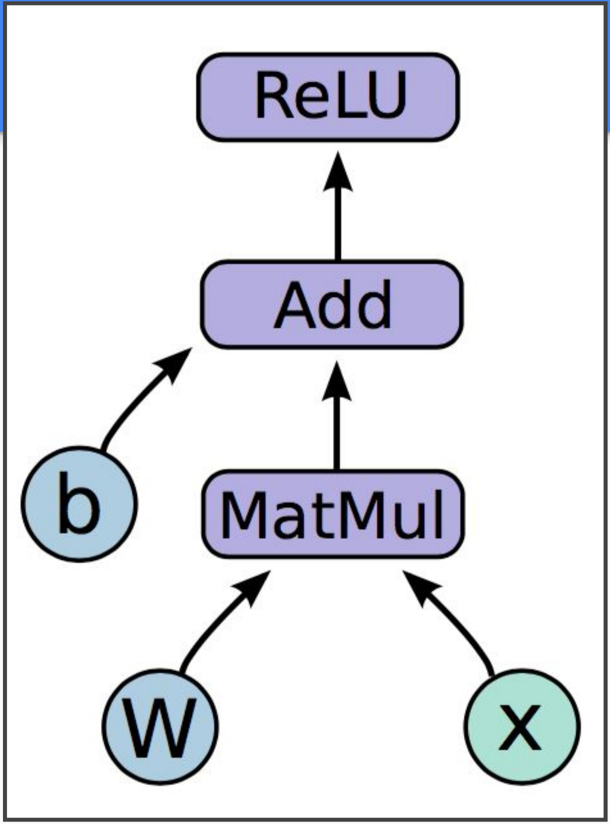
TensorFlow and R

Mochan Shrestha

What is TensorFlow

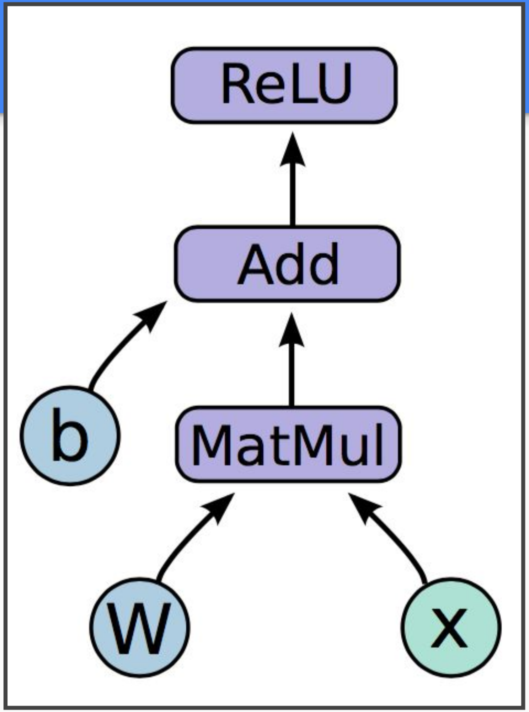
- Library that allows distributed numeric computation
- Computation is expressed as a graph
    - Graph nodes are operations
    - Graph edges are tensors (n-dimensional arrays, n-index matrices)

$$h_i = \mathrm{ReLU}(Wx + b)$$



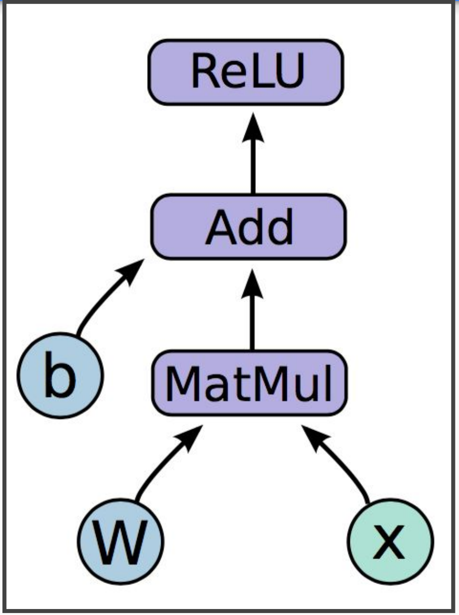
TensorFlow Model

- **Variables** are nodes with values (retained across multiple executions of a graph) - here W and b
- **Placeholders** are inputs - here X



In code (variables setup)

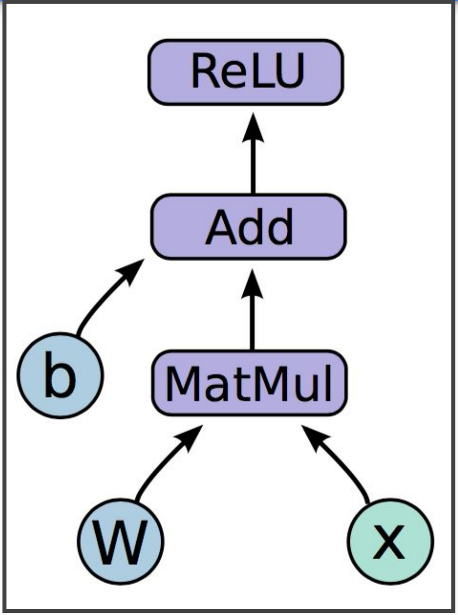
```
library(tensorflow)

x = tf$placeholder(tf$float32,
shape=shape(NULL, 7))

b = tf$Variable(tf$zeros(list(10)))

W = tf$Variable(tf$random_uniform
(list(7L,10L), -1, 1))
```



In code (computation graph)

```
h_i =
tf$nn$relu(tf$matmul(x,W)+b)
```

Basic block of a neuron computation



In code (execution)

```
sess = tf$Session()

sess$run(tf$global_variables_initializer())



xp = matrix(runif(5*7), 5, 7)
sess$run(h_i, dict(x = xp))
```

Installing TensorFlow for R

- Install the R-package devtools and then R-tools
- Get the package from github and build it using the command:

  ```
  devtools::install_github("rstudio/tensorflow")
  ```

- Setup the relevant python environment to run TensorFlow
  - Install Anaconda for Python
  - Use pip to get the TensorFlow package
  - If using GPU, install CUDA and then CuDNN

Hello World

- HelloWorld example (*HelloWorld.R*)
- Basic computation example (*Basic.R*)
- Computation Graph from before example (*CompGraph.R*)

TensorBoard (Visualization Tool)

- Visualize TensorFlow graph
  - Grouped by name-scopes
  - Expands to show each operation
- Plots: Statistical and 3d embeddings
- Images and Audio
- Graph of Example Computation Graph (*CompGraphTensorBoard.R*)

TensorFlow for Machine Learning

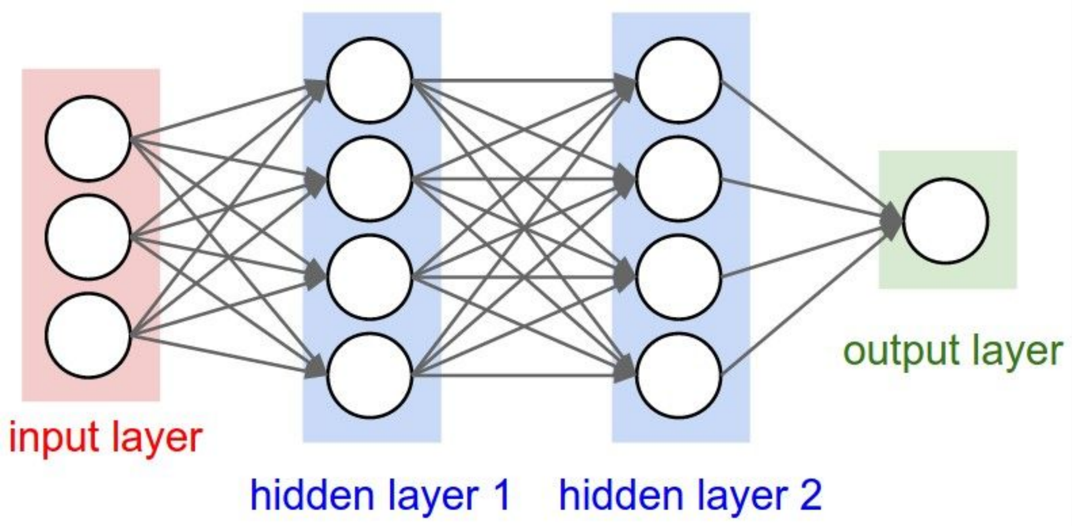
- Provides library of functions for machine learning and neural networks (deep learning) operations (on top of matrix operations)
- Computation graph leads very naturally to automated gradient computation - implemented function also has known derivatives and can optimize chain rule computations
- So optimization using stochastic gradient descent can be done on any computation graph

Example (Regression using Gradient Descent)

- Regression using gradient descent (*GradDesc.R*)
- TensorBoard visualization (*GradDescTensorBoard.R*)

Neural Networks

1. Directional Graph
2. Each layer is only connected to the one below it
3. Each edge is given a weight, each node a bias
4. Each input activates nodes based on the weights and biases

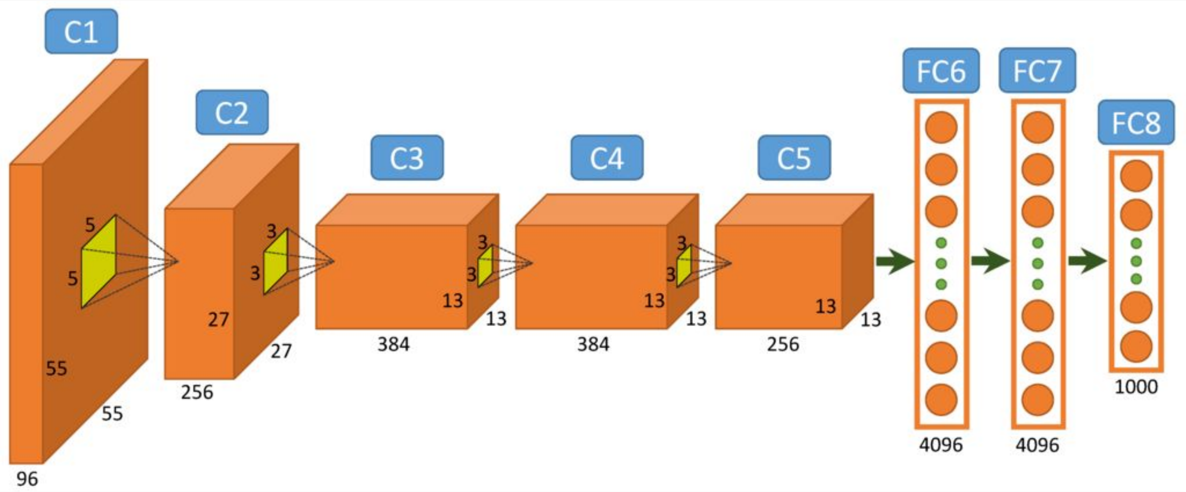


Using Fully Connected Neural Networks for IRIS dataset

- Neural Network for IRIS dataset (*iris.R*)
- TensorBoard for the above model (*irisTensorBoard.R*)

Convolutional Neural Networks

- Modeled after the visual cortex and effective for image and video machine learning
- More 2D local connections



MNIST Dataset

Dataset of hand-written digits

Sample MNIST Machine Learning

- Single Layer classified (*mnist_softmax.R*)
- Multi Layer fully connected (*fully_connected_feed.R*)
- Multi-layer convnet - *No Example :(*

  -
    ```
    conv2d <- function(x, W) {
      tf$nn$conv2d(x, W, strides=c(1L, 1L, 1L, 1L), padding='SAME')
    }
    ```

  -
    ```
    h_conv1 <- tf$nn$relu(conv2d(x_image, W_conv1) + b_conv1)
    ```

Questions?

Slides and code will be uploaded to github:

https://github.com/MochanShrestha

Will add ConvNet example to the repo.