Introduction to Computer Science:

Machine Learning

June 2020

Honguk Woo

Many slides are from Dr. Andrew Ng's ML class notes

Topics of Intro. to Computer Science

1. Course overview
2. Data representation → System Program
3. Computer Components → Computer Architecture
4. Python programming 1 → Programming
5. Python programming 2
6. Python programming 3
7. Algorithm → Algorithm, Problem Solving
8. Algorithm – Sorting
9. Algorithm – Backtracking
10. Operating system → OS
11. Network → NW
12. Programming language → PL
13. Data structure → DS
14. Machine learning → ML
15. Exam

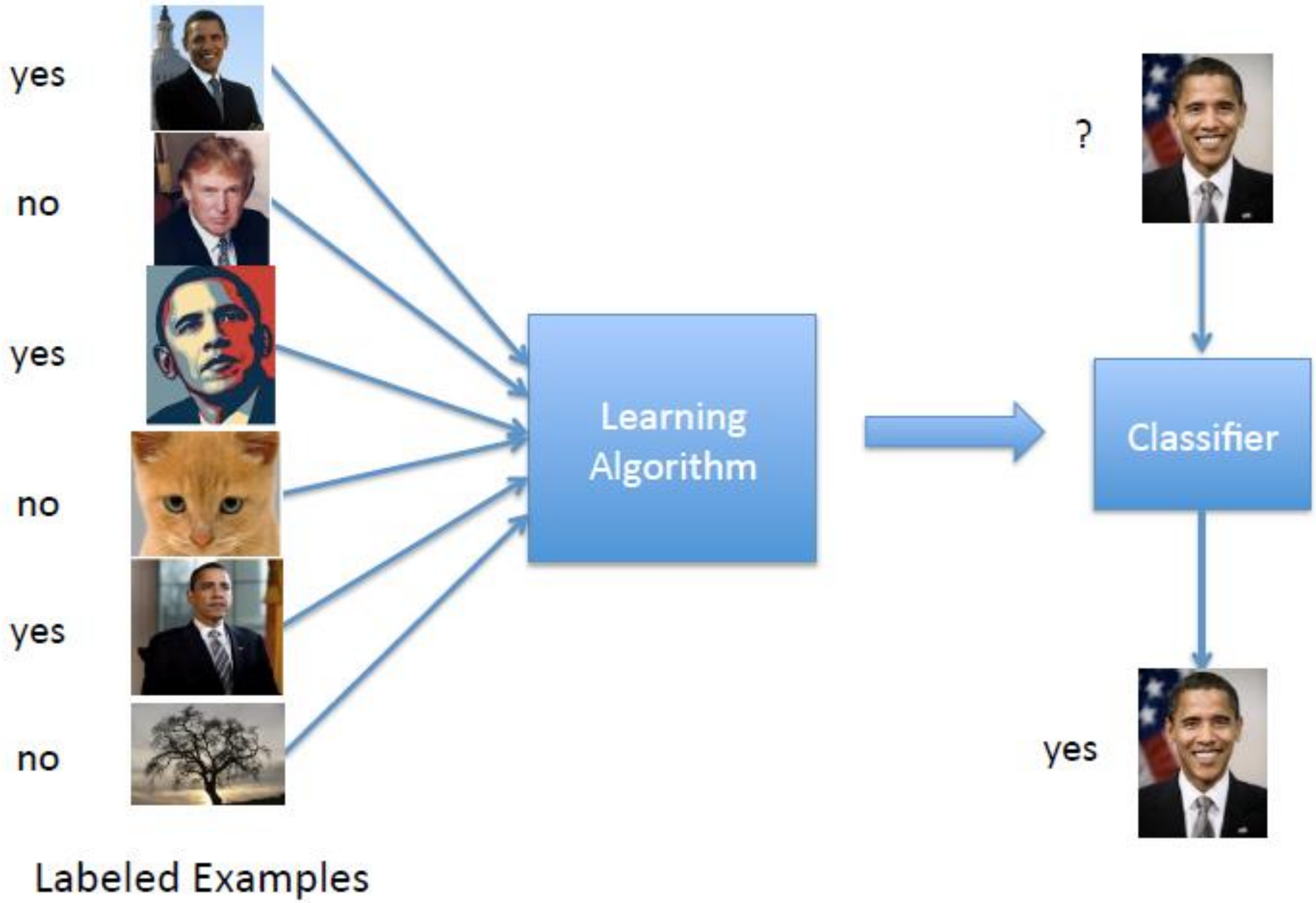
ML (machine learning)

- What is ML ?
  - Think about how to recognize faces, how to recommend movies, how to decide which web pages are shown for search query ?
  - Hard to write programs for those jobs. We don't know what program to write because we don't know how our brain does it. The program might be horrendously complicated

  - Recall what algorithm is; seem to be difficult to invent an algorithm and implement it for e.g., recognizing faces; but you can show a computer how to solve the task
    - Logical thinking -> Observation

  - ML is the practice of programming a computer to learn to solve a task through experience, rather than directly programming it to solve the task

Learning from examples



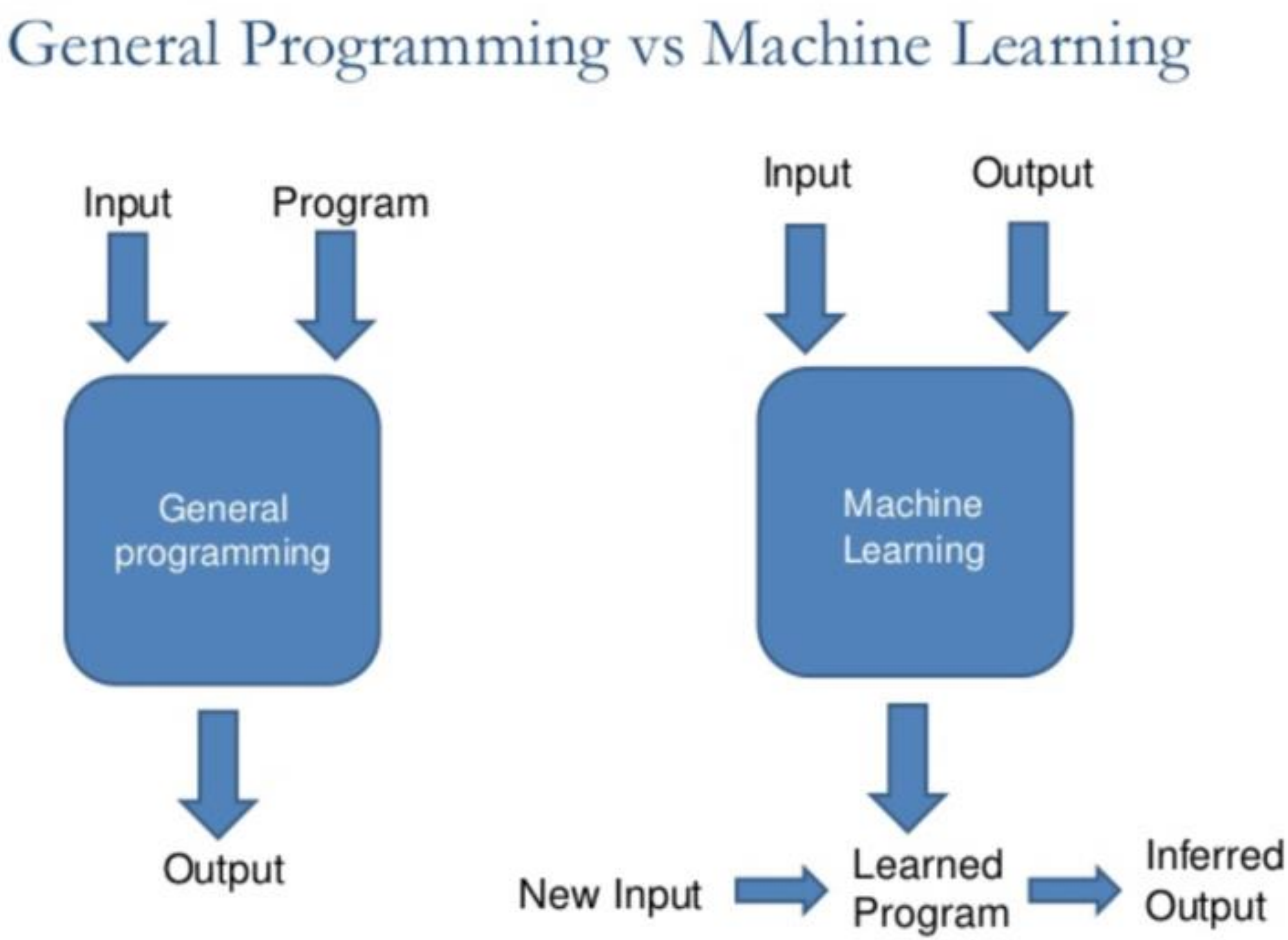
ML (machine learning)

- Tom Mitchell (1998) Well-posed Learning Problem:
  - A computer program is said to *learn* from **experience E** with respect to some **task T** and some **performance measure P**, if its performance on T, as measured by P, **improves with experience E**

Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam. What is the **task T** in this setting?

- Classifying emails as spam or not spam.
- Watching you label emails as spam or not spam.
- The number (or fraction) of emails correctly classified as spam/not spam.
- None of the above—this is not a machine learning problem.

ML (machine learning)



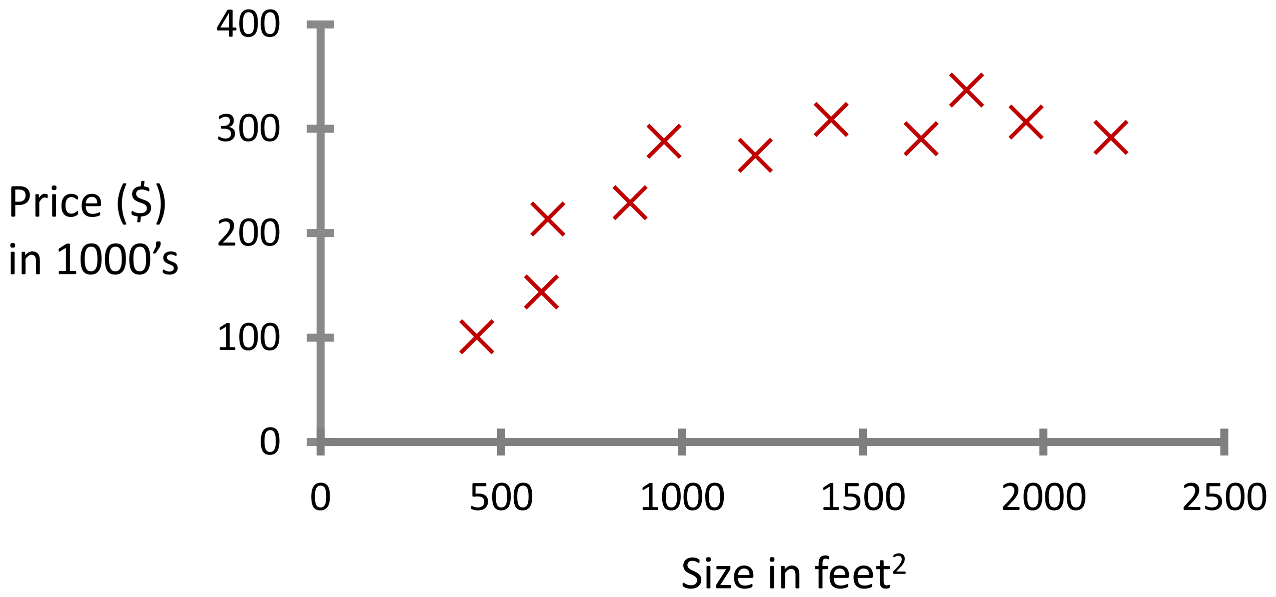
https://www.experfy.com/blog/the-beginner-s-guide-to-understanding-machine-learning-for-businesses

Machine learning terminology & algorithms

- ML systems learn how to combine input to produce useful predictions on never-before-seen data
  - Label : is that we’re predicting
  - Feature: input variables
  - Example : data
    - Labeled data
    - Unlabeled data

- ML algorithms
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning

e.g., Housing price prediction



<u>Supervised Learning</u>

"right answers" given

<u>Regression</u>: Predict continuous valued output (price)

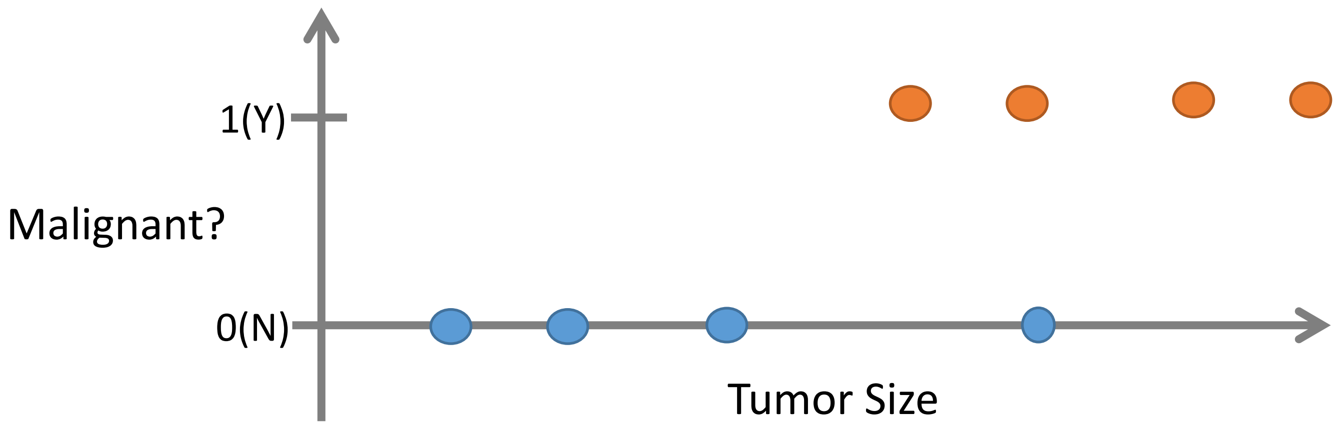
What's the value of a 1700 feet^2 house ?

Supervised learning

- The most common area of ML
- Learned by examples (labeled data)
- “supervised” means
  - Training the model like having a teacher supervise the whole process
  - The process of teaching a model by feeding it input data as well as correct output data.
- Regression
- Classification

e.g., Breast cancer

Breast cancer (malignant, benign)

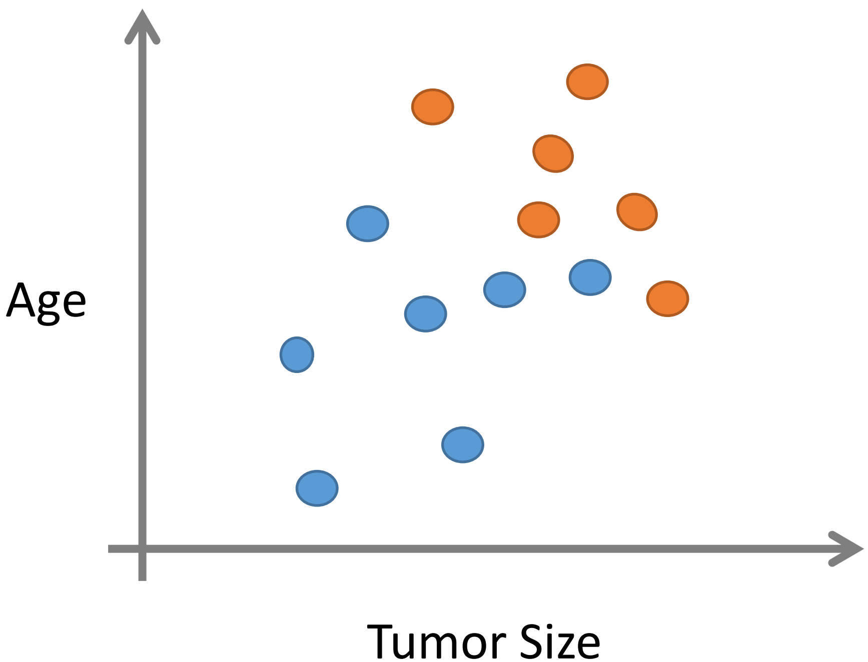


<u>Classification</u>

Discrete valued output (0 or 1)

Is a given tumor malignant or benign?

e.g., Breast cancer



- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape

…

Q

You’re running a company, and you want to develop learning algorithms to address each of two problems.

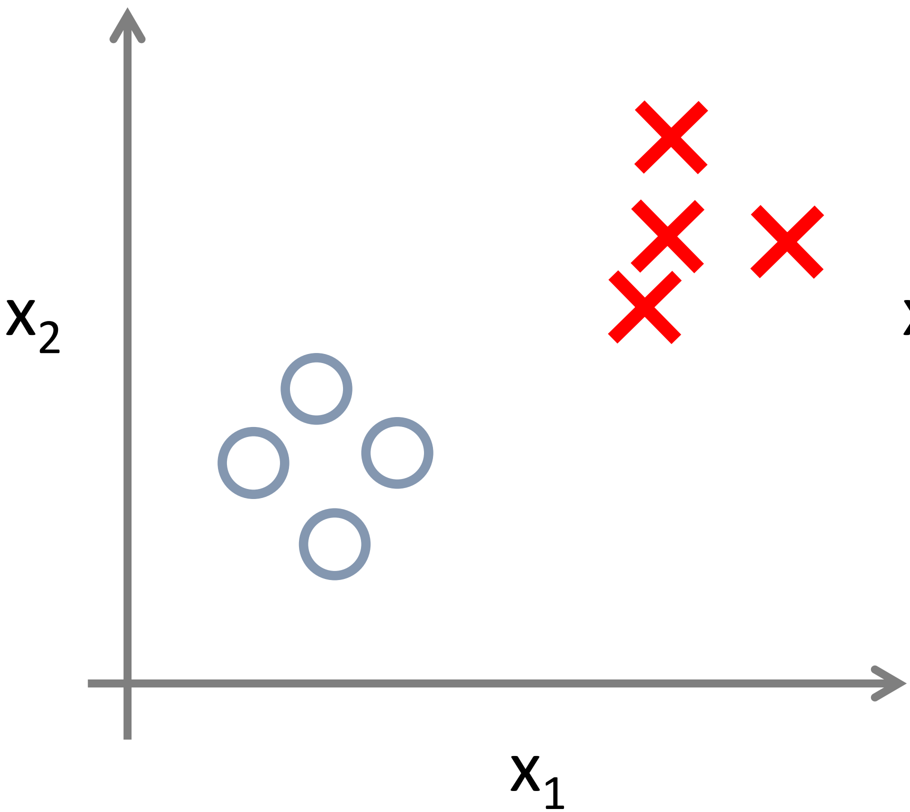
Problem 1: You have a large inventory of identical items. You want to predict how many of these items will sell over the next 3 months.
Problem 2: You’d like software to examine individual customer accounts, and for each account decide if it has been hacked/compromised.

Should you treat these as classification or as regression problems?

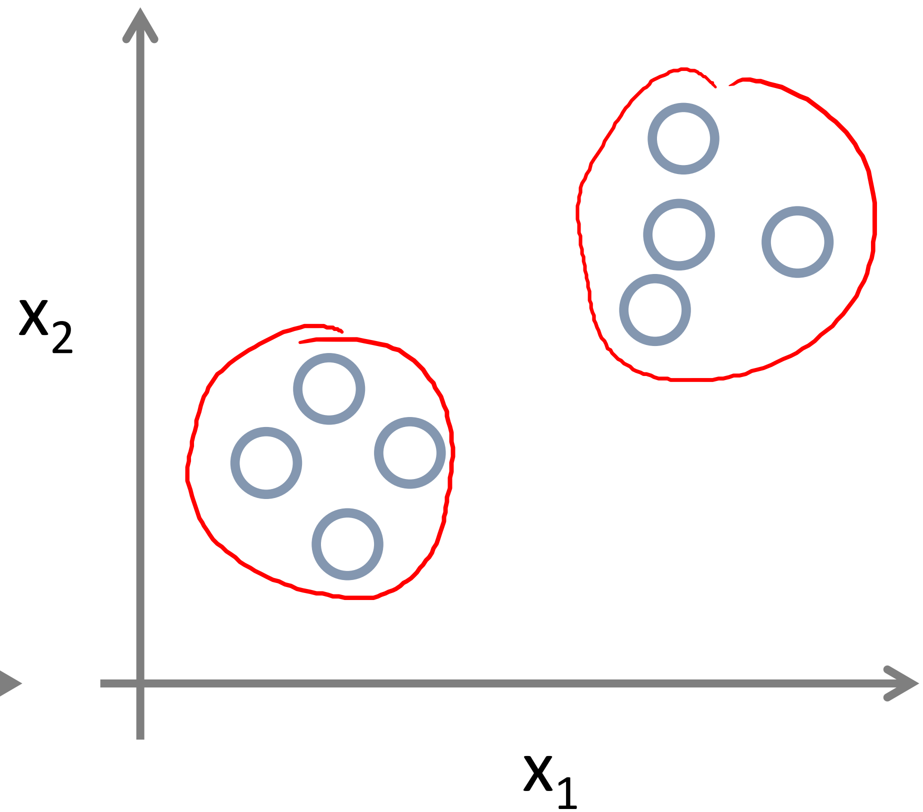
- Treat both as classification problems.
- Treat problem 1 as a classification problem, problem 2 as a regression problem.
- Treat problem 1 as a regression problem, problem 2 as a classification problem.
- Treat both as regression problems.

Supervised vs Unsupervised

Supervised Learning



Unsupervised Learning



Q

Of the following examples, which would you address using an unsupervised learning algorithm? (Check all that apply.)

- Given email labeled as spam/not spam, learn a spam filter.
- Given a set of news articles found on the web, group them into set of articles about the same story.
- Given a database of customer data, automatically discover market segments and group customers into different market segments.
- Given a dataset of patients diagnosed as either having diabetes or not, learn to classify new patients as having diabetes or not.

Google News

Search for topics, locations & sources

- Top stories
- For you
- Following
- Saved searches

- COVID-19

- U.S.
- World
- Business
- Technology
- Entertainment
- Sports

Headlines

More Headlines

COVID-19 news: See the latest coverage of the coronavirus (COVID-19)

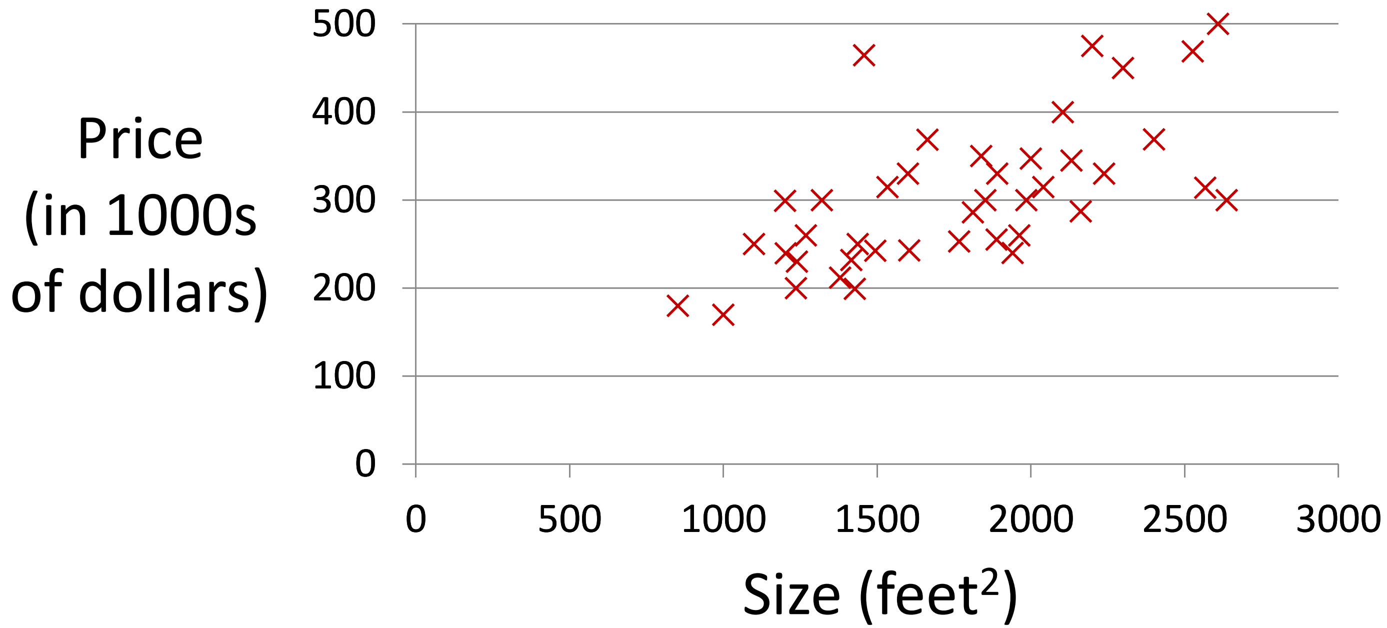
Protests against police violence in Washington, D.C.

The Sun · 4 hours ago

- Protests across the globe after George Floyd's death
  CNN · 9 hours ago
- UK anti-racism protesters clash with mounted police
  Reuters UK · 1 hour ago
- George Floyd protest news: Live news from protests today
  NBCNews.com · 10 hours ago
- This powerful protest movement deserves more from the press: Journalists must demand change
  Salon · 5 hours ago

View Full Coverage

Regression : e.g., **Housing Prices**



<u>Supervised Learning</u>

Given the “right answer” for each example in the data.

<u>Regression Problem</u>

Predict real-valued output

Regression : Housing Prices dataset

Training set of housing prices

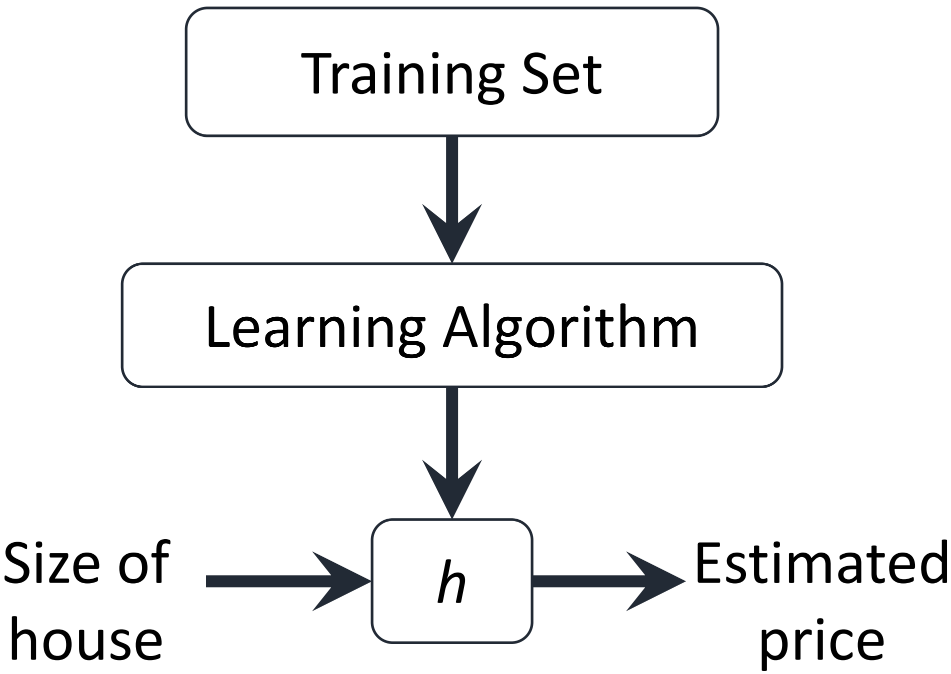
Size in feet^2 (x)	**Price ($) in 1000's (y)**
2104	460
1416	232
1534	315
852	178
…	…

Notation:

- **m** = Number of training examples
- **x**’s = “input” variable (features)
- **y**’s = “output” variable (“target or desired” variable, label)

Regression : **hypothesis**

Hypothesis (Model)



How do we represent *h* ?

Linear regression with one variable.
Univariate linear regression.

Regression : **hypothesis**

Training dataset

Size in feet² (x)	**Price ($) in 1000's (y)**
2104	460
1416	232
1534	315
852	178
…	…

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

How to choose Parameters θ_i?

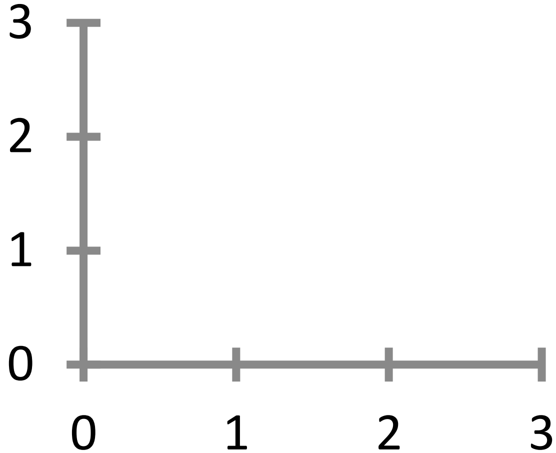
Idea: Choose the parameters so that

$h_\theta(x)$ is close to y (= **cost function**)

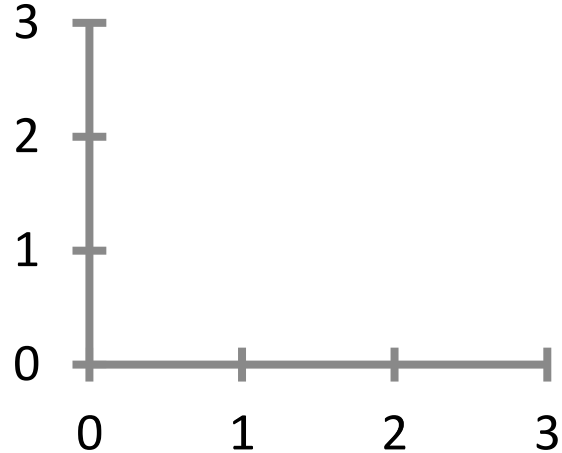
for our training examples (x, y)

Regression : **cost function**

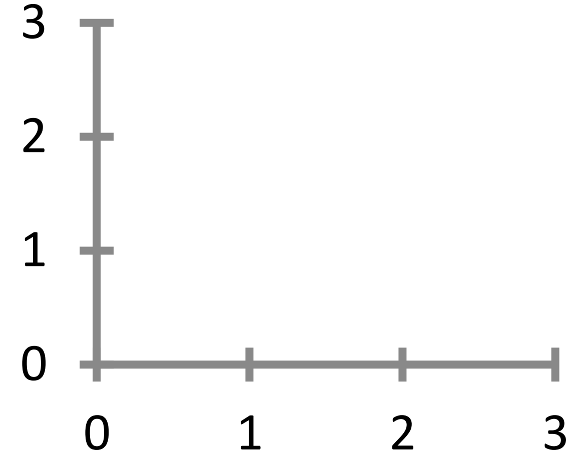
$$h_\theta(x) = \theta_0 + \theta_1 x$$



$$\theta_0 = 1.5$$
$$\theta_1 = 0$$



$$\theta_0 = 0$$
$$\theta_1 = 0.5$$



$$\theta_0 = 1$$
$$\theta_1 = 0.5$$

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

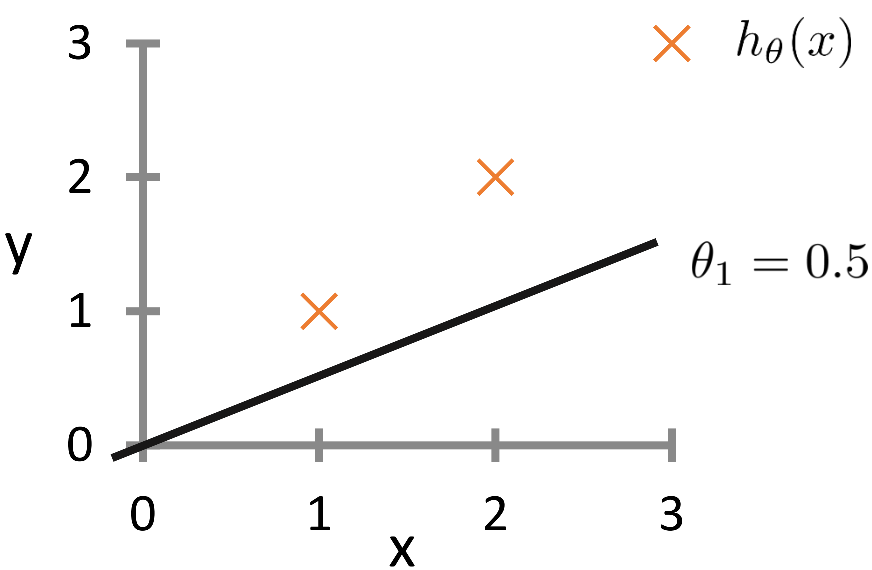
Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Regression : **cost function**

For simplicity, assume $\theta_0 = 0$

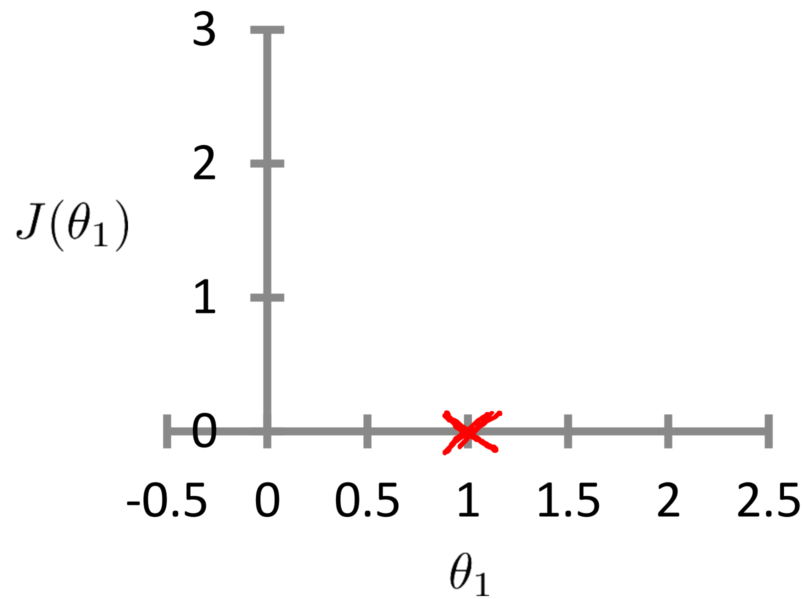
$h_\theta(x)$

(for fixed θ_1, this is a function of x)

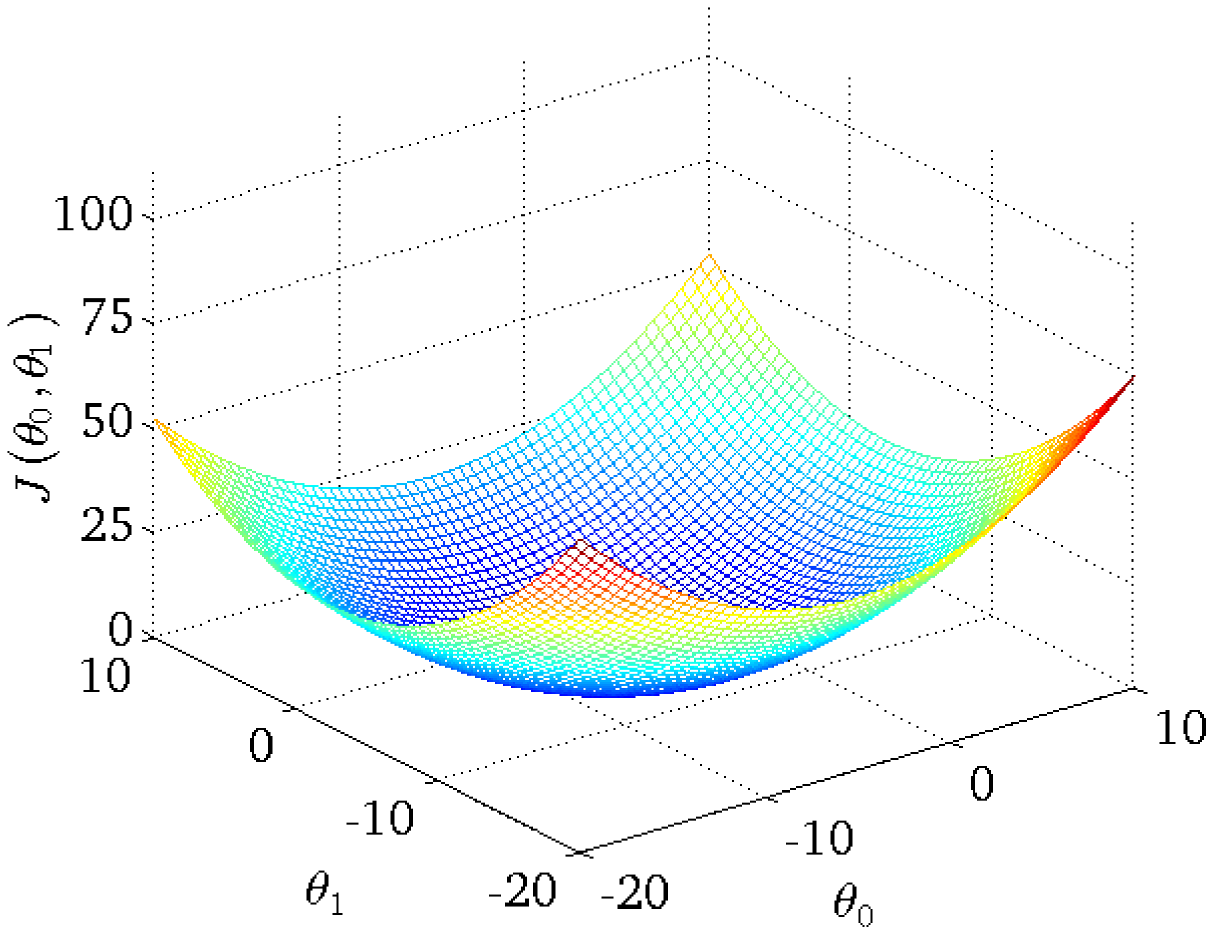


$J(\theta_1)$

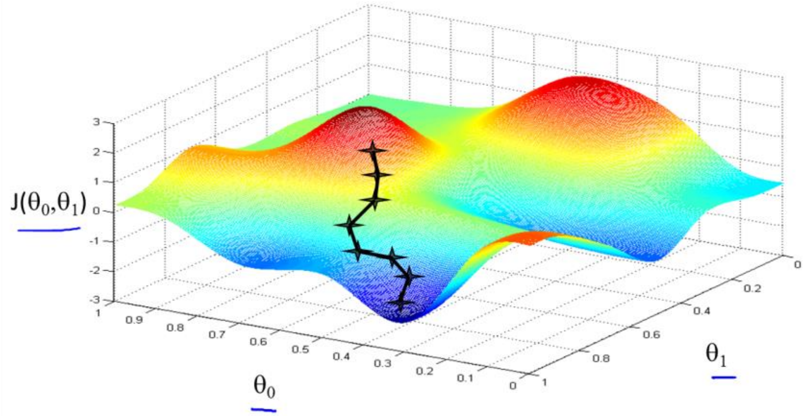
(function of the parameter θ_1)



Regression : **cost function**



Regression : **cost function**



Have some function $J(\theta_0, \theta_1)$

Want $\min_{\theta_0,\theta_1} J(\theta_0, \theta_1)$

Outline:

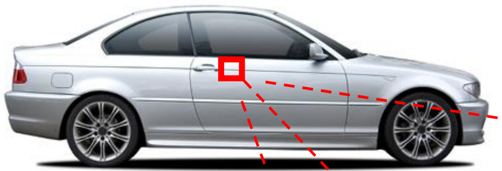
- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$ until we hopefully end up at a minimum

Gradient descent algorithm

$$
\begin{aligned}
&\text{repeat until convergence } \{ \\
&\quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \qquad (\text{for } j = 0 \text{ and } j = 1) \\
&\}
\end{aligned}
$$

$$
\begin{aligned}
&\text{temp0} := \theta_0 - \alpha \tfrac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) \\
&\text{temp1} := \theta_1 - \alpha \tfrac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) \\
&\theta_0 := \text{temp0} \\
&\theta_1 := \text{temp1}
\end{aligned}
$$

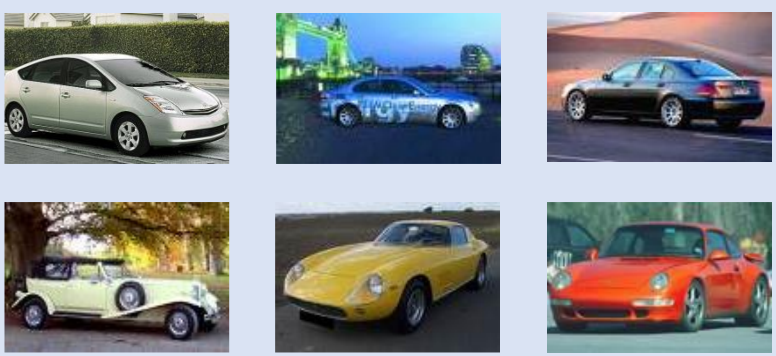
Gradient descent algorithm



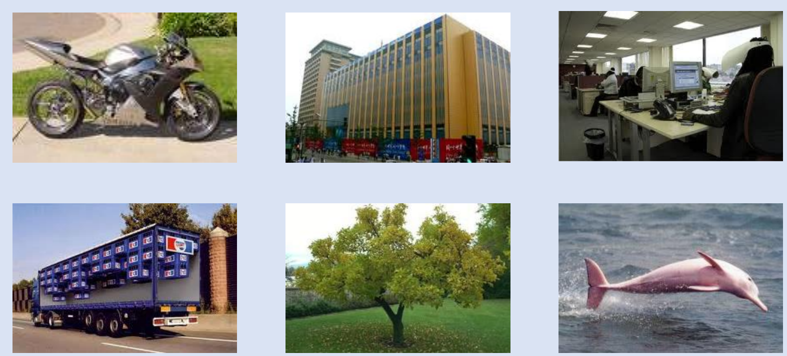
the camera sees this:

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

Computer Vision: Car detection



Cars

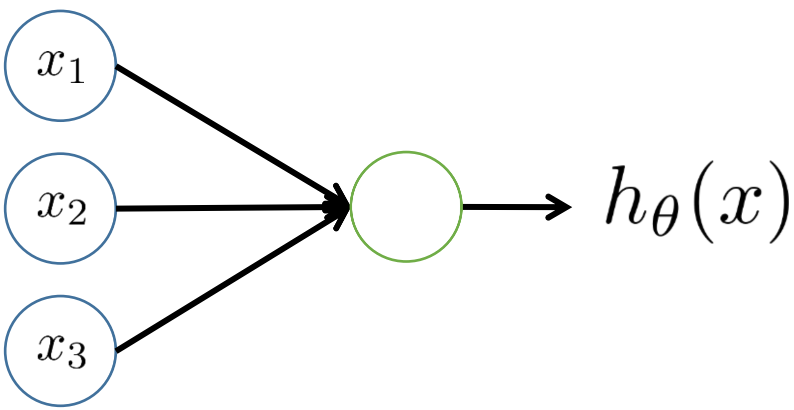


Not a car

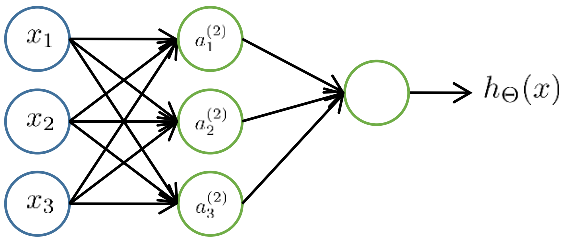
Testing:
What is this?

Neural Networks

- Origins: Algorithms that try to mimic the brain
- Was very widely used in 80s and early 90s; popularity diminished in late 90s.
- Recent resurgence: State-of-the-art technique for many applications



Neural Networks



$a_i^{(j)} =$ "activation" of unit i in layer j

$\Theta^{(j)} =$ matrix of weights controlling function mapping from layer j to layer $j+1$

$$
\begin{aligned}
a_1^{(2)} &= g(\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2 + \Theta_{13}^{(1)}x_3) \\
a_2^{(2)} &= g(\Theta_{20}^{(1)}x_0 + \Theta_{21}^{(1)}x_1 + \Theta_{22}^{(1)}x_2 + \Theta_{23}^{(1)}x_3) \\
a_3^{(2)} &= g(\Theta_{30}^{(1)}x_0 + \Theta_{31}^{(1)}x_1 + \Theta_{32}^{(1)}x_2 + \Theta_{33}^{(1)}x_3) \\
h_\Theta(x) = a_1^{(3)} &= g(\Theta_{10}^{(2)}a_0^{(2)} + \Theta_{11}^{(2)}a_1^{(2)} + \Theta_{12}^{(2)}a_2^{(2)} + \Theta_{13}^{(2)}a_3^{(2)})
\end{aligned}
$$

If network has s_j units in layer j, s_{j+1} units in layer $j+1$, then $\Theta^{(j)}$ will be of dimension $s_{j+1} \times (s_j + 1)$.

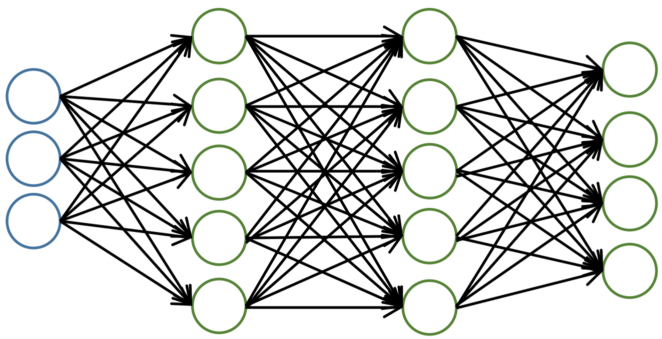
Neural Networks

Pedestrian

Car

Motorcycle

Truck



$h_\Theta(x) \in \mathbb{R}^4$

Want $h_\Theta(x) \approx \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$, $h_\Theta(x) \approx \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$, $h_\Theta(x) \approx \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$, etc.

when pedestrian when car when motorcycle