Class 10: Artificial Intelligence

BUS 696

Prof. Jonathan Hersh

Class 10: Announcements

1. Midterms graded
   - Average 88%.
2. ~~Pset 4 due Nov 18.~~
   - Will be optional (for extra credit) and due Dec 2nd
   - Pleased with midterm grades and would rather you focus on final project
3. Final Project One Sheet due Nov 18th
4. Today: Interview Ana Rocca, PhD
5. Next week: Please complete Udemy SQL Course chapman.udemy.com

Today: Ana Rocca, PhD
Formerly: Head of Central and US&C Research and Insights, Uber

- PhD, Economics UC Berkeley
- AB Economics, University of Chicago

- Will discuss case study "Applying Machine Learning to Improve the Customer Pickup Experience."

Case Study Today

Northwestern
Kellogg
School of Management

KE1161
January 14, 2020

Copy or Post

MOHANBIR SAWHNEY, BIRJU SHAH, RYAN YU, EVGENY RUBTSOV, AND PALLAVI GOODMAN

Uber:
Applying Machine Learning to Improve the Customer Pickup Experience

In 2018, Birju Shah, group product manager of maps and sensors, Ryan Yu, senior product manager of pickup experience, and Evgeny Rubtsov, product analyst of maps at Uber Technologies, were working on the best way to measure and improve the quality of the pickup experience for riders and drivers. Ensuring that the pickup experience went flawlessly was a top priority at Uber. Flawed pickups could lead to rider and driver dissatisfaction, reduce driver productivity, and increase the frequency of canceled rides.

Picking up a rider sounded like a simple task. Pulling off a flawless pickup experience was very challenging in practice, however. Finding the precise location of the rider and navigating the driver to the best rendezvous location in an efficient manner was not easy, especially in crowded locations like airports and concert venues. GPS navigation signals could be flawed in urban areas with tall buildings. One-way streets and parking restrictions could also create problems for drivers. Further, drivers and riders across the world had different definitions of what constituted a good experience.

- Chapman has purchased this case study for you!
- Please use the following link to download the case study and read for next week to prepare for interview

https://www.thecasecentre.org/educators/courses?id=1356705&pdid=171935&opid=855816

Class 10: Outline

1. Interview with Ana Rocca / Discussion of Uber Case Study
2. Break
3. Deep Learning – Feedforward Models
4. Deep Learning – CNN/Computer Vision Models
5. Deep Learning – Lab (time permitting)

November 18th – Due: students must upload to Canvas a one-page outline of their project

a) identify a dataset you will use
b) the outcome you are trying to predict, and what variables you will use to predict it
c) motivation to your project -- as in the business or practical management use case of such a prediction
d) the names of the students who will be part of your group (up to three per group)

How to Find Dataset and Join Groups?

- Think about the topic/industry you want to cover in your final project
    - Finance?
    - Sports Analytics?
    - Entertainment?
    - Marketing Analytics?
    - Personnel Analytics?
    - Other?

Breakout Rooms!

- Spend 5-10 minutes in a breakout rooms
- (Upgrade to latest zoom and can self-join different rooms based on topics)
- Find a group of up to three students
- Brainstorm ideas for datasets (see links at last slide)

“Reading” for Next Week: Udemy SQL Course

BUS 696

Course recommendations for learning SQL and

Editor

JH Jonathan Hersh

Overview

4hr 4min • 1 item | Public

1 **SQL - Please View by November 18th**

1 item | 4hr 4min

MySQL for Data Analysis - SQL Database for Beginners

Learn SQL for Business Intelligence & Big Data Analytics w/ MySQL Workbench (apply to SQL Server, Oracle, PostgreSQL)

99 of 107 items • 4hr 4min

2 **Tableau - Please View by December 2nd**

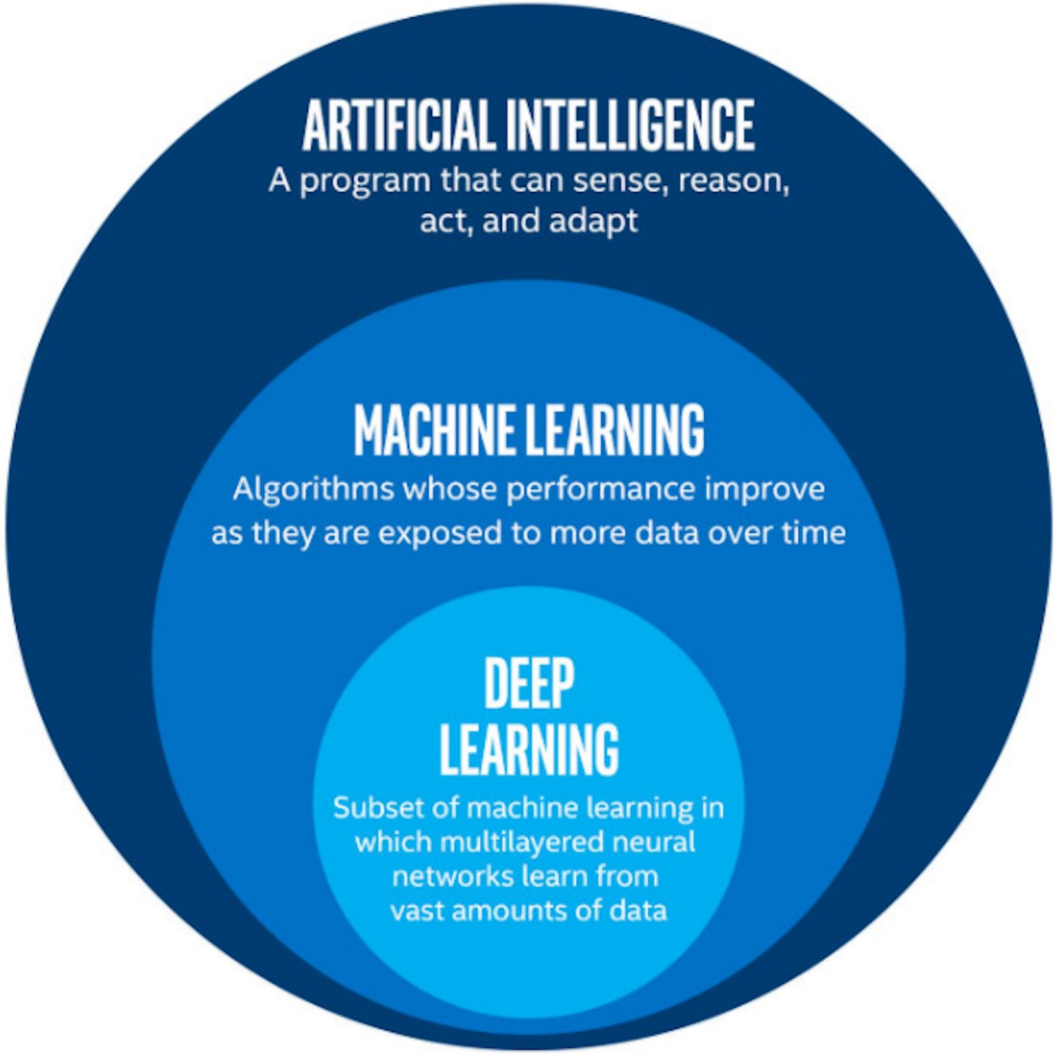
0 items | 0min

- You should have all received an invitation to chapman.udemy.com
- One you register, you will have access to all the courses on Udemy
- **For next week, I’d like to to watch the videos for the MySQL for Data Analysis course** – it’s the best resource I’ve found for learning SQL.

Class 10: Outline

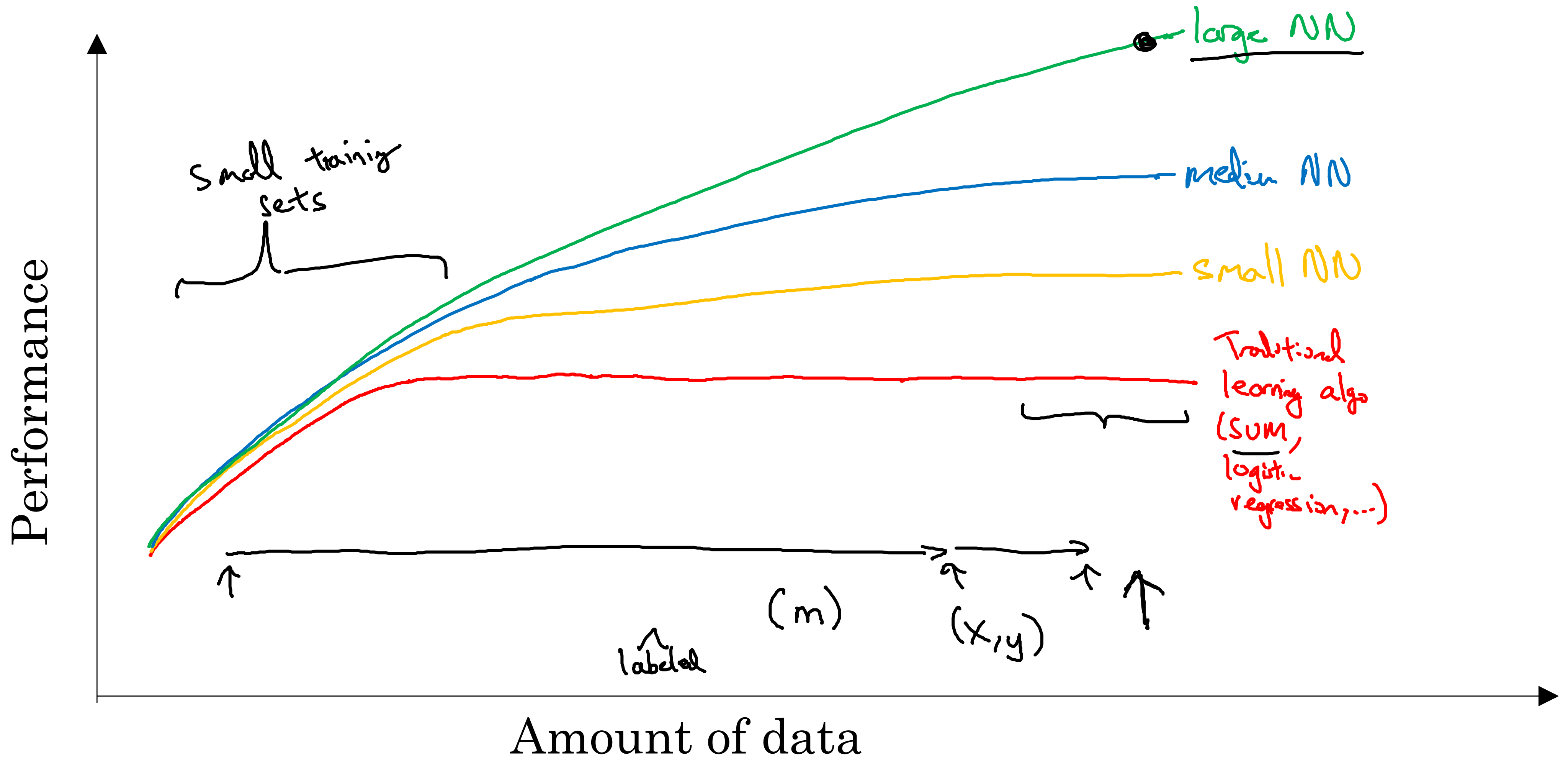
1. Interview with Ana Rocca / Discussion of Uber Case Study
2. Break
3. Deep Learning – Feedforward Models
4. Deep Learning – CNN/Computer Vision Models
5. Deep Learning – Lab (time permitting)

AI vs ML vs Deep Learning

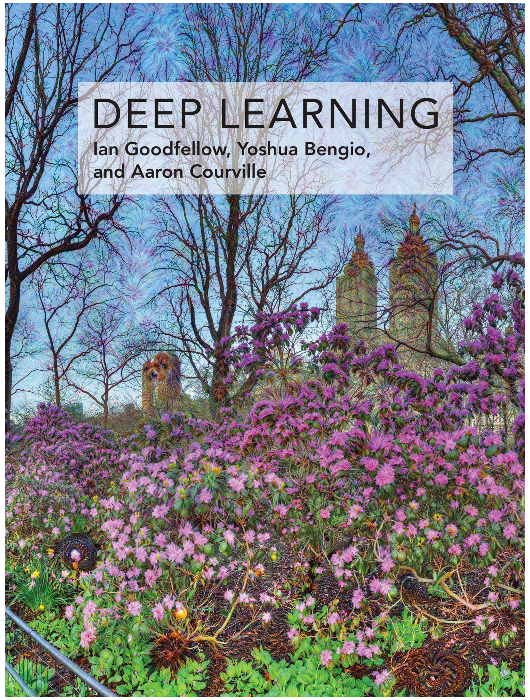


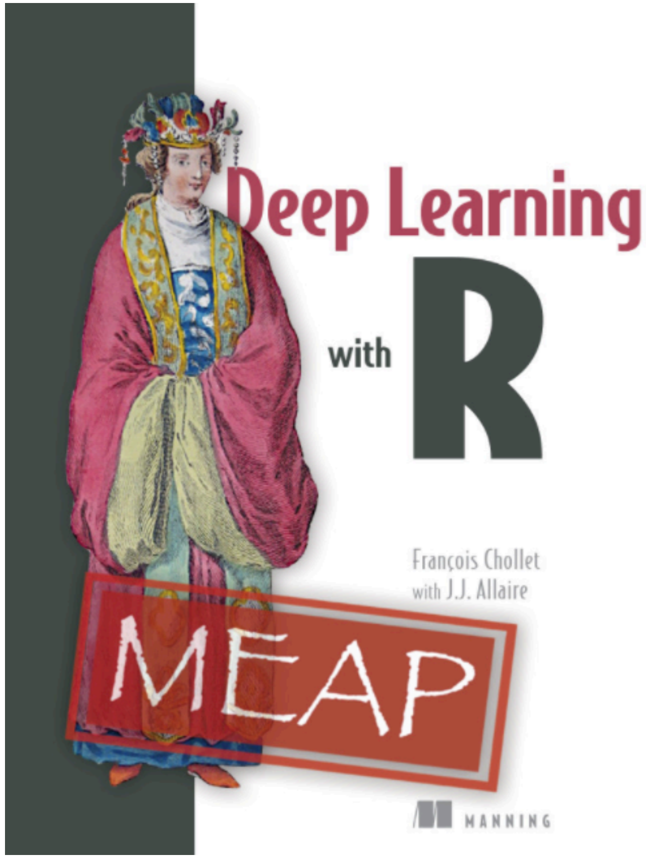
"**Deep learning is a specific subfield of machine learning:** a new take on learning representations from data that puts an emphasis on learning successive layers of increasingly meaningful representations. **The deep in deep learning isn't a reference to any kind of deeper understanding achieved by the approach**; rather, it stands for this idea of successive layers of representations. How many layers contribute to a model of the data is called the depth of the model" – Francois Chollet

~~Why~~ Deep Learning?



Sources



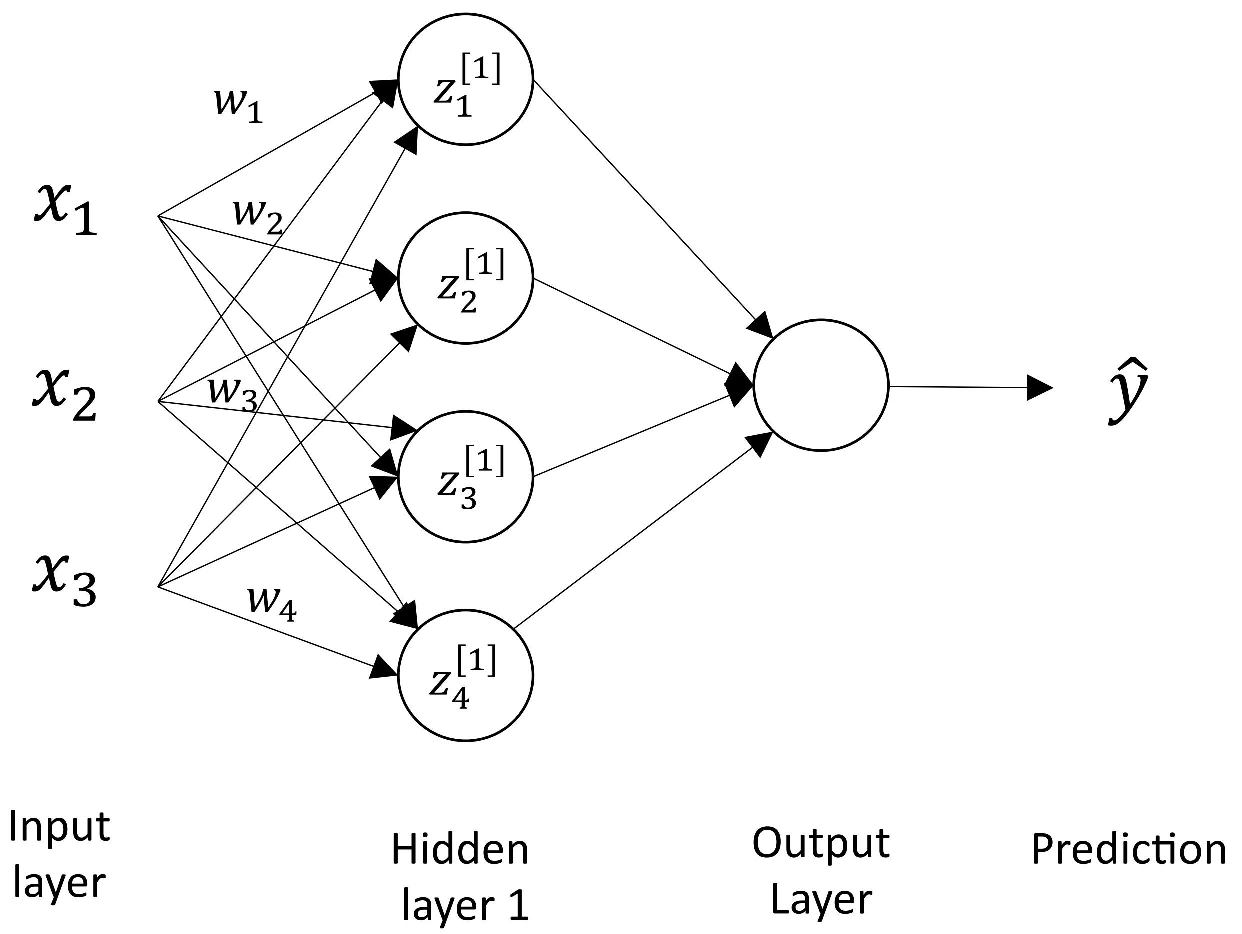


deeplearning.ai

Class 10: Outline

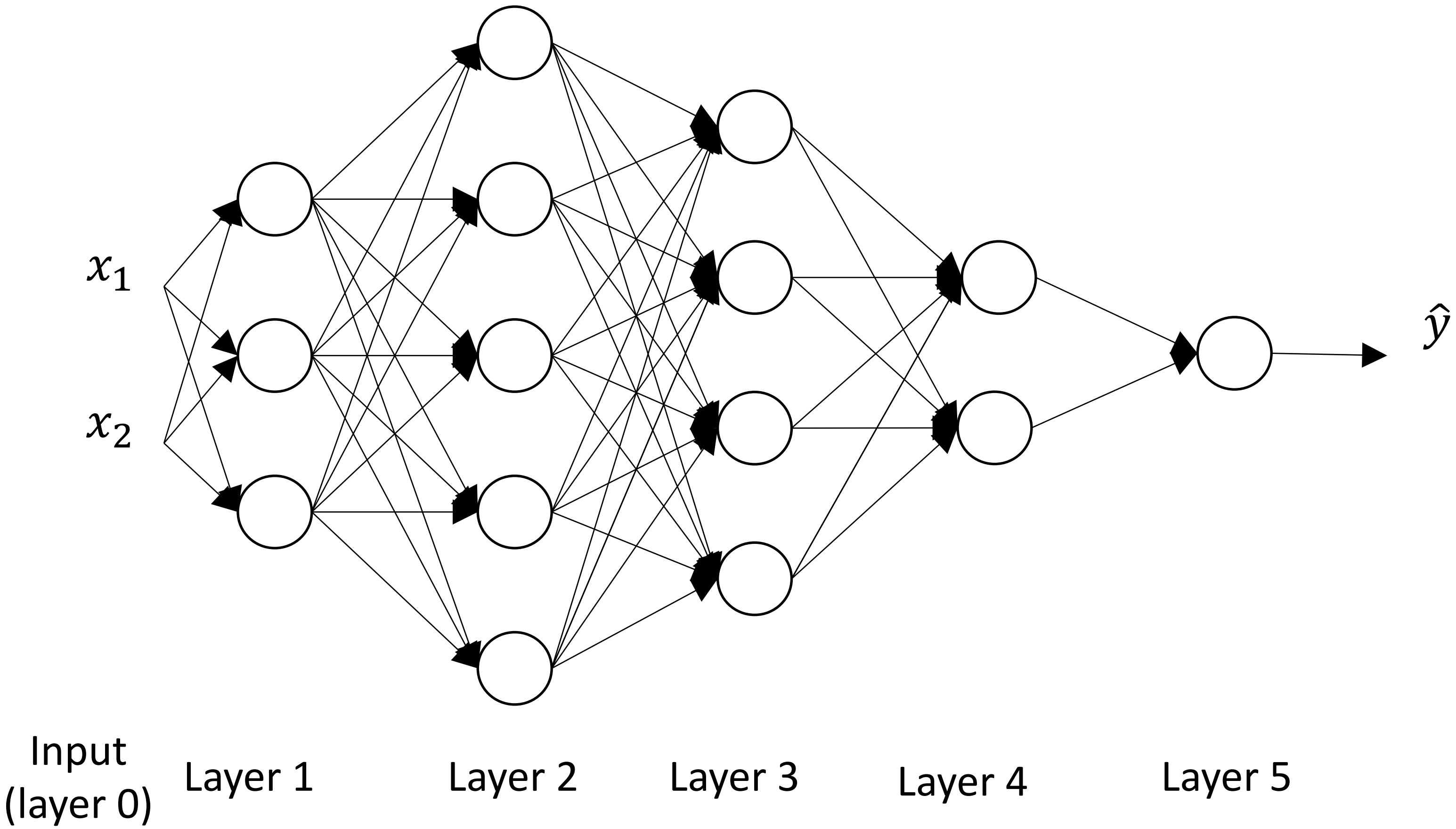
1. Interview with Ana Rocca / Discussion of Uber Case Study
2. Break
3. Deep Learning – Feedforward Models
4. Deep Learning – CNN/Computer Vision Models
5. Deep Learning – Lab (time permitting)

What is a Neural Network



- A neural network is a model that multiples several input layers (x1, x2, x3) with weight vectors (w1,w2,w3,w4) to produce hidden layers (z1,z2,z3,z4).
- These hidden layers are combined to produce an output layer or prediction
- Neural Nets are extremely flexible and can approximate any function (Universal Approximation Theorem)

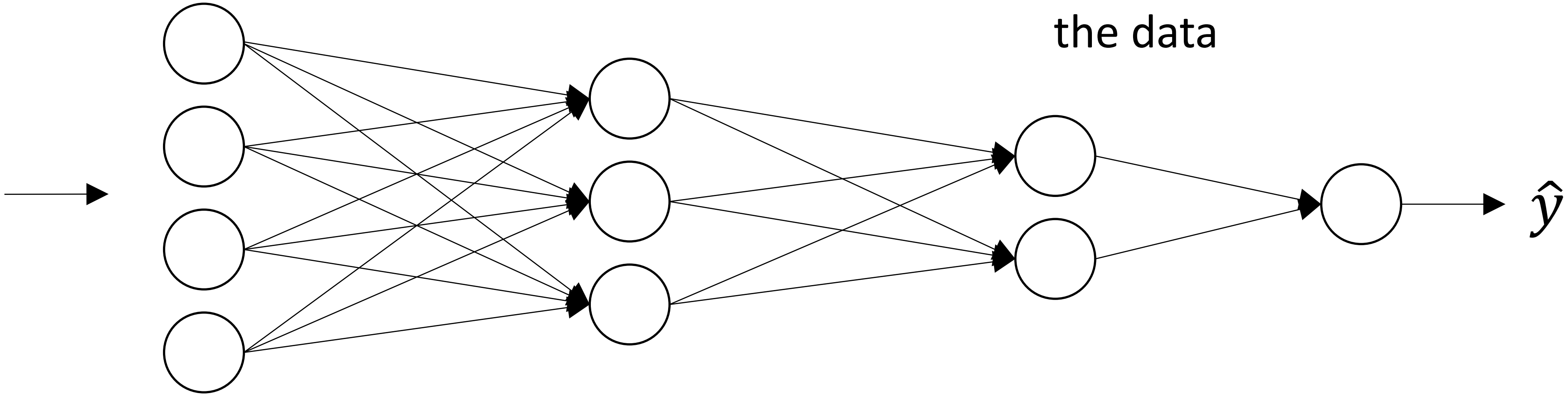
“Deep” neural networks



- A deep neural network is just a neural network that has more than one hidden layer
- The choice and number of hidden layers is called the **network architecture** and must be specified by the designer of the network

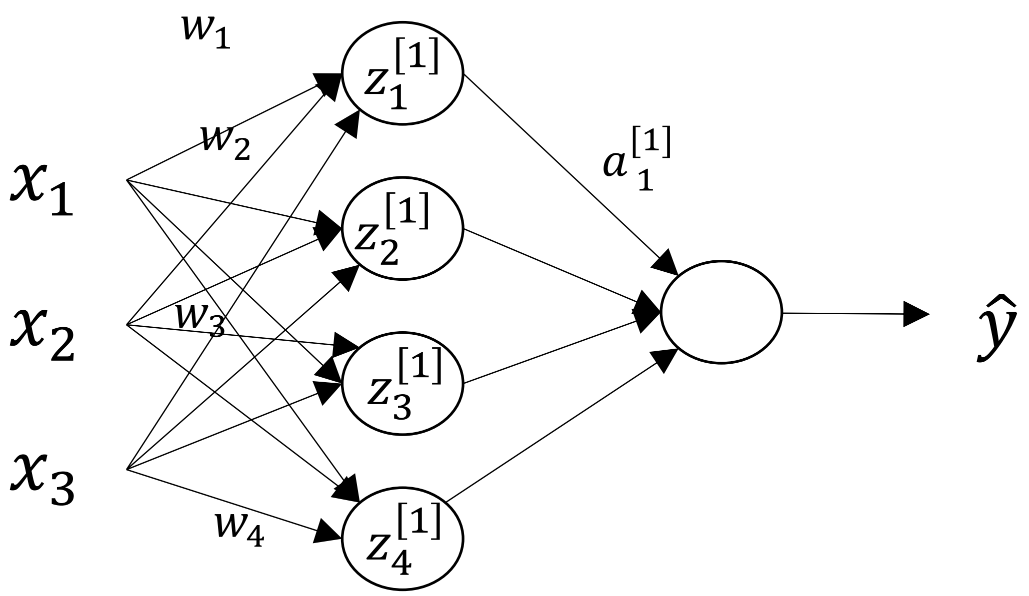
“Deep” neural networks

- Deeper hidden layers of the network learn more complex representations of the data



Deeper layers -> more complex representations

How Hidden Layers Combine to Form Output



$$z_1^{[1]} = w_1^{[1]T} x + b_1^{[1]},\ a_1^{[1]} = \sigma(z_1^{[1]})$$

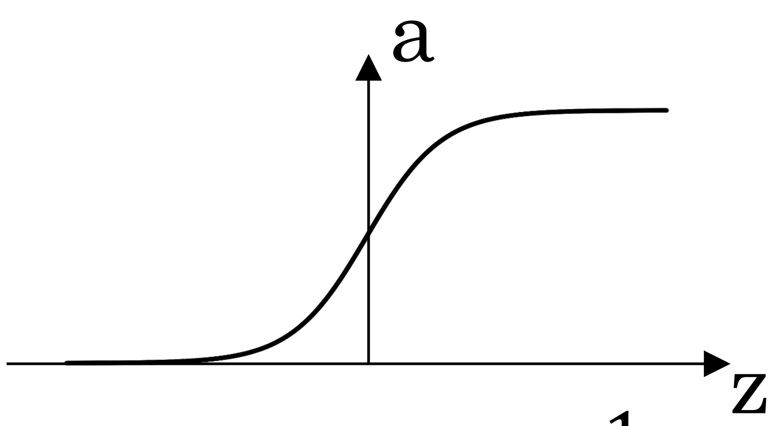
$$z_2^{[1]} = w_2^{[1]T} x + b_2^{[1]},\ a_2^{[1]} = \sigma(z_2^{[1]})$$

$$z_3^{[1]} = w_3^{[1]T} x + b_3^{[1]},\ a_3^{[1]} = \sigma(z_3^{[1]})$$

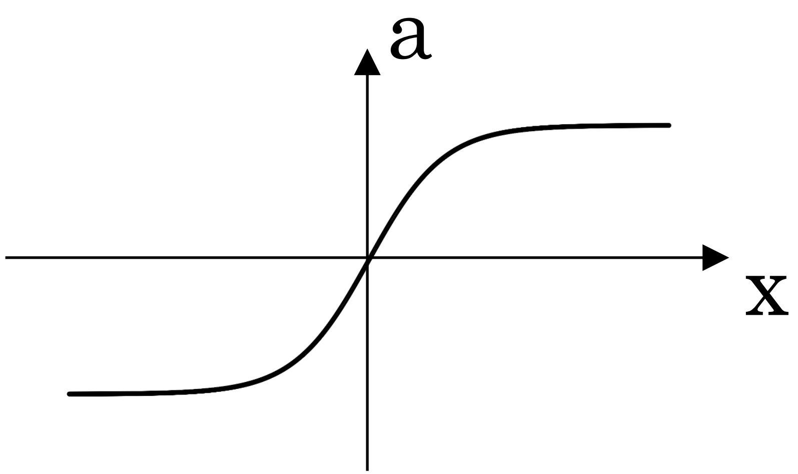
$$z_4^{[1]} = w_4^{[1]T} x + b_4^{[1]},\ a_4^{[1]} = \sigma(z_4^{[1]})$$

- Inputs are combined with simple algebraic to produce hidden layers (z) that are transformed by an **activation function** $\sigma(\mathrm{z})$ to produce the final predictions $\hat{y}$

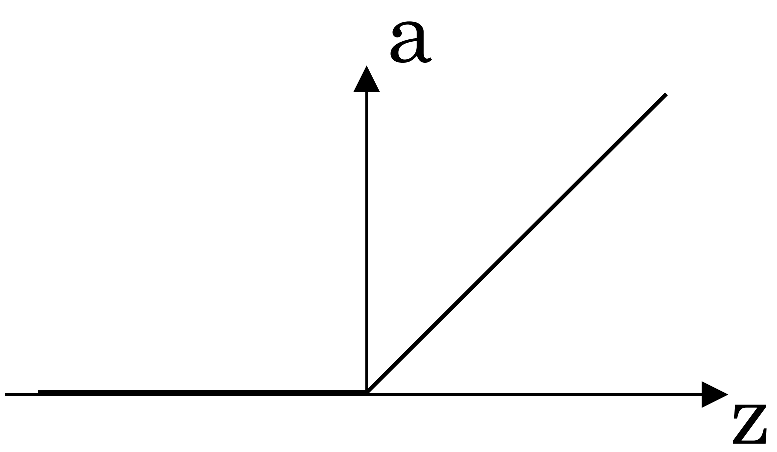
Output to Prediction: Activation Functions



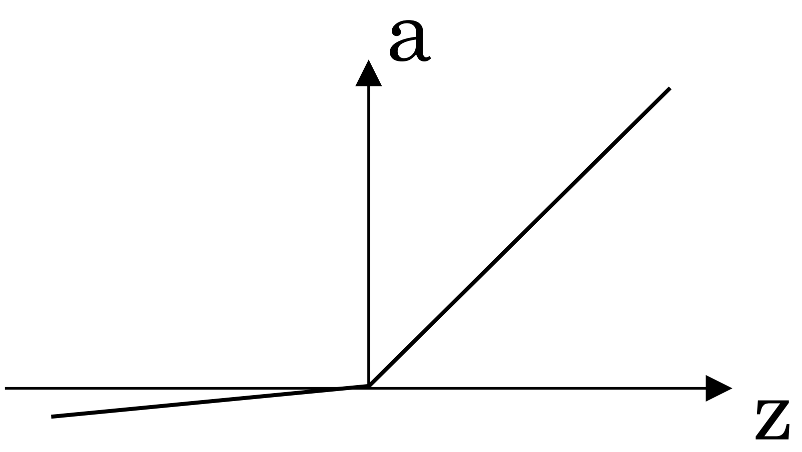
sigmoid: $a = \dfrac{1}{1 + e^{-z}}$



tanh



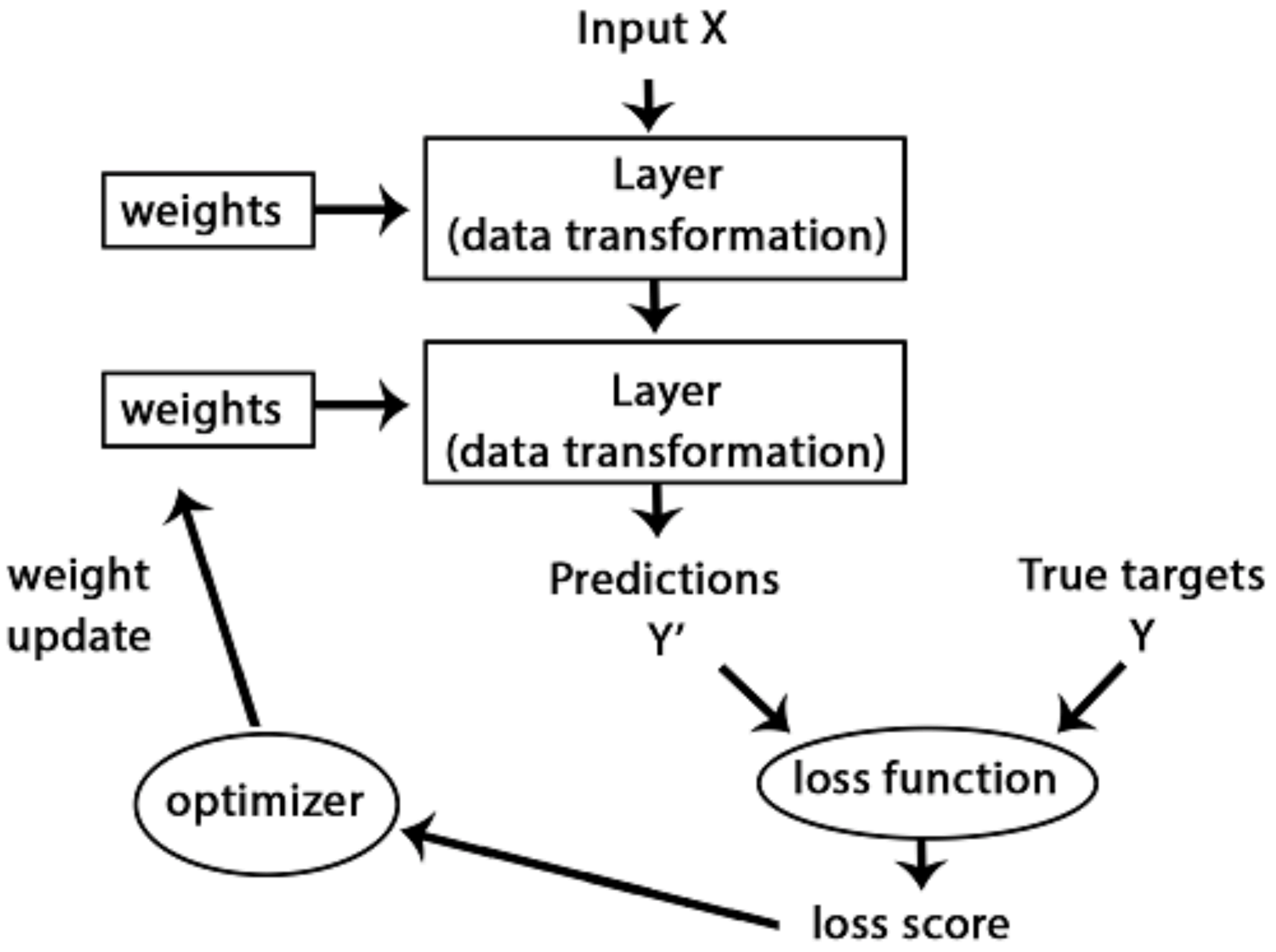
RELU



Leaky RELU

- Activation functions take many forms and transform the hidden layers so that they are the same domain as the desired prediction

Deep learning estimation framework

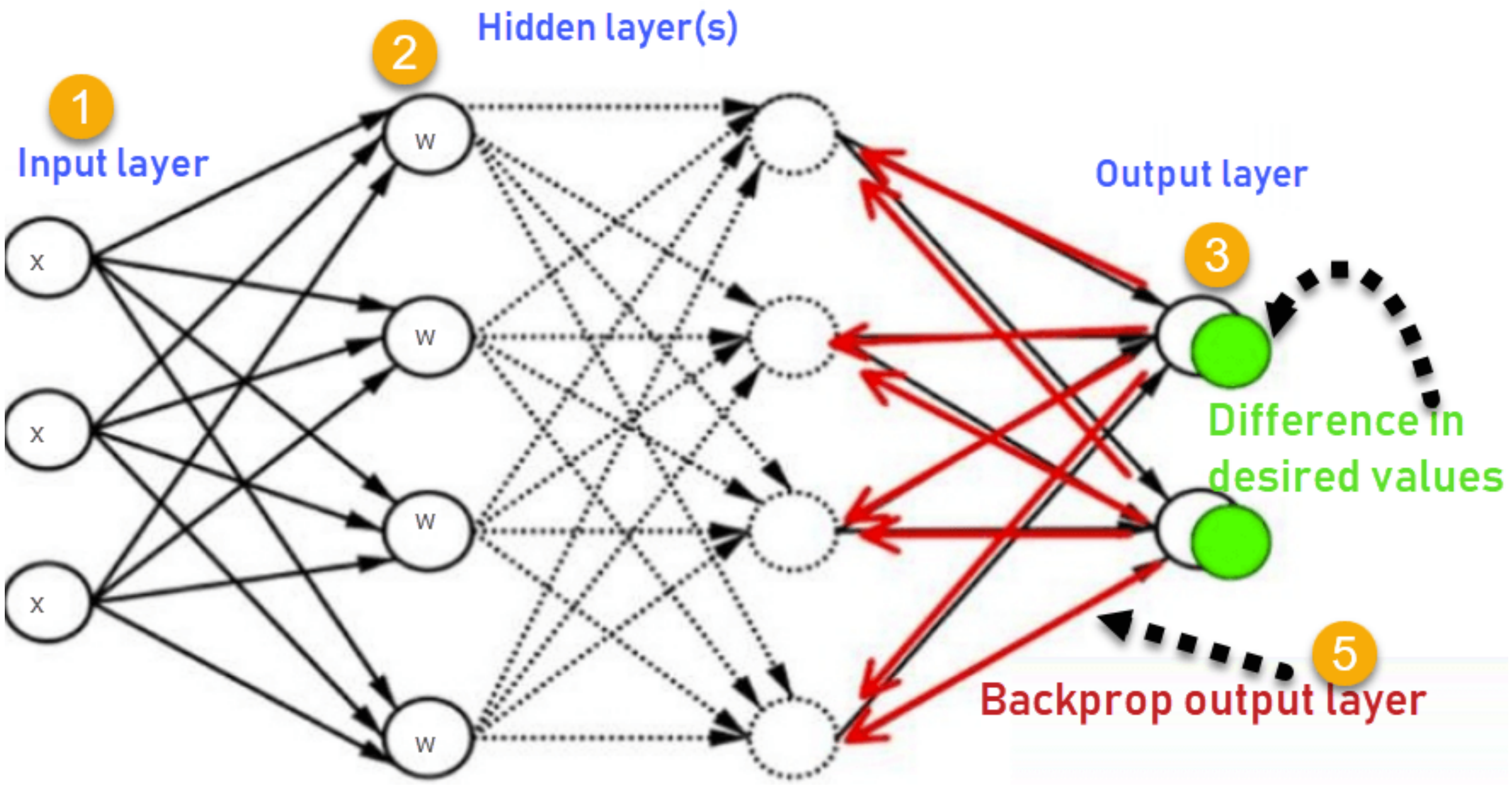


Backpropagation to Optimize Model Parameters

- Model weights (parameters) are chosen through a method called backpropogation

- Based on '86 Rumelhart, Hinton, and Williams

- Learning model weights is 90% of the work of deep learning!

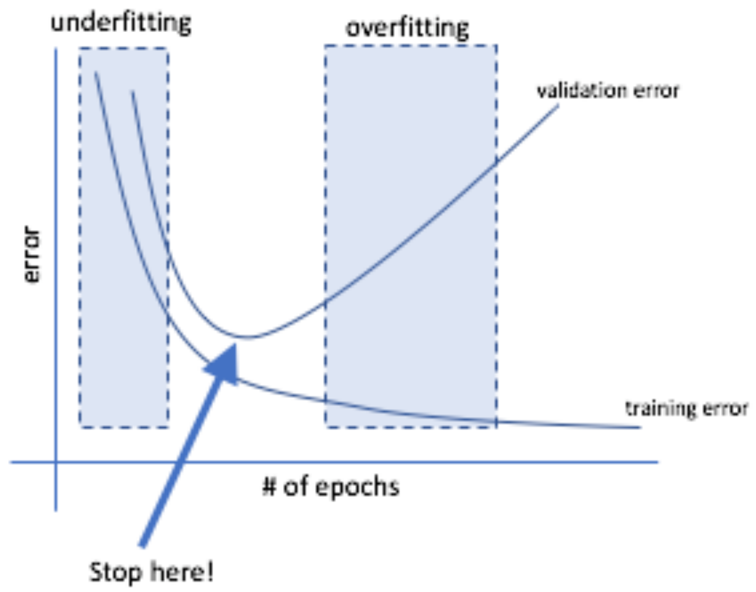


Methods of Preventing Overfitting

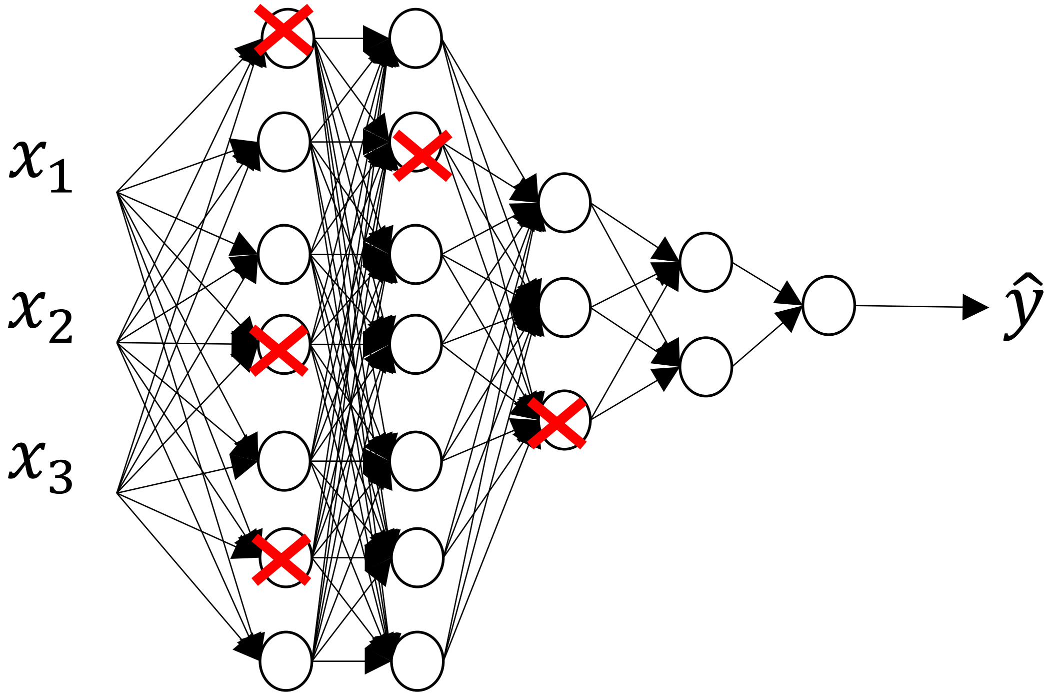
- Because deep learning models are so flexible we constantly worry about them overfitting.

- **Methods to reduce overfitting:**

1. Add regularization to our weights.
2. Reduce the capacity of the network (fewer layers/hidden units)
3. Add dropout
4. Get more training data.



Dropout

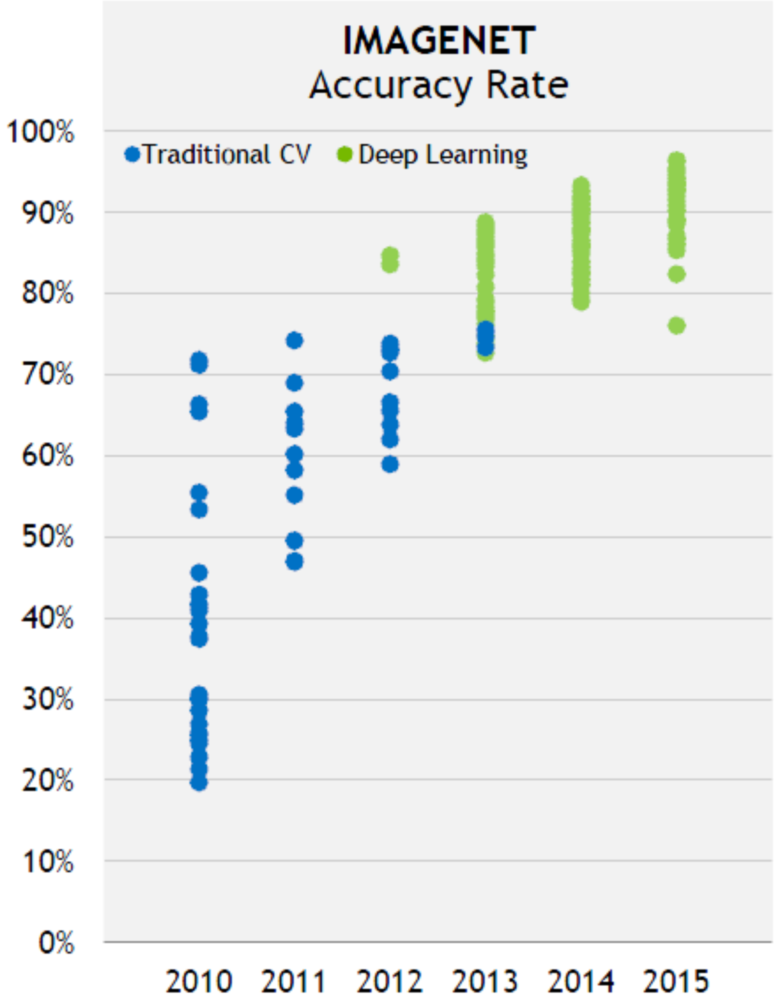


- Dropout, applied to a layer, consists of randomly dropping out (setting to zero) a number of output features of the layer during training.
- Dropout rate is usually 0.2 to 0.5

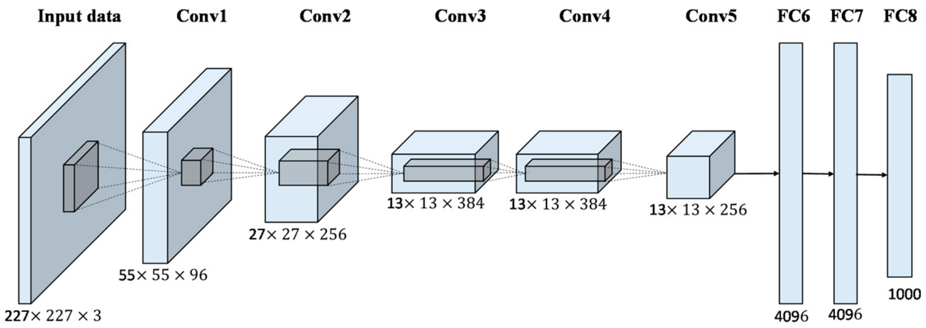
Class 10: Outline

1. Interview with Ana Rocca / Discussion of Uber Case Study
2. Break
3. Deep Learning – Feedforward Models
4. Deep Learning – CNN/Computer Vision Models
5. Deep Learning – Lab (time permitting)

Deep Learning on Images: Convolutional Neural Networks



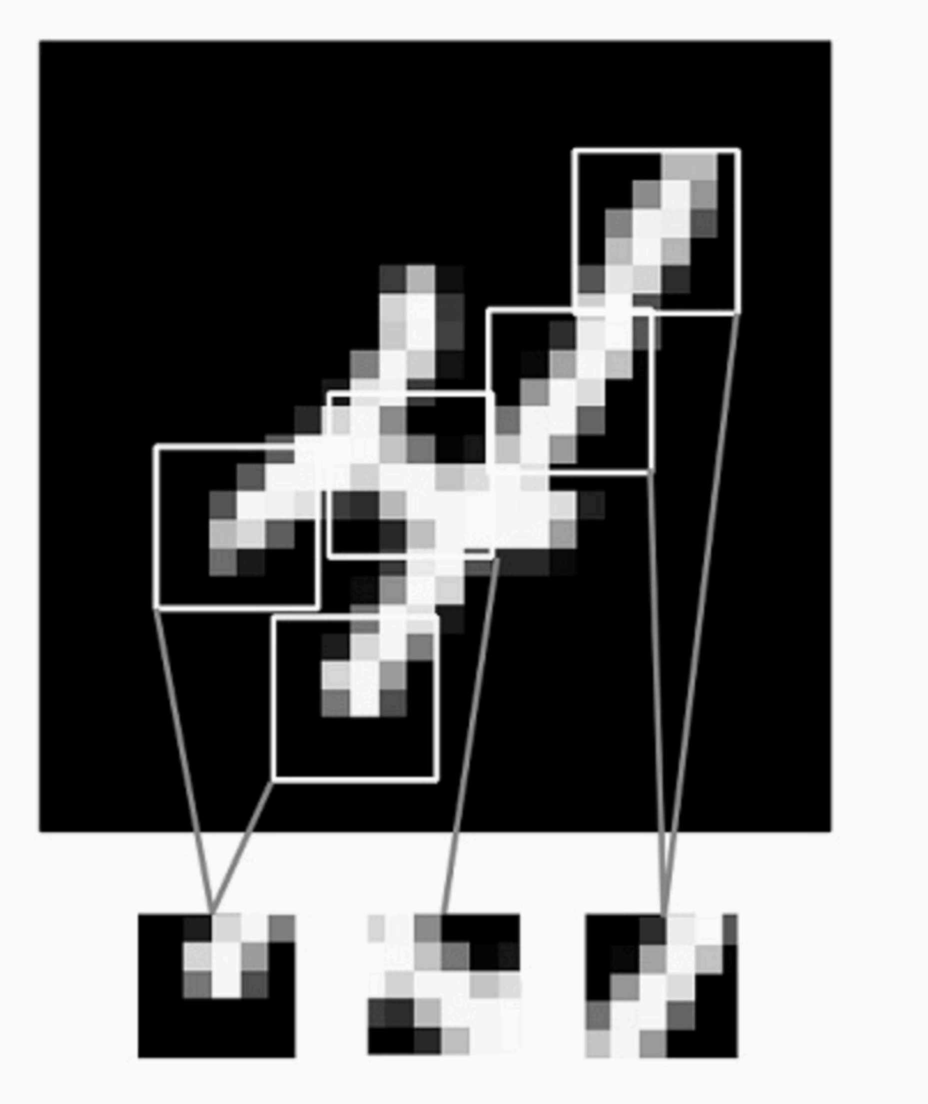
- Before deep learning, computers couldn't recognize objects from images.
- In 2012, Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton applied AlexNet, a CNN trained off of Graphical Processing Units (GPUs) to the ImageNet competition, designed to test computer's ability to see objects in images
- This kicked off a deep learning boom!



Why Are CNNs Special? Translational Invariance

- CNNs are capable of learning local features that satisfy **translational invariance**

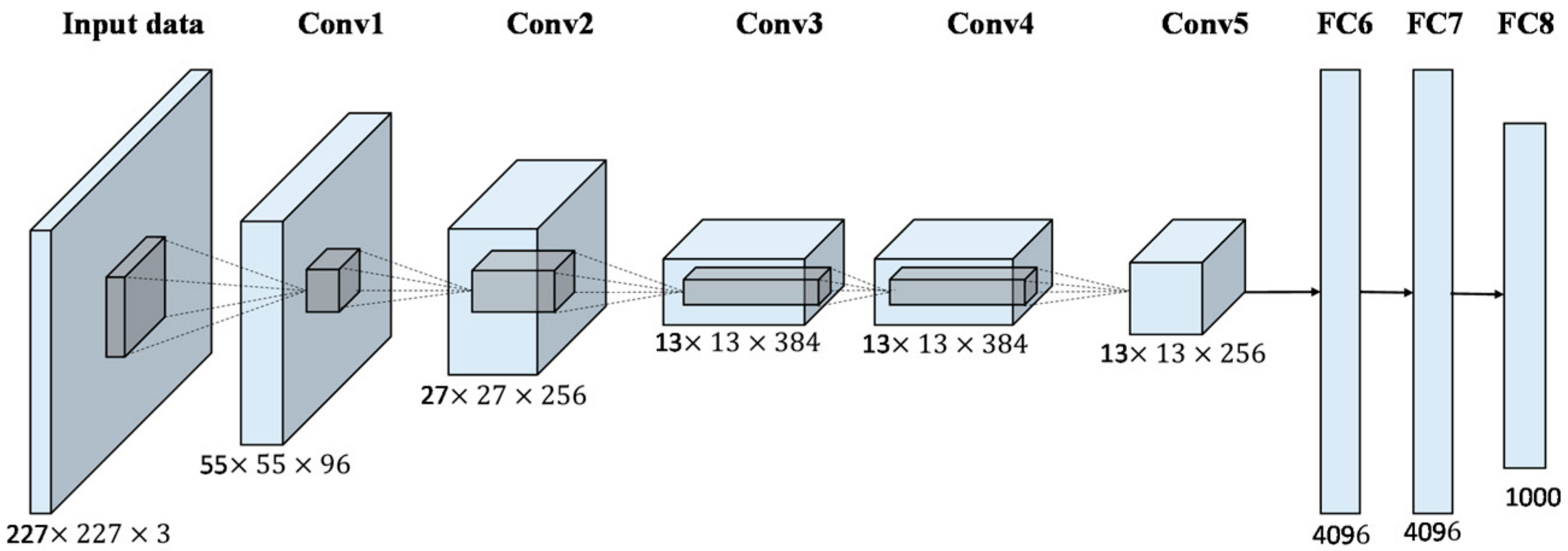
- After learning a certain pattern in the lower-right corner of a picture, a CNN can recognize it anywhere: for example, in the upper-left corner.



CNN architecture

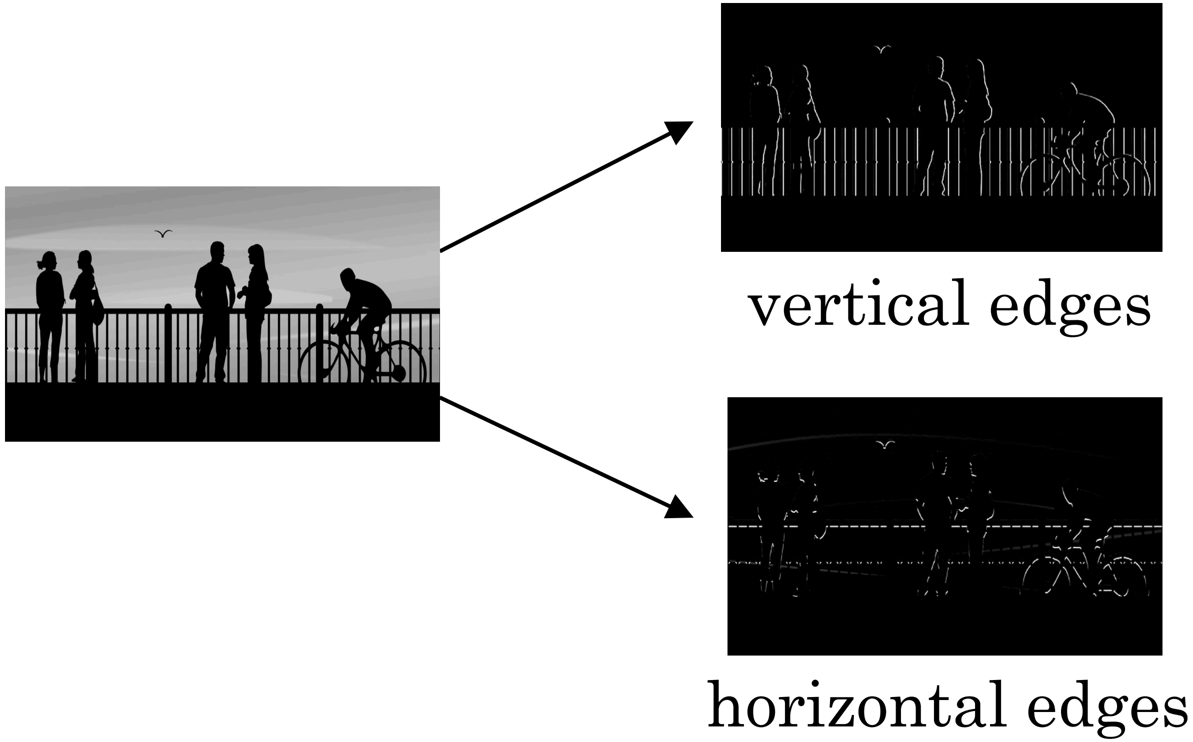
- A CNN is composed of a number of subsequent layers, where each layer has neurons arranged in 3 dimensions: width, height and depth

- Each layer creates many **feature maps** which highlights certain aspects of an image

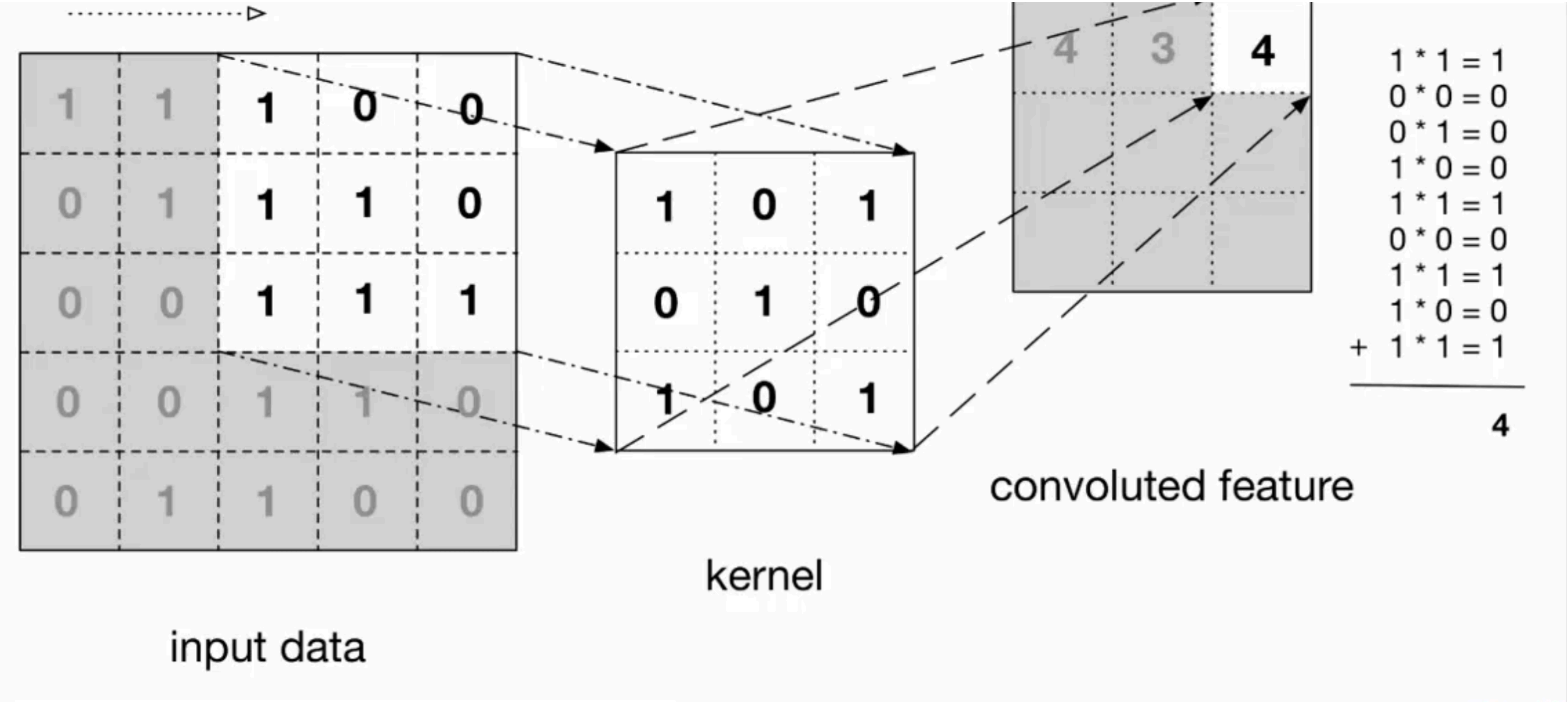


CNN Magic: Convolutional LAyers

- CNNs **convolve** an image with a **filter** or **kernel**
- These filters extract only certain salient features from the image, which vary depending on the filter used.
- CNN models "learn" these filters given image classification labels and example data

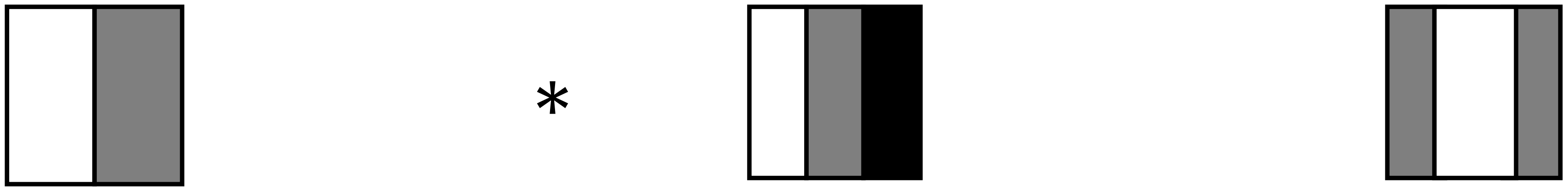


Filters in action

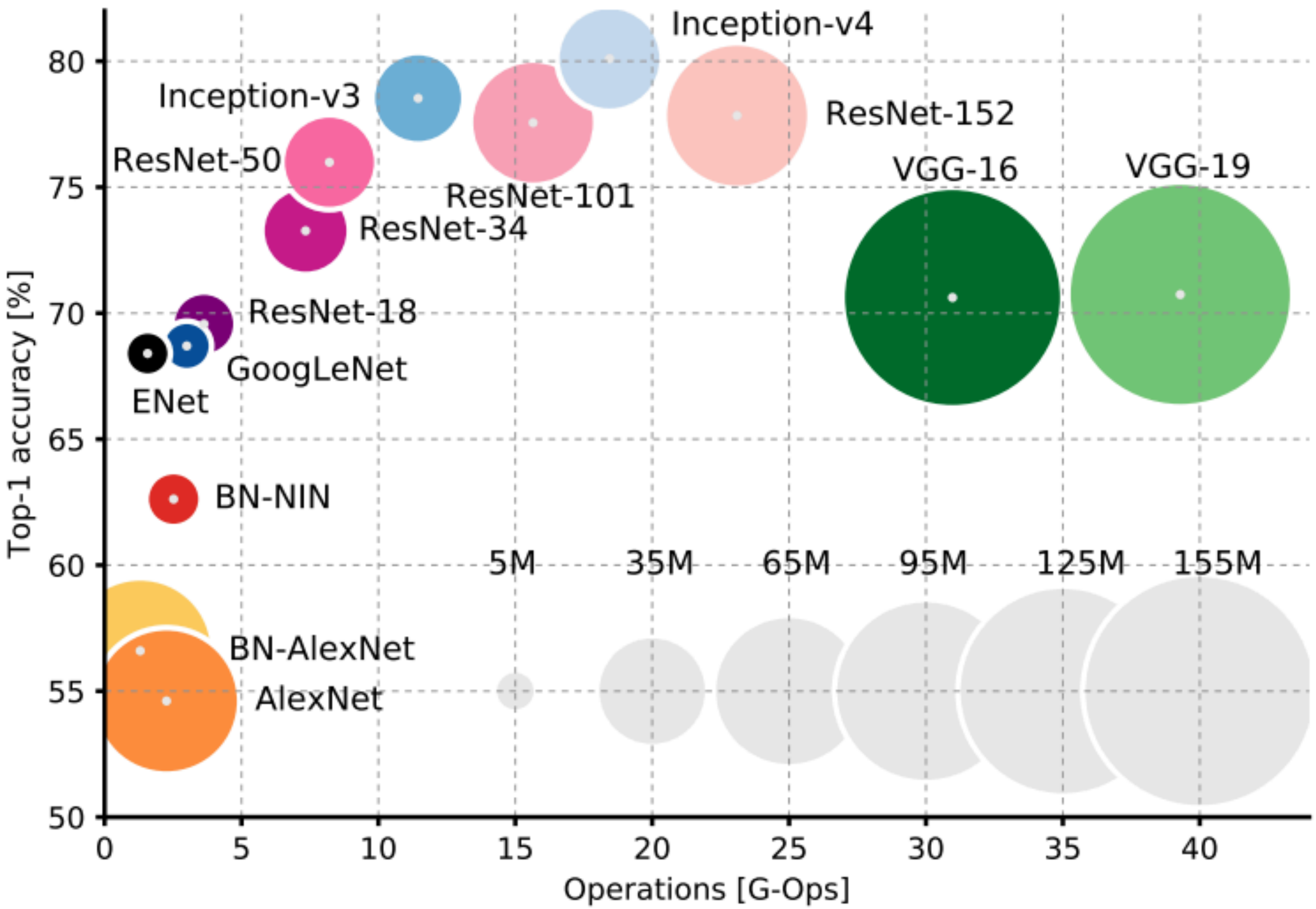


Vertical Edge Detection Filter Example

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \end{bmatrix}$$



Many Different Computer Vision Models



Top-1 accuracy: top answer must equal label

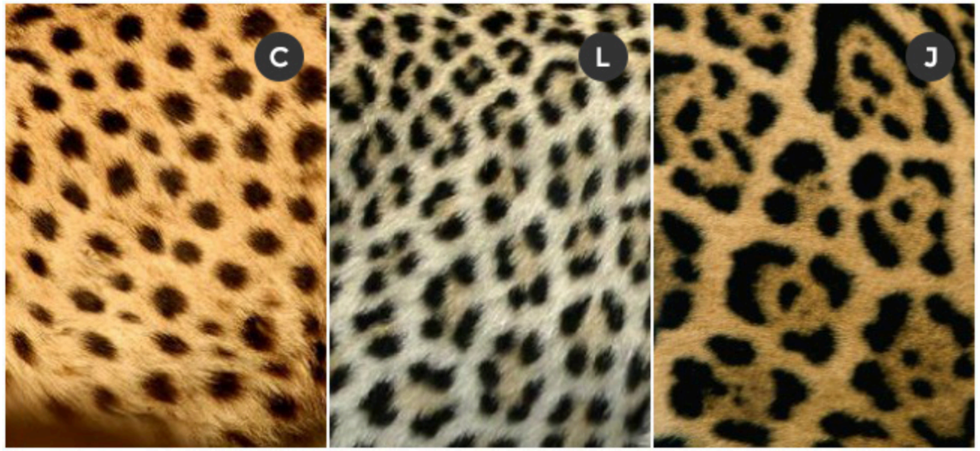
G-Ops: gigaflops, 10^9 floating point ops

Size of blobs is proportional to # of network parameters.

Side note: CNNs Don't Always Learn General Properties of Images

Source: https://rocknrollnerd.github.io/ml/2015/05/27/leopard-sofa.html

CNN Looks At Very Distinct Patterns

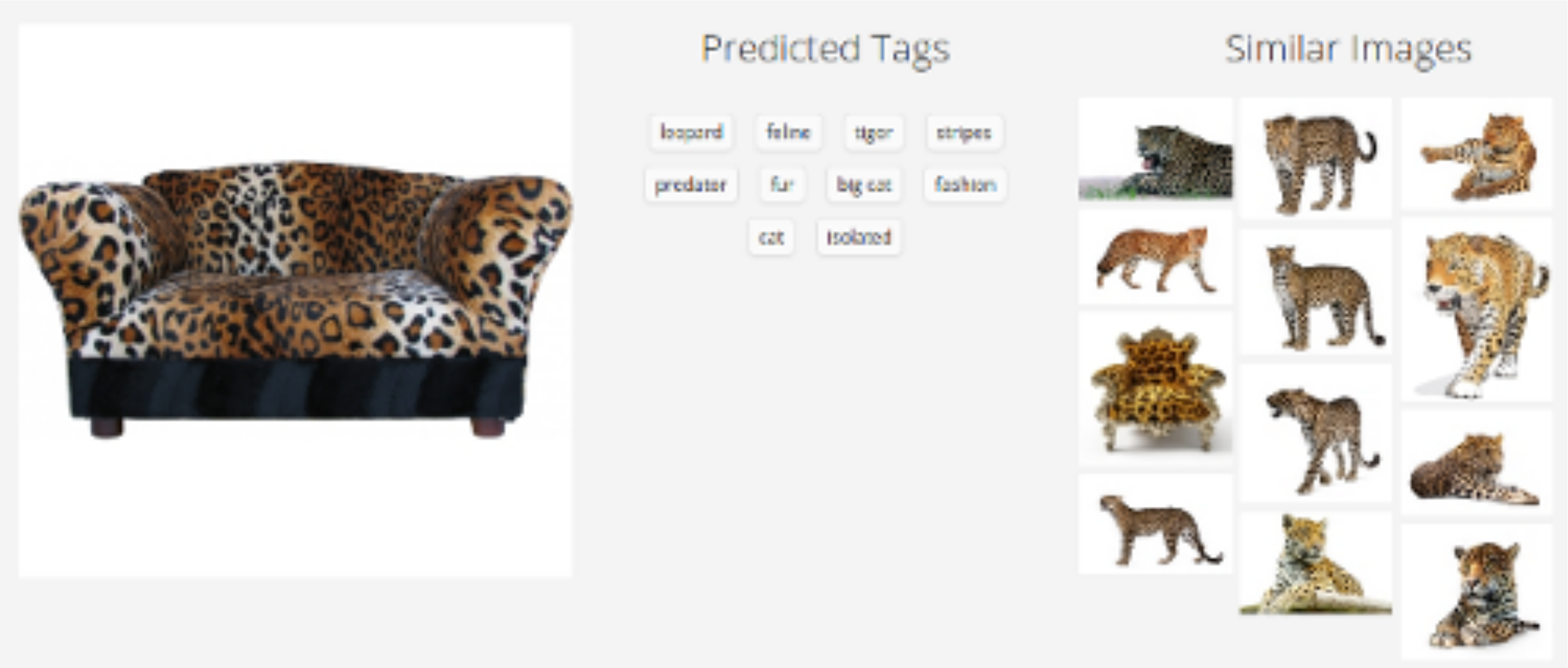


Source: https://rocknrollnerd.github.io/ml/2015/05/27/leopard-sofa.html

Are CNNs Easily Fooled?

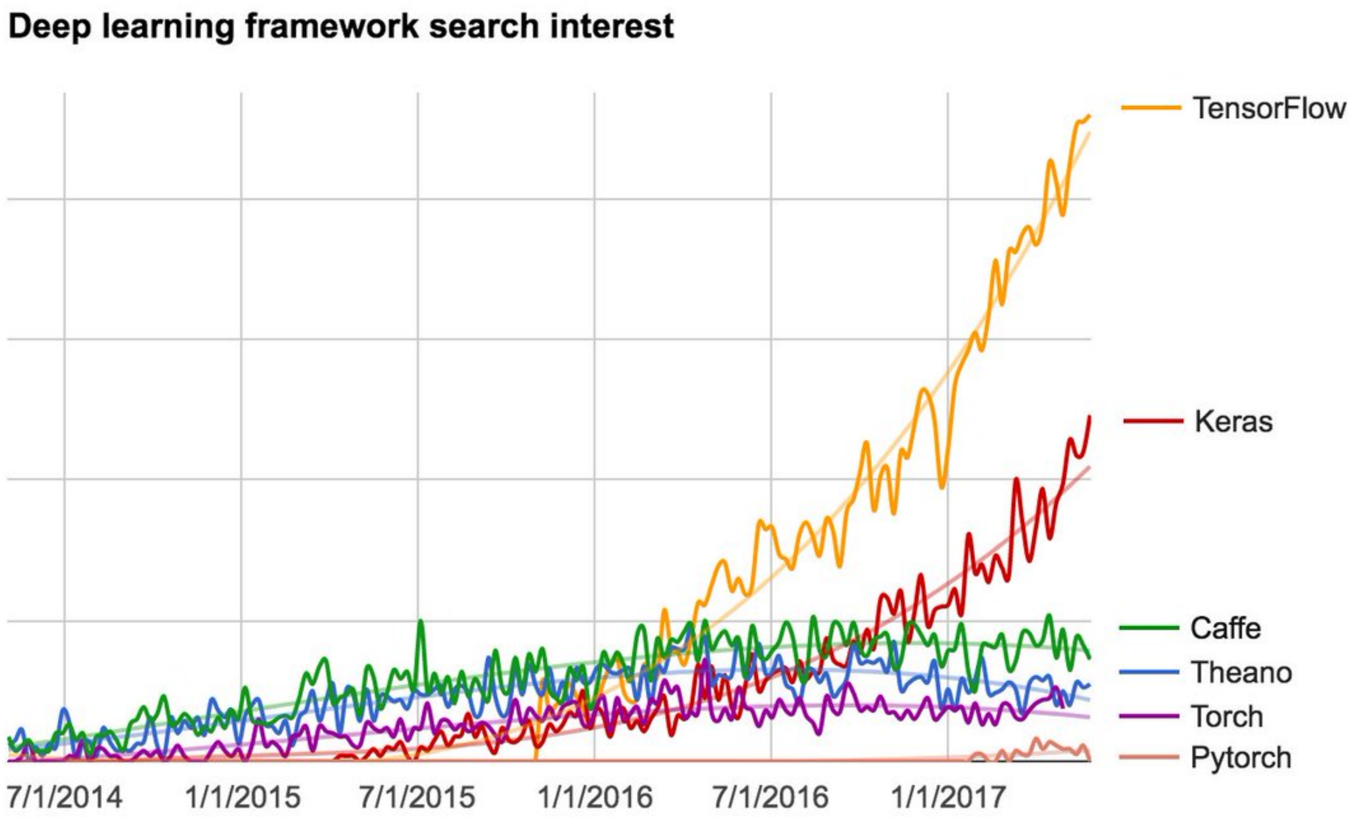
Source: https://rocknrollnerd.github.io/ml/2015/05/27/leopard-sofa.html

Are CNNs Easily Fooled? Yes!



Source: https://rocknrollnerd.github.io/ml/2015/05/27/leopard-sofa.html

Popular Deep Learning Libraries



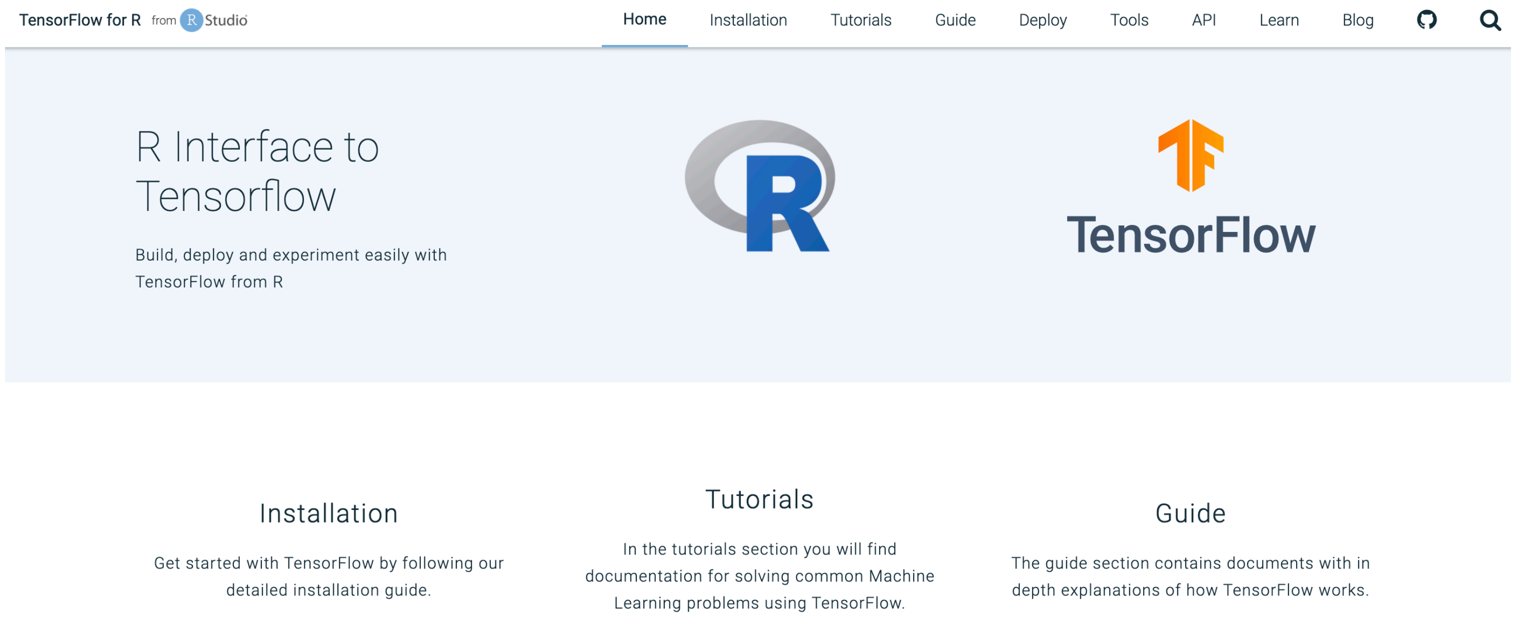
- Tensorflow (Google) is one of the most popular deep learning libraries

- Keras is a higher level package that lets us quickly build deep learning models using Tensorflow

- PyTorch is also increasingly popular

https://tensorflow.rstudio.com/

Deep Learning Models in R/Keras/Tensorflow



- R has excellent integration with Keras using the package ‘keras’

https://tensorflow.rstudio.com/

Deep Learning for Digit Recognition

Communicated by Dana Ballard

Backpropagation Applied to Handwritten Zip Code Recognition

Y. LeCun
B. Boser
J. S. Denker
D. Henderson
R. E. Howard
W. Hubbard
L. D. Jackel
AT&T Bell Laboratories Holmdel, NJ 07733 USA

The ability of learning networks to generalize can be greatly enhanced by providing constraints from the task domain. This paper demonstrates how such constraints can be integrated into a backpropagation network through the architecture of the network. This approach has been successfully applied to the recognition of handwritten zip code digits provided by the U.S. Postal Service. A single network learns the entire recognition operation, going from the normalized image of the character to the final classification.

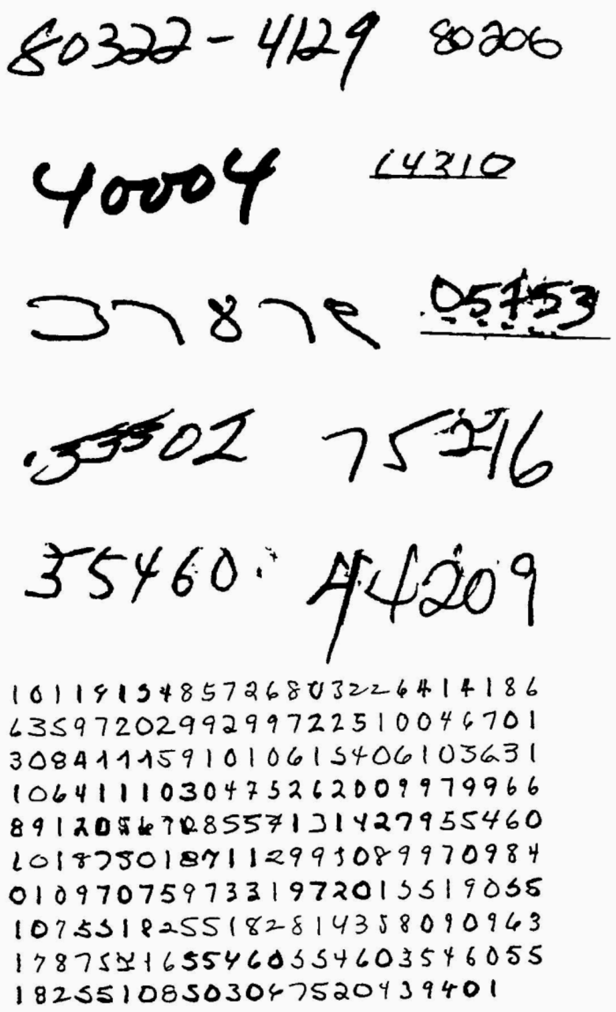


Figure 1 Examples of original zip codes (top) and normalized digits from the testing set (bottom).

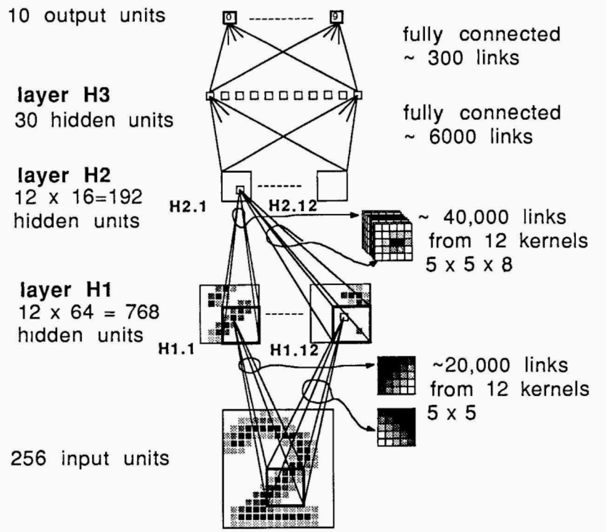
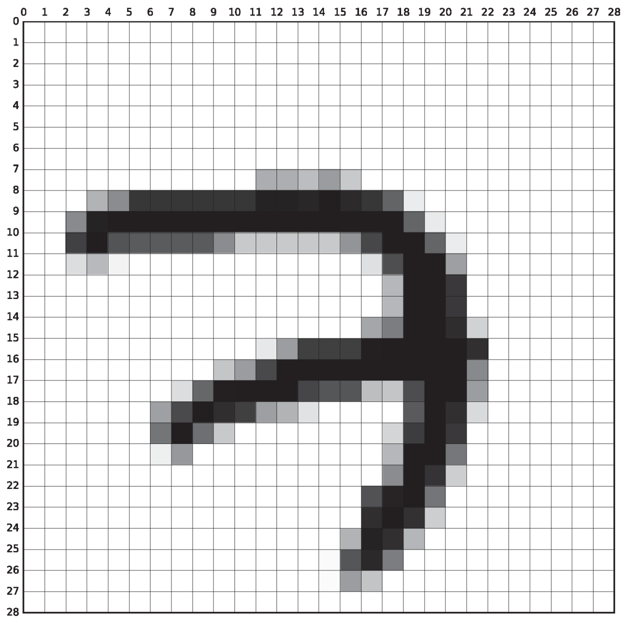


Figure 3 Log mean squared error (MSE) (top) and raw error rate (bottom) versus number of training passes

https://tensorflow.rstudio.com/

MNIST Dataset to Estimate Hand Written Digits (0,1,…,9)



(**a**) MNIST sample belonging to the digit '7'.

(**b**) 100 samples from the MNIST training set.

- MNIST is an example dataset with 1000s of hand-written digits

- We will build deep learning models in Keras/TensorFlow to predict digits from hand-written images

Lab Time! (Time Permitting)

lab_class_10_artificial_intelligence.R

```r
#-------------------------------------------
# Setup
#-------------------------------------------
rm(list = ls())
# note, run install_keras() after you have
# installed the package 'keras'
library('keras')

# this deletes any existing keras objects
k_clear_session()

#-------------------------------------------
# Load MNIST dataset
#-------------------------------------------
# load the MNIST dataset using this keras function
mnist <- dataset_mnist()
summary(mnist)
dim(mnist$train$x)


```