

# Tensorflow Workshop

Shotaro Ikeda & Akshay Mishra

*<http://sigai.ml>*

September 22, 2017

# Overview

## 1 Basic Tensorflow

- Computational Graphs

## 2 Review: Supervised Learning

- Review: Problem Statement

## 3 Supervised Learning in Tensorflow

- Linear Regression/Classification
- Loss Functions

## 4 LeNet

## 5 Extra Resources

# Basic Tensorflow

# Remarks about Tensorflow

- Quite overwhelming, especially for those new to the library.
- There is multiple ways of doing the same thing.
- Very different way of thinking vs. other libraries.

# Nice Things about Tensorflow

- Very fast computation on GPUs (nearly 80x faster than CPU).
- Great distributed support for **large** clusters ( $\geq 8$  GPUs).
- Many available implementations.

# Tensorflow “Quirks” /Vocabulary

- Define and Run, instead of immediate feedback.
- Placeholders to transfer data into the computational graph.
- Variables for values/weights we want to learn.
- Ops to operate on individual Variables/Values
- Session to allocate the computational graph, and assign values

# Computational Graphs

- Let's say we want to compute some values, given variables  $x$ ,  $y$ , and  $z$ .
- $x \in \mathcal{R}^{2 \times 3}$
- $y \in \mathcal{R}^{3 \times 1}$
- $z \in \mathcal{R}^{2 \times 1}$
- Let's try to compute  $\text{MEAN}(xy + z)$

# Demo – Calculations in Tensorflow

basics.ipynb

## Example – Conclusions

- In order to run any computation in Tensorflow, you need a Session.
  - You need to define the computational graph first
  - Then initialize your variables and session.
- Tensorflow batching is very easy to do, and fast on GPUs

# Review: Supervised Learning

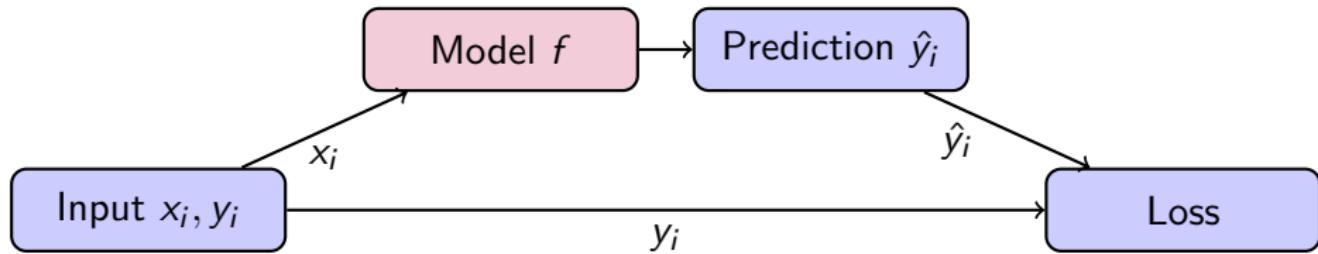
# Supervised Learning - Problem Statement

- We have some dataset  $d \sim \mathcal{D}$ .
- $d$  contains a bunch of points,  $x_i$  and  $y_i$ , where  $x_i$  is the  $i$ th datapoint and  $y_i$  is the  $i$ th label
- You can think of  $x_i$  as an image, and  $y_i$  being the corresponding label.
- Given enough  $x_i$ 's and  $y_i$ 's, is it possible to accurately predict  $\hat{y}$  for some foreign, unseen inputs  $\hat{x}$ ?

# Supervised Learning - Problem Statement

- Idea: suppose there exists some function  $f$ , such that  $f : a \rightarrow b$ , essentially mapping our inputs  $x$  to output  $y$ .
- For the workshop today, this  $f$  is approximated by Tensorflow.

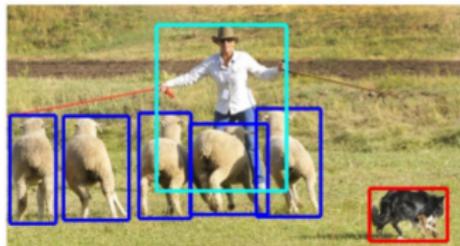
# Typical Deep Network Pipeline



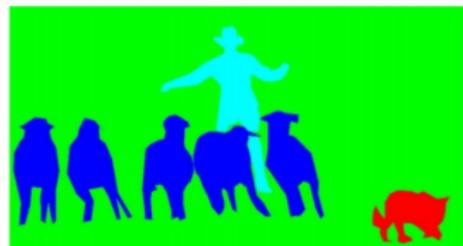
# Examples of Supervised Learning



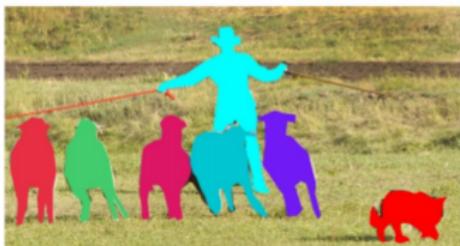
(a) Image classification



(b) Object localization

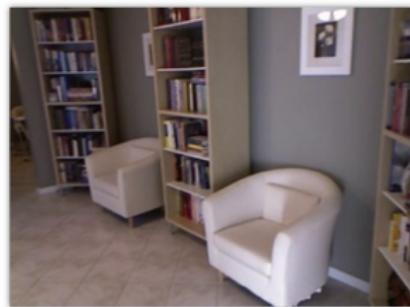


(c) Semantic segmentation

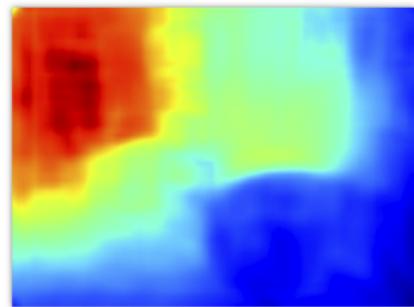


(d) This work

# Examples of Supervised Learning



Single RGB Image



Depth Map

# Linear Regression/Classification

## Linear Models

If we have a series of points  $x$  and  $y$ , then we want to fit the function

$$y = \alpha x + \beta$$

where we learn  $\alpha$  and  $\beta$ , such that it is “close” to our desired point,  $\hat{y}$ .  
Intuition: Fitting 2D points to a line.

# Tensorflow & Linear Models

You can think of a 1 layer neural network with biases, as a special case of a linear model!

# MSE Loss

## MSE Loss Function

We want to optimize the average distance between our predictions, so we want

$$\text{minimize } \frac{1}{n} \sum_{i=0}^n (f(x_i) - y_i)^2$$

# Cross Entropy Loss Function

## Cross Entropy Loss Function

We want to optimize the probability outputs, so we want

$$\text{minimize} - \sum_{c=0}^C (\hat{y}_{ic} \log(y_{ic}))$$

for individual datapoints.

$C$  is the total number of classes, and  $y_{ic}$  gives the  $i$ th datapoint and probability of the  $c$ th class.

# Demo – Linear Regression and Classification

linear.ipynb

# Linear Regression and Classification – Conclusion

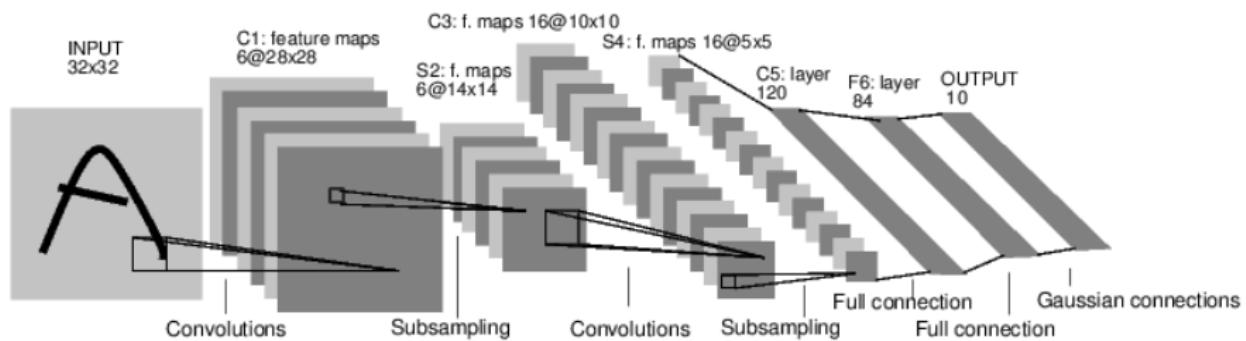
- Placeholders can be used to dynamically set the values of tensors at runtime.
- In Tensorflow, we use Variables to define what we want to learn.
- Optimizers can automatically differentiate Variables.

# LeNet

# Demo – Convolutions

google

# LeNet Architecture

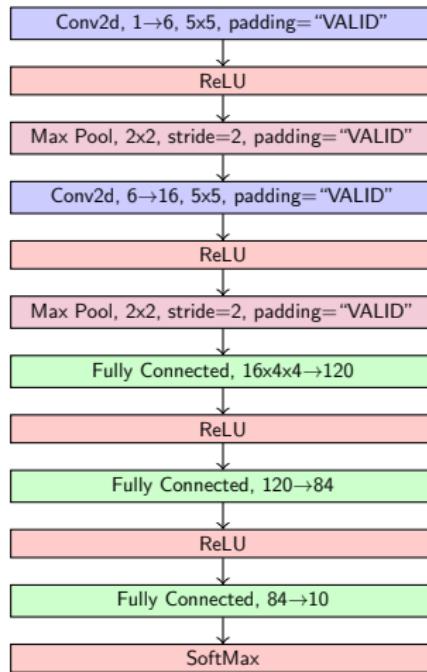


## Note!

Note – slight bug in diagram, the image size is 28x28, not 32x32.

The next slide shows the LeNet architecture that we will implement.

# LeNet Architecture (in more detail)



## Activity – Convolutions

mnist.ipynb

# Extra Resources

- CS231n – Recommended for beginners
- Extra Tutorial – If you want more practice using Tensorflow
- What is a Convolution?
- What is Pooling?