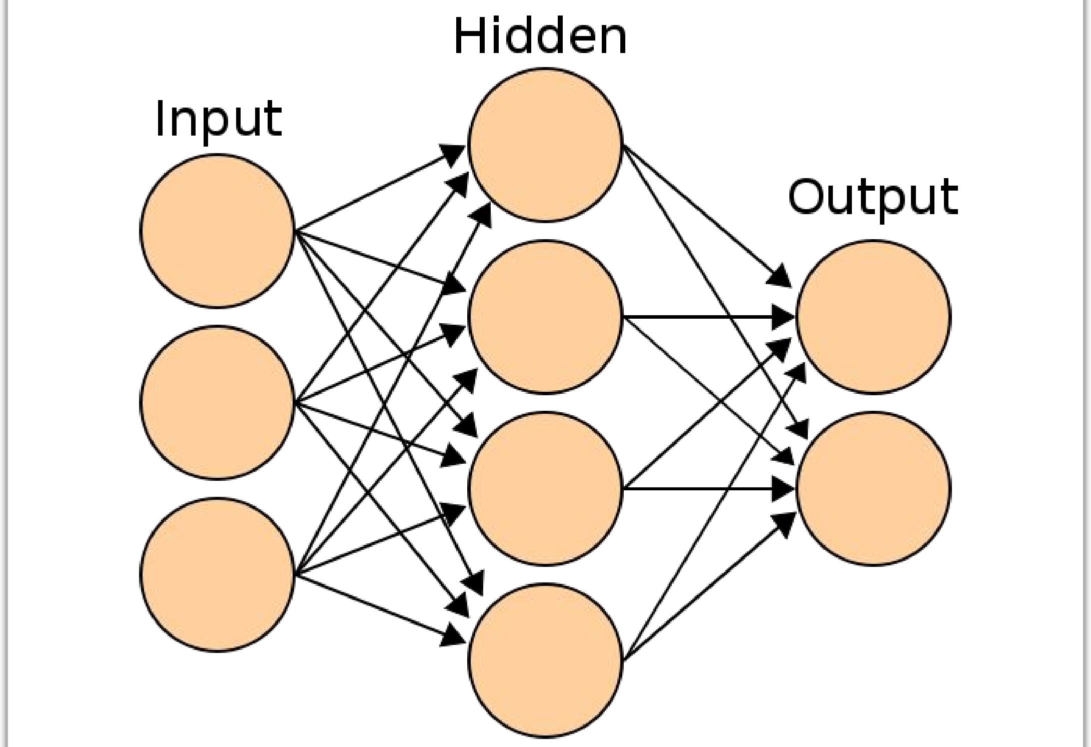
Neural Network Inspiration

- Real Neurons
  - Dendrites receive an input
  - Based on input, axons output something to next neuron

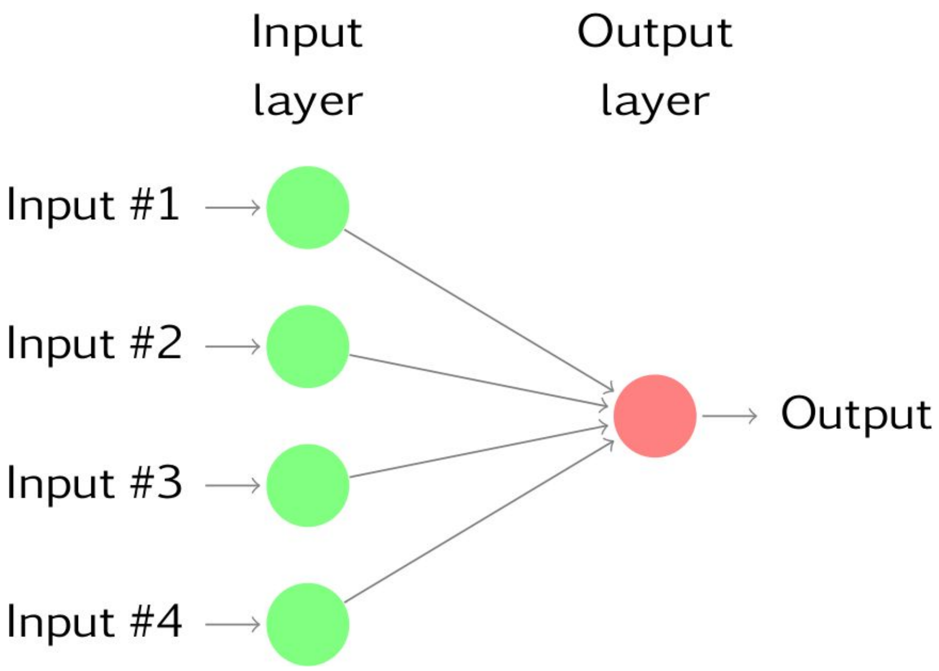


Artificial Neural Networks

- Set of nodes connected by directional lines representing weights
  - Nodes represent mathematical operations
  - Weights learned by training



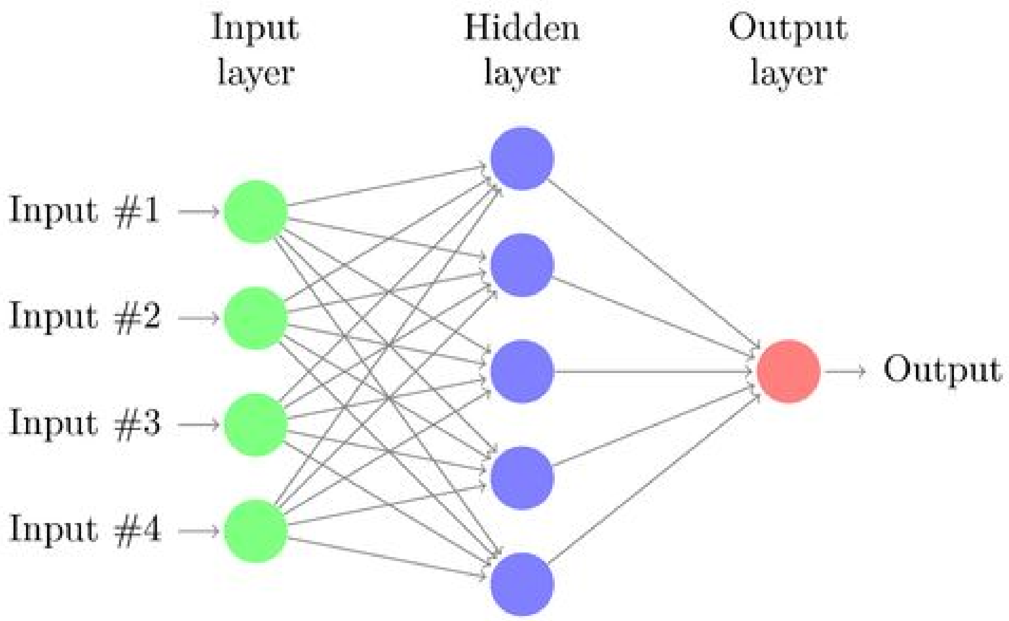
Linear Regression Example



- No hidden layers
- Inputs * Weights = Output
- Weights selected that minimize the error
- Great at modeling linear relationships

Adding Hidden Layers

- Add hidden layer(s)
- Input x Weights$_1$ = Hidden Layer
- Hidden Layer * Weights$_2$ = Output
- Weights selected to minimize error
- Can model more complex relationships

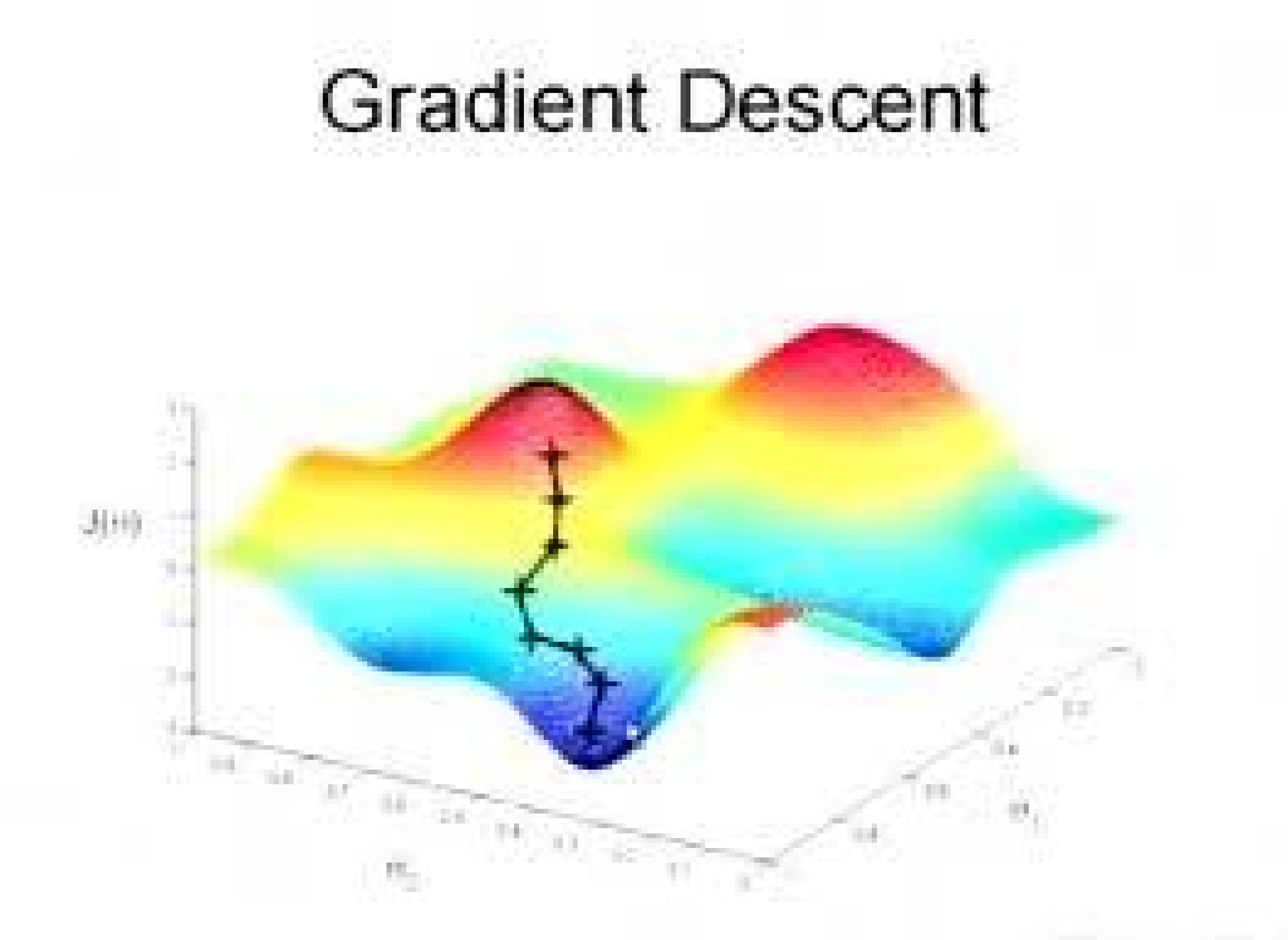


Learning

- Start
  - Weights have random initialization
  - Biases initialized at zero
- Forward Propagation
  - Batch of data goes through the network
  - Prediction/Classifications are output
- Error Calculation
  - Differences between true and outputted values are calculated

Learning weights

- Back Propagation
  - Errors are back propagated through the model
  - Determines the errors at each neuron in the network
- Gradient Descent
  - Optimization method
  - Determine how to change the weights
  - Takes a step down gradient of the function
- Iterative process



Gradient Descent

Google TensorFlow

- Open source machine learning library
- Released November 9, 2015
- Now the most popular machine learning framework

Features

- Can be used on desktop, mobile, servers
  - Linux, Mac OS, and Windows (added Nov, 2016)
    - GPU support on Linux and Mac OS (June, 2016)
- Written in C++ with Python interface
  - Added experimental Go (Nov, 2016) and Java (Feb, 2017) APIs
- Excellent step by step tutorials and documentation
- Auto-differentiation
- Includes Tensorboard for graph visualization
- Active improvement and growth
  - Google cloud (March 2016)
  - Distributed computing support (April 2016)
  - TensorFlow 1.0.0 released 5 days ago

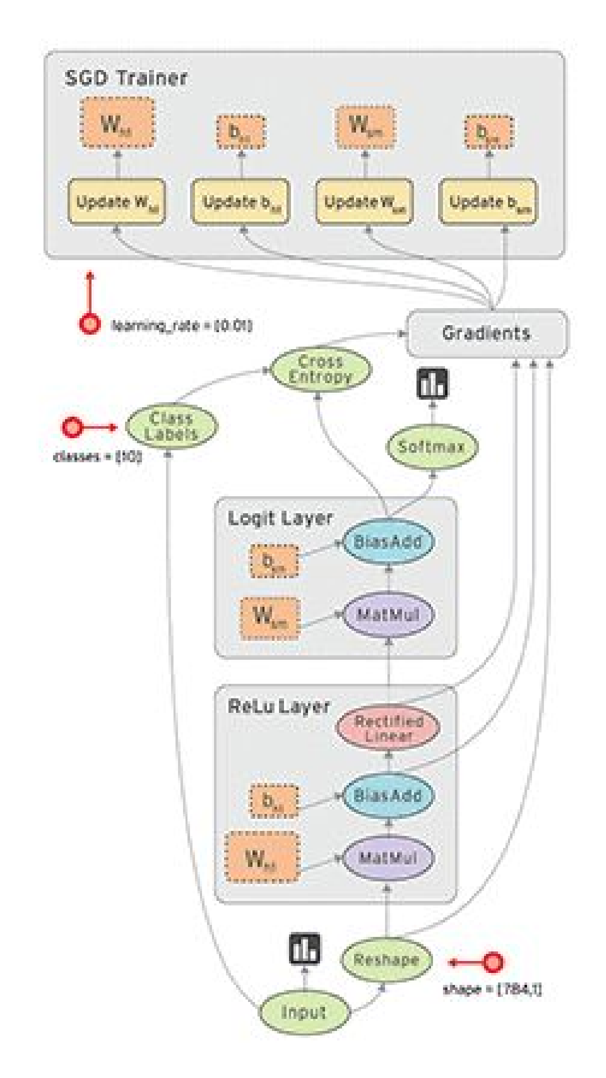
TensorFlow R Package

- Released October 2016
- Provides access to Python API through R
  - dependent on Python
  - ~~Does not work with Ananconda~~
  - ~~Not compatible with Windows~~
- When used with RStudio the package provides code completion and inline help
- Does include TFlearn higher level functions
- Cons
  - Not many examples
  - Points you to Python API documentation for

Basics

- Data flow graph with nodes and edges
  - Nodes: mathematical operations
  - Edges: input/output relationship between nodes
  - Edges carry tensors

Tensors flow through the graph

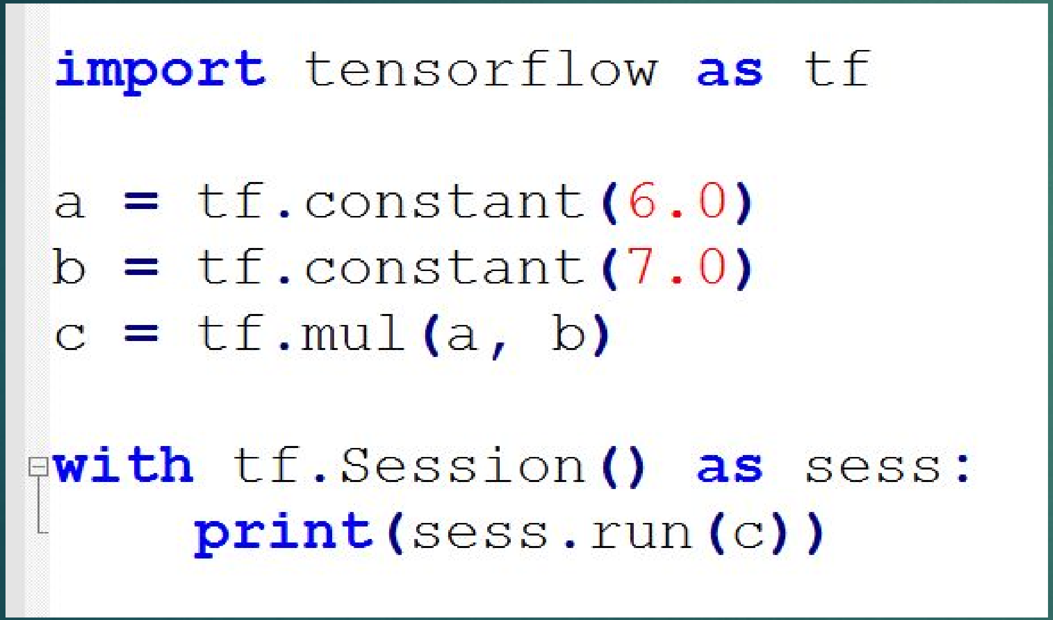


Building a model

"TensorFlow programs are usually structured into a construction phase, that assembles a graph, and an execution phase that uses a session to execute ops in the graph." - TensorFlow docs

- Define computation graph
  - Inputs, operations, outputs
  - Symbolic representation of model
- Run session
  - Execute graph
  - Fetch output

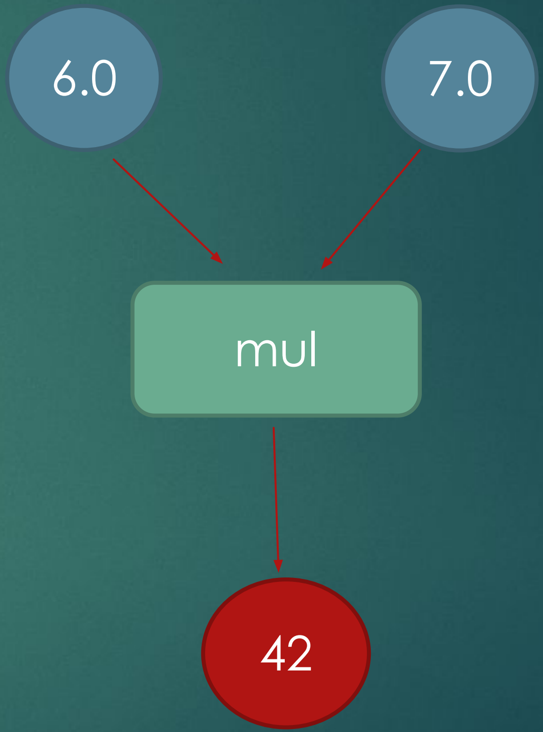
Simplest Example



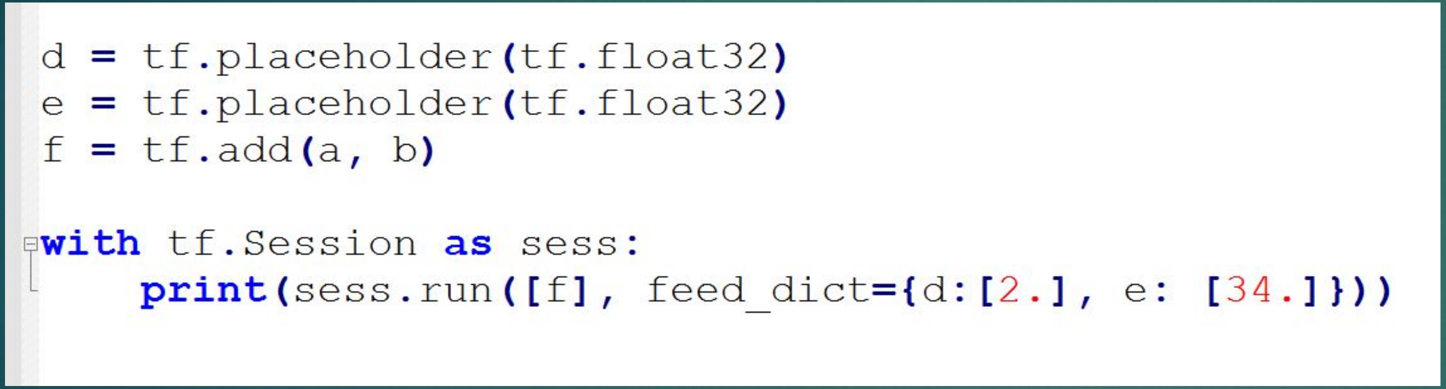
```
import tensorflow as tf

a = tf.constant(6.0)
b = tf.constant(7.0)
c = tf.mul(a, b)

with tf.Session() as sess:
    print(sess.run(c))
```

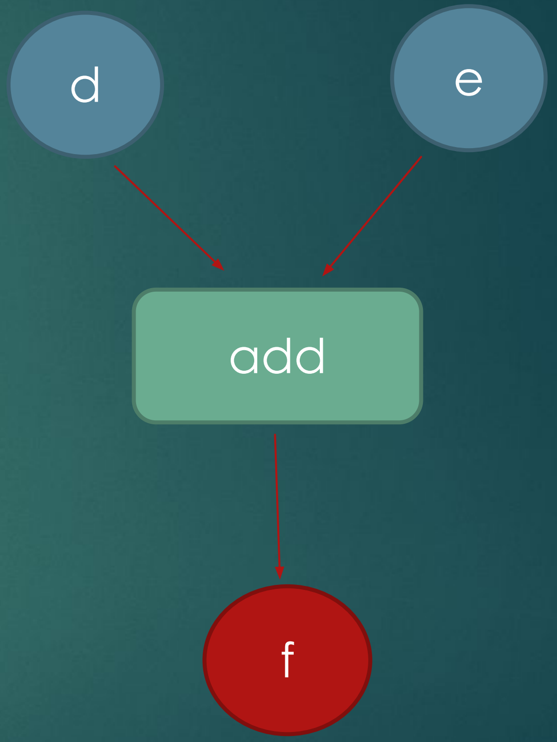


Simple Example



```
d = tf.placeholder(tf.float32)
e = tf.placeholder(tf.float32)
f = tf.add(a, b)

with tf.Session as sess:
    print(sess.run([f], feed_dict={d:[2.], e: [34.]}))
```



Classifying Covertypes

Covertype data set from UCI Machine Learning Repository

- Predict 7 cover types from 54 input variables
  - 10 quantitative and 44 binary
- Data used for 2000 paper "Comparative Accuracies of Artificial Neural Networks and Discriminant Analysis in Predicting Forest Cover Types from Cartographic Variables."
  - Trained neural network and attained 70.52% accuracy
  - Let's see if we can do better!

Some details

- 581,012 observation
  - Unbalanced categories
- Train, Test, and Validation
  - Train: 1620 observations of each covertype (11,340 obs)
    - 60% of observations of the least numerous cover type
  - Validation: 540 observations of each covertype (3,780 obs)
    - 20% of least numerous
  - Test: the rest (562,892 obs)
- Their network
  - 1 layer of 120 hidden nodes
  - Learning rate <- 0.05
  - Momentum rate <- 0.5

Additional Resources

- TensorFlow Tutorials
- Udacity Deep Learning Course
- Awesome TensorFlow
  - TensorFlow Examples
- WildML
- TF Learn (Scikit Flow)
- Keras
- Standford CS224d Lecture 7
- TensorBoard
- TensorFlow Playground

Questions