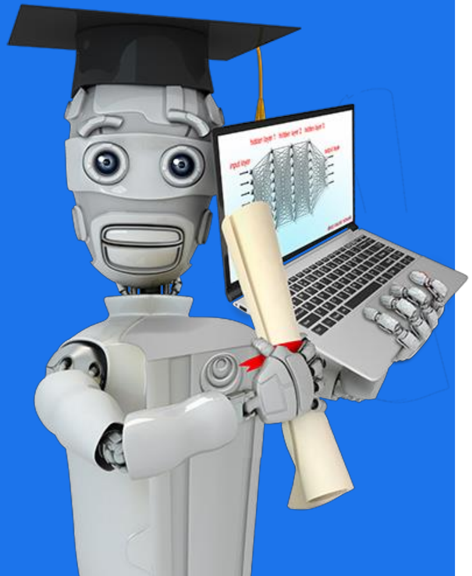
Stanford
ONLINE

DeepLearning.AI

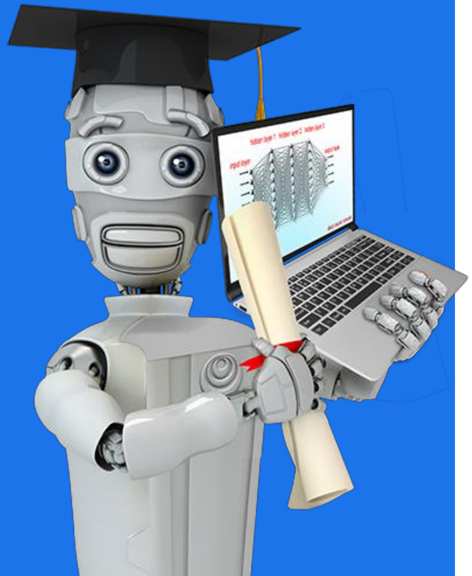


Machine Learning

Welcome!

Stanford ONLINE

DeepLearning.AI



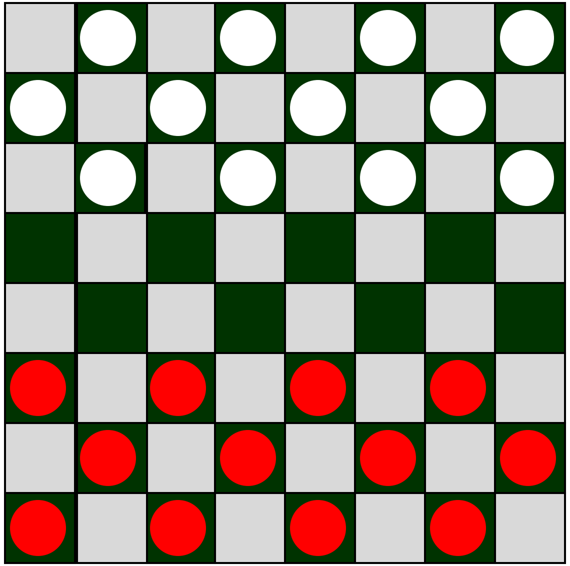
Machine Learning Overview

What is Machine Learning?

Machine learning

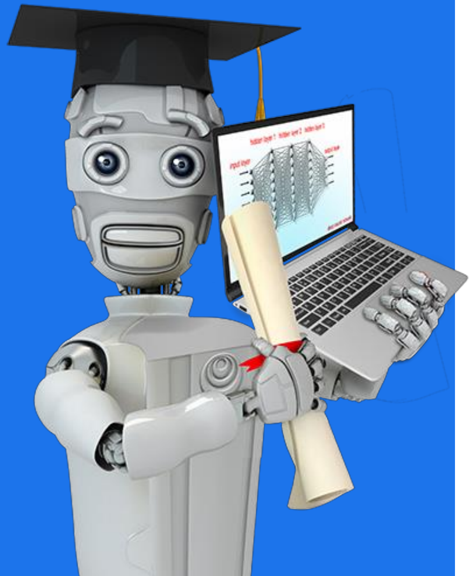
"Field of study that gives computers the ability to learn without being explicitly programmed."

Arthur Samuel (1959)



Stanford ONLINE

DeepLearning.AI



Machine Learning Overview

Supervised Learning Part 1

Supervised learning

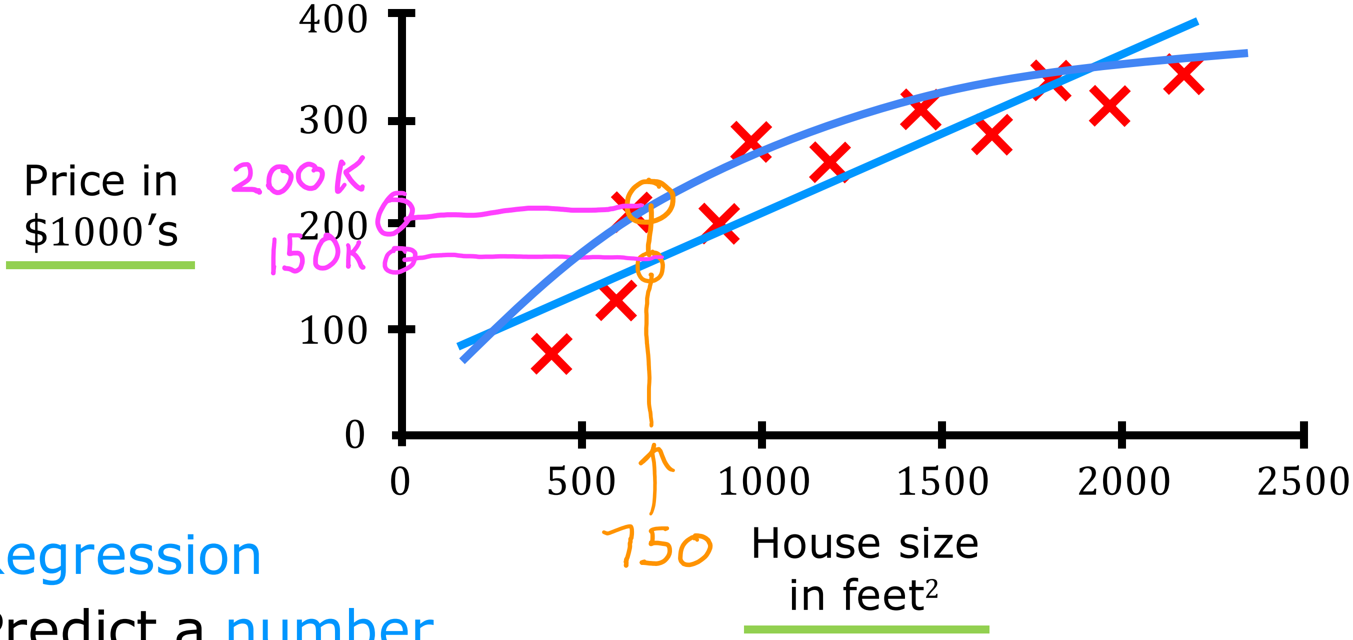
$X \longrightarrow Y$

input $\longrightarrow$ output label

Learns from being given "right answers"

Input (X)	Output (Y)	Application
email →	spam? (0/1)	spam filtering
audio →	text transcripts	speech recognition
English →	Spanish	machine translation
ad, user info →	click? (0/1)	online advertising
image, radar info →	position of other cars	self-driving car
image of phone →	defect? (0/1)	visual inspection

Regression: Housing price prediction



150,000
70,000
183,000

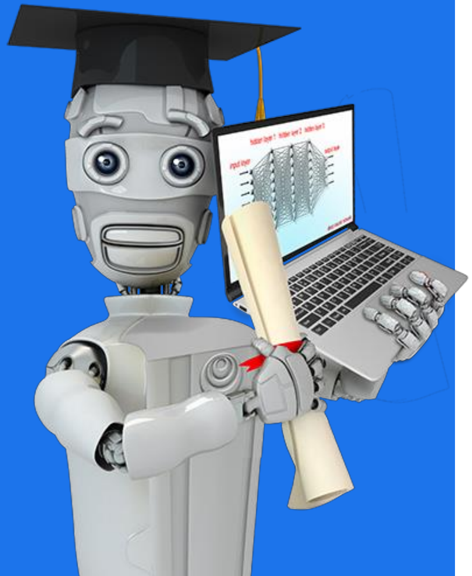
Regression

Predict a number

infinitely many possible outputs

Stanford
ONLINE

DeepLearning.AI

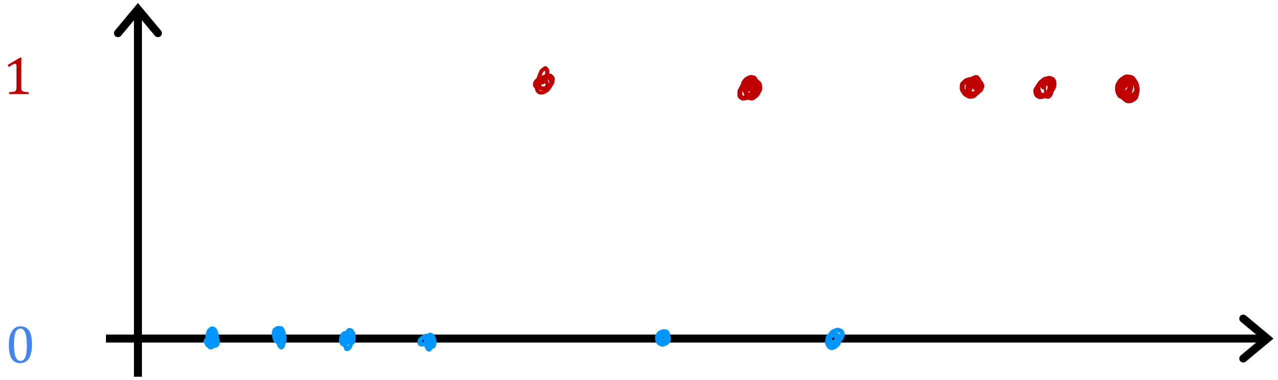


Machine Learning Overview

Supervised Learning Part 2

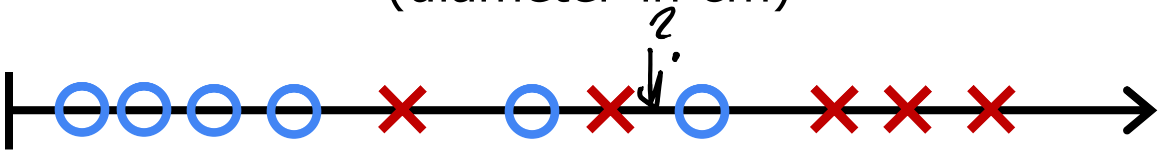
Classification: Breast cancer detection

malignant benign



tumor size x
(diameter in cm)

size	diagnosis
2	0
5	1
1	0
7	1
⋮	



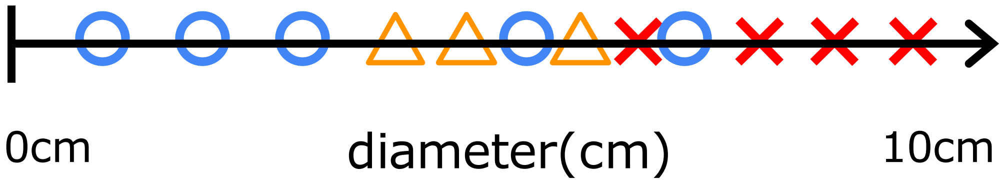
○ benign
× malignant

Classification: Breast cancer detection

○ benign

× malignant type1

△ malignant type 2



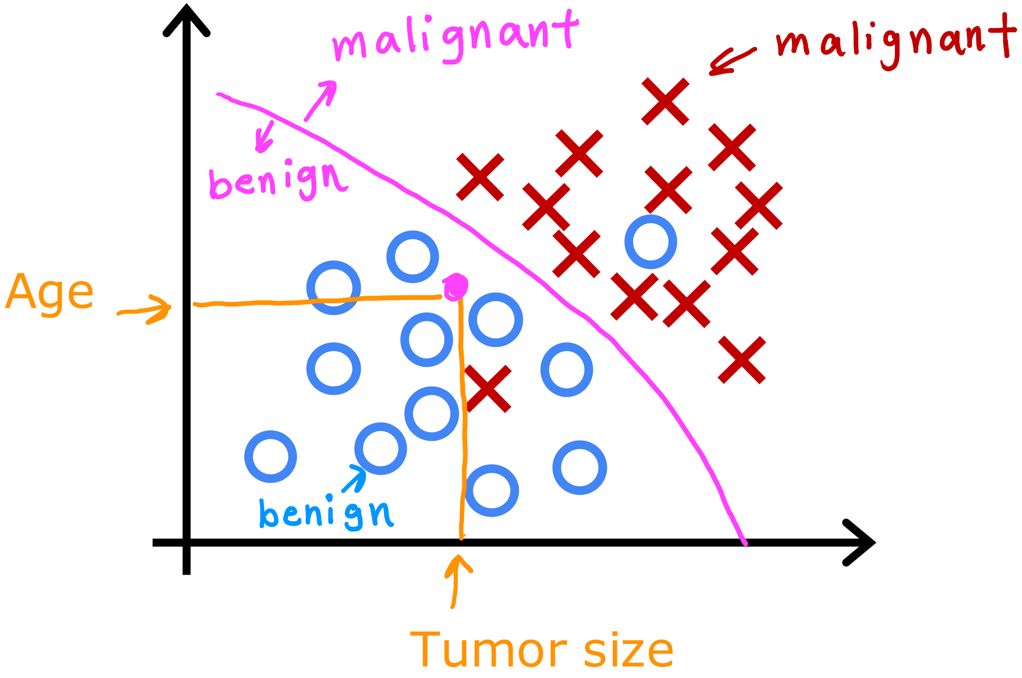
class category

Classification

predict categories cat dog benign malignant 0, 1, 2

small number of possible outputs

Two or more inputs



Supervised learning

Learns from being given "right answers"

Regression

Predict a number

infinitely many possible outputs

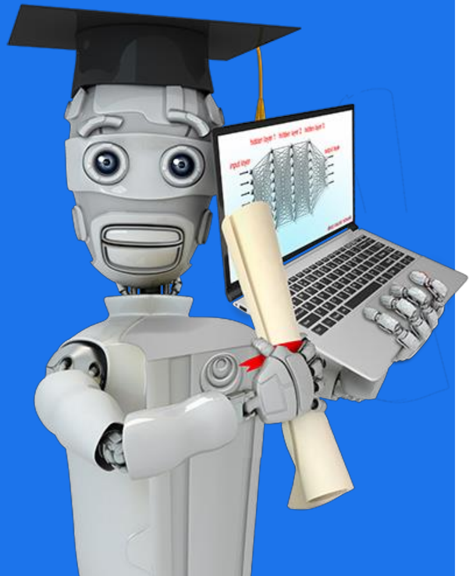
Classification

predict categories

small number of possible outputs

Stanford
ONLINE

DeepLearning.AI

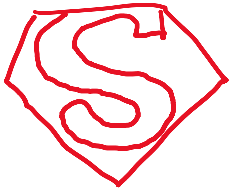


Machine Learning Overview

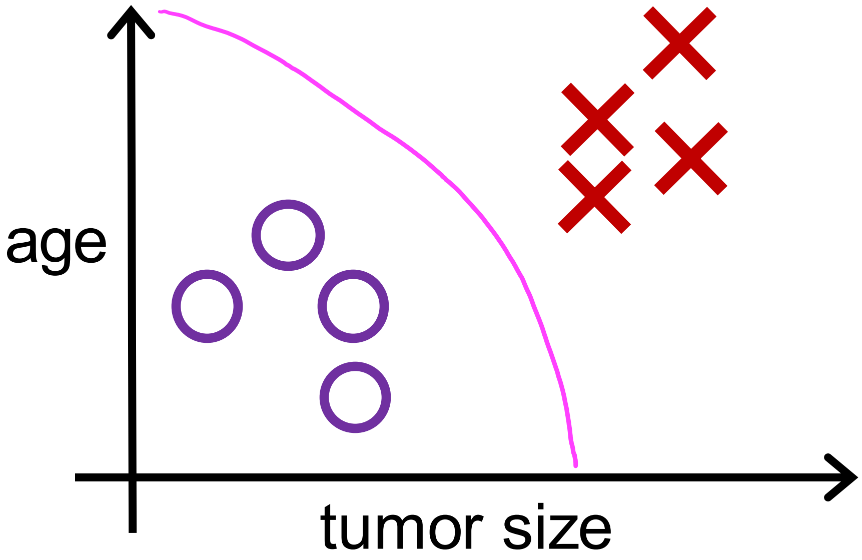
Unsupervised Learning Part 1

Previous: Supervised learning

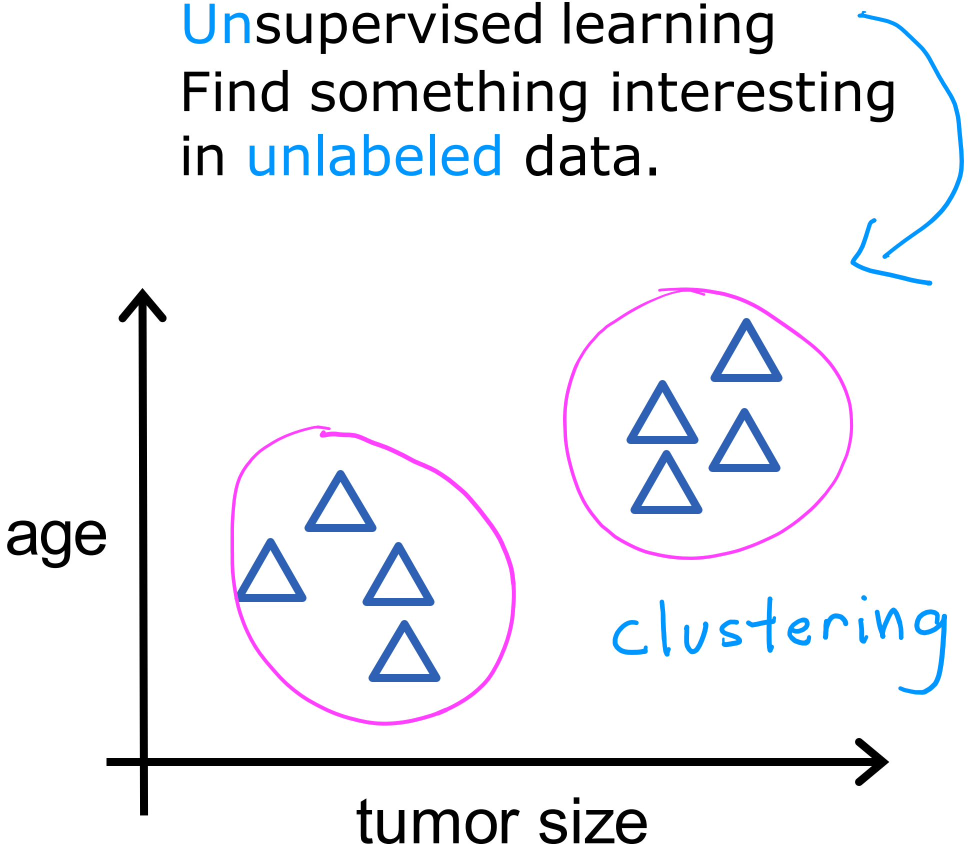
Now: Unsupervised learning



Supervised learning
Learn from data labeled
with the "right answers"

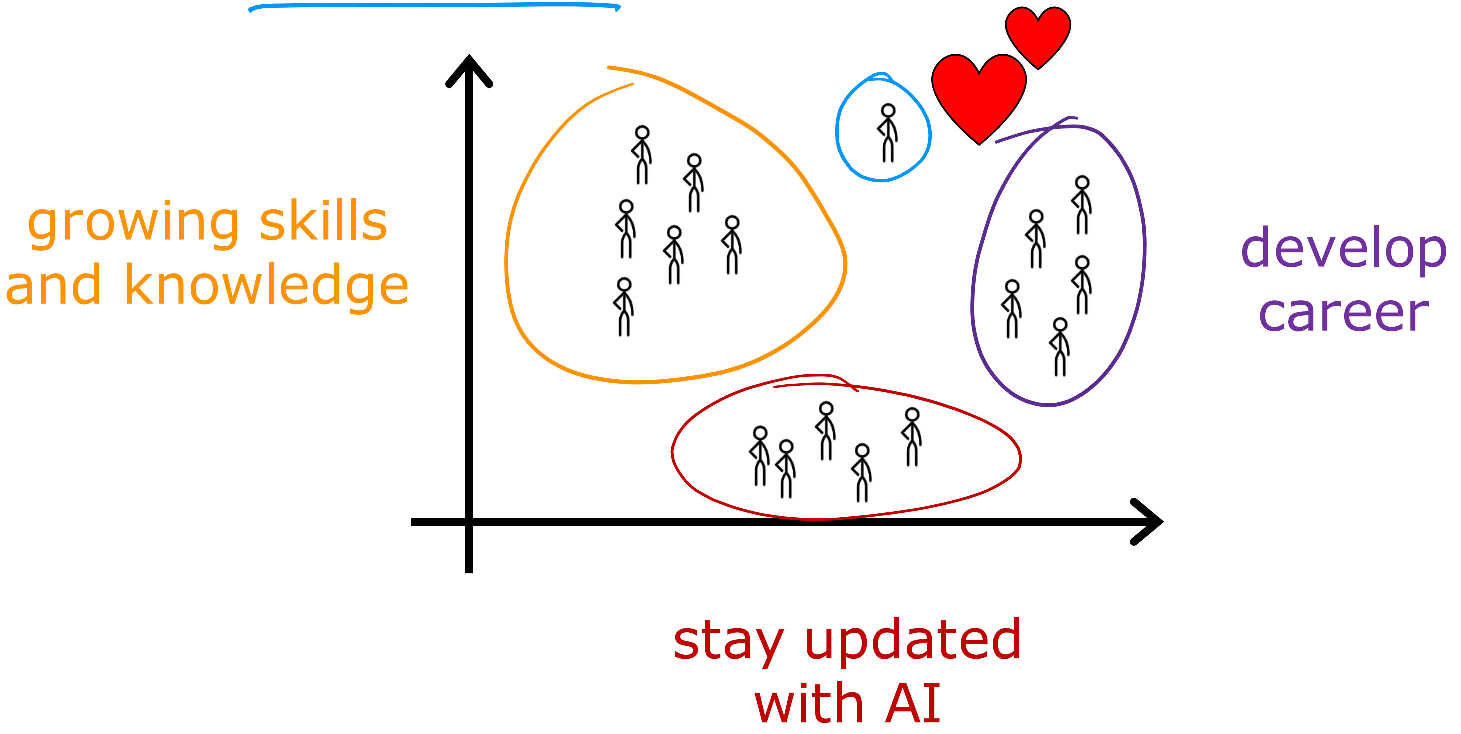


Unsupervised learning
Find something interesting
in unlabeled data.



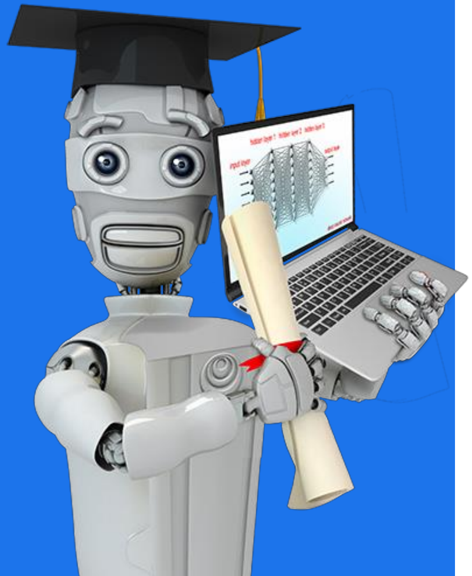
Clustering: Google news

Clustering: Grouping customers



Stanford ONLINE

DeepLearning.AI



Machine Learning Overview

Unsupervised Learning Part 2

Unsupervised learning

Data only comes with inputs x, but not output labels y.
Algorithm has to find structure in the data.

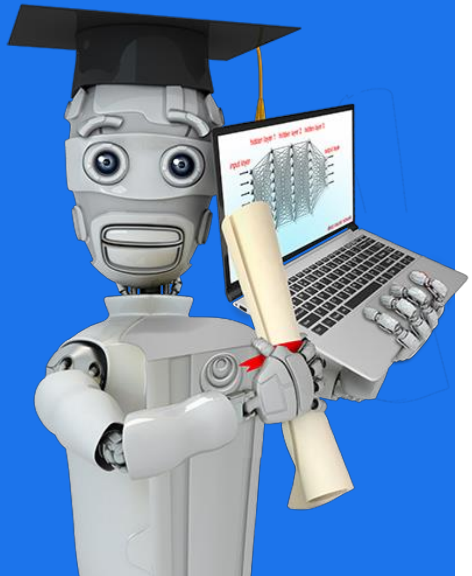
<u>Clustering</u>
Group similar data points together.

<u>Dimensionality reduction</u>
Compress data using fewer numbers.

<u>Anomaly detection</u>
Find unusual data points.

Stanford
ONLINE

DeepLearning.AI



Linear Regression with One Variable

Linear Regression Model Part 1

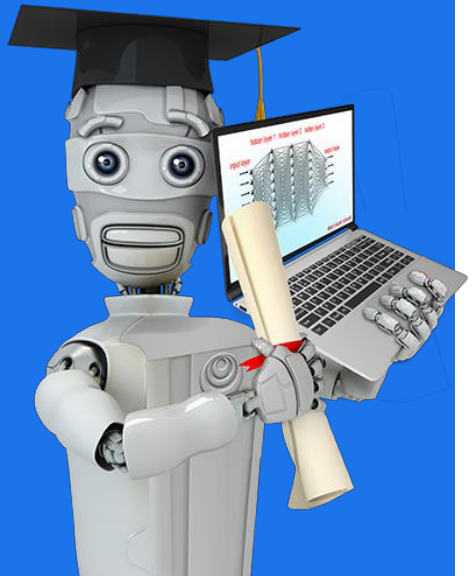
House sizes and prices

Data table

size in feet²	price in $1000's
2104	400
1416	232
1534	315
852	178
...	...
3210	870

Stanford
ONLINE

DeepLearning.AI



Linear Regression with One Variable

Linear Regression Model Part 2

training set

features
targets

↓

learning algorithm

↓

$x \rightarrow \boxed{f} \rightarrow \hat{y}$

"y-hat"

~~hypothesis~~ function

feature   model   prediction

(estimated y)

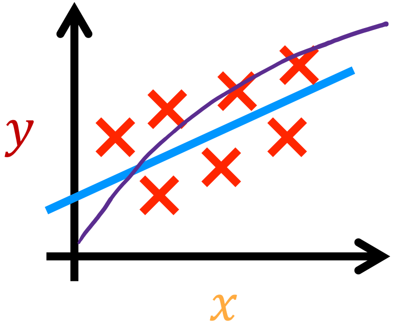
target

$\text{size} \rightarrow \boxed{f} \rightarrow \text{price}$ (estimated)

How to represent f?

$$f_{w,b}(x) = wx + b$$

$$f(x)$$



$f_{w,b}(x) = wx + b$

$f(x) = wx + b$

linear

single feature x

Linear regression with one variable.

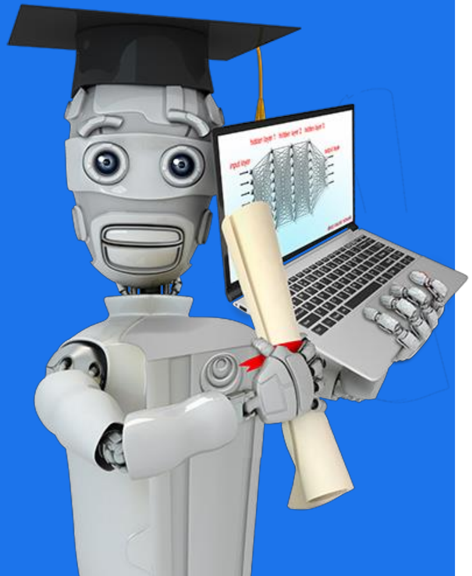
size

Univariate linear regression.

one variable

Stanford
ONLINE

DeepLearning.AI



Linear Regression with One Variable

Cost Function

Training set

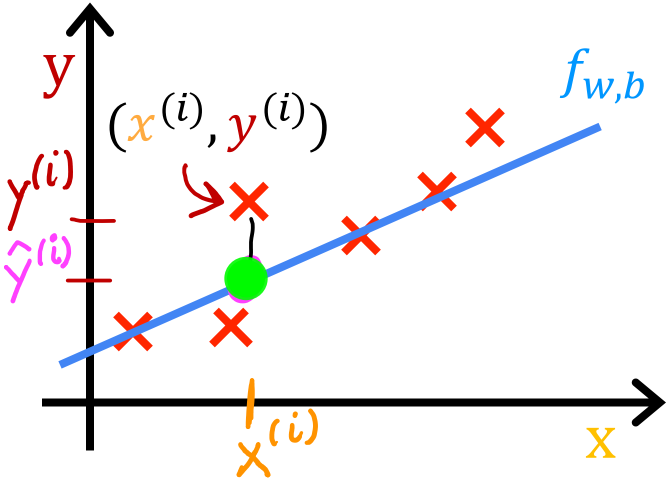
features size in feet^2 (x)	targets price \$1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

Model: $f_{w,b}(x) = wx + b$

w, b: parameters

coefficients
weights

What do w, b do?



$\hat{y}^{(i)} = f_{w,b}(x^{(i)})$

$f_{w,b}(x^{(i)}) = wx^{(i)} + b$

Cost function: Squared error cost function

$$J(w,b) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)})^2$$

error

m = number of training examples

$$J(w,b) = \frac{1}{2m} \sum_{i=1}^{m} (f_{w,b}(x^{(i)}) - y^{(i)})^2$$

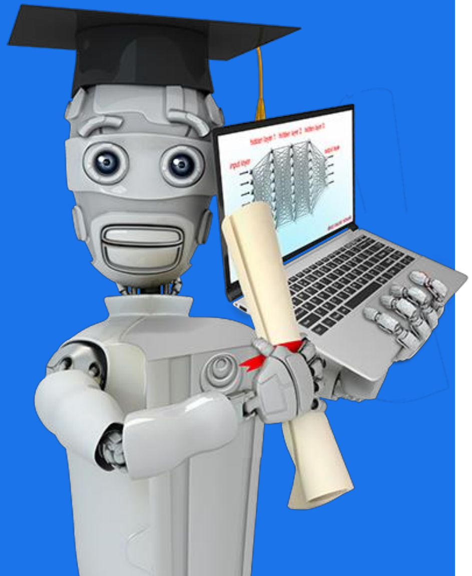
intuition (next!)

Find w, b:

$\hat{y}^{(i)}$ is close to $y^{(i)}$ for all $(x^{(i)}, y^{(i)})$.

Stanford
ONLINE

DeepLearning.AI



Training Linear Regression

Gradient Descent

Have some function $J(w,b)$ for linear regression or any function

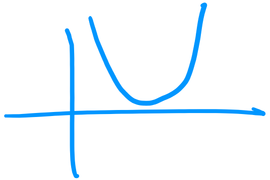
Want $\min\limits_{w,b} J(w,b)$ $\qquad \min\limits_{w_1,\dots,w_n,b} J(w_1,w_2,\dots,w_n,b)$

Outline:

- Start with some w,b (set $w=0$, $b=0$)
- Keep changing w,b to reduce $J(w,b)$
- Until we settle at or near a minimum

may have >1 minimum

J not always

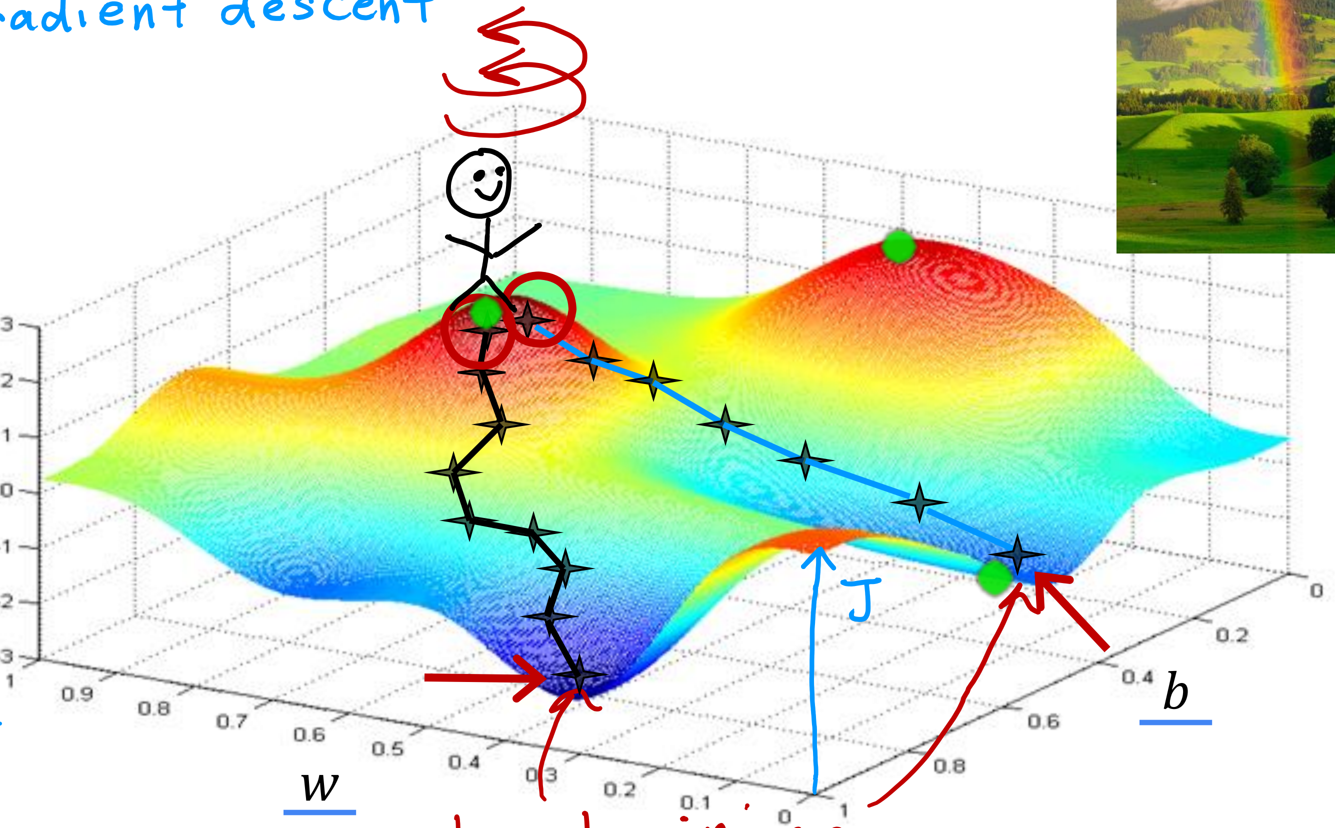


gradient descent

$J(w,b)$

not squared error cost

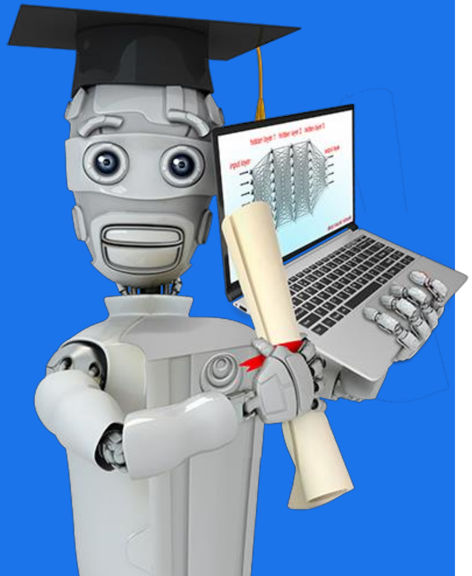
not linear regression



local minima

Stanford
ONLINE

DeepLearning.AI



Training Linear Regression

Gradient Descent Intuition

Gradient descent algorithm

repeat until convergence {

learning rate

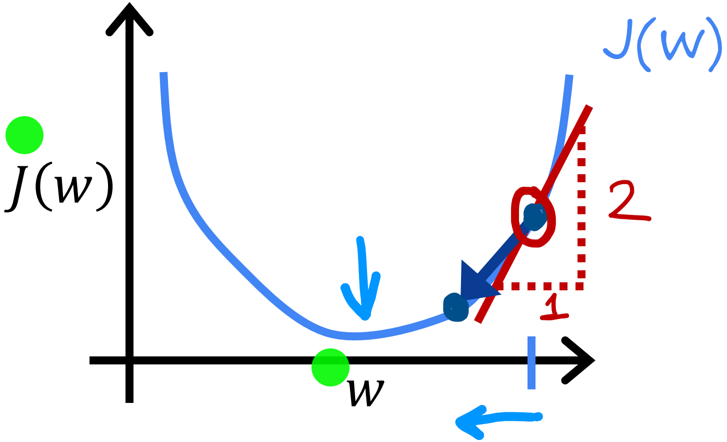
$$\begin{cases} \underline{w} = w - \alpha \boxed{\frac{\partial}{\partial w} J(w,b)} \\ \underline{b} = b - \alpha \frac{\partial}{\partial b} J(w,b) \end{cases}$$

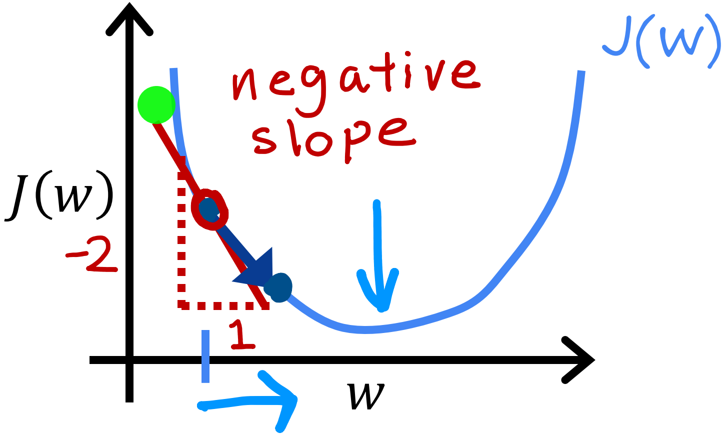
derivative

$J(w)$

$$w = w - \alpha \frac{\partial}{\partial w} J(w)$$

$$\underline{\min_{w}} J(w)$$





$$w = w - \alpha \boxed{\frac{d}{dw}J(w)}$$

>0

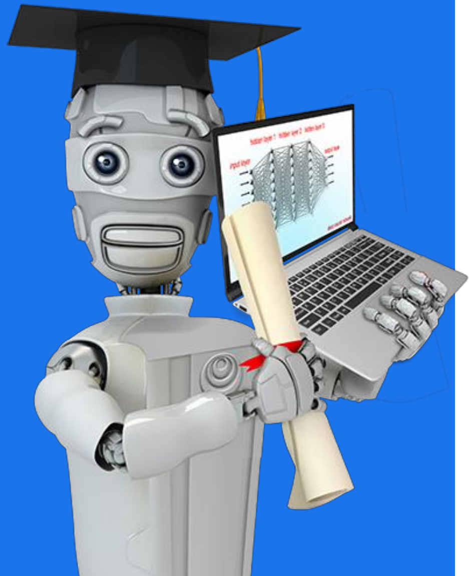
$$w = w - \alpha \cdot (positive\ number)$$

$$\frac{d}{dw}J(w) \quad < 0$$

$$w = w - \alpha \cdot (negative\ number)$$

Stanford
ONLINE

DeepLearning.AI



Training Linear Regression

Learning Rate

$$w = w - \alpha \frac{d}{dw} J(w)$$

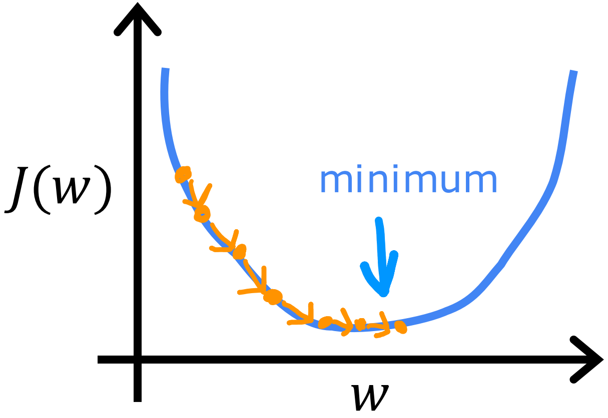
If α is too small...

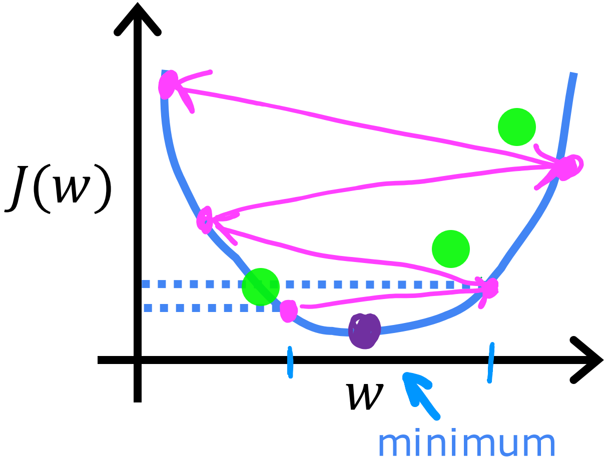
Gradient descent may be slow.

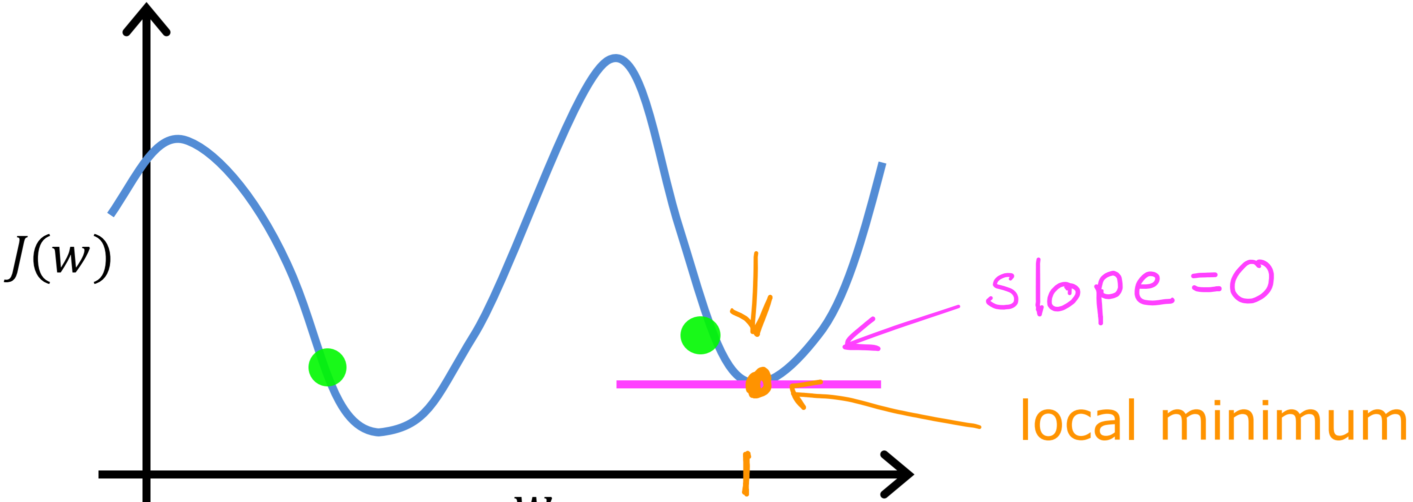
If α is too large...

Gradient descent may:

- Overshoot, never reach minimum
- Fail to converge, diverge







current value of w → 5

$$w = w - \underset{0.1}{\alpha} \underbrace{\boxed{\frac{d}{dw}J(w)}}_{=0}$$

5 − 0.1 · 0

$$w = w - \alpha \cdot 0$$

5 5

$$w = w$$

Can reach local minimum with fixed learning rate

α

$$w = w - \underbrace{\boxed{\alpha \frac{d}{dw} J(w)}}$$

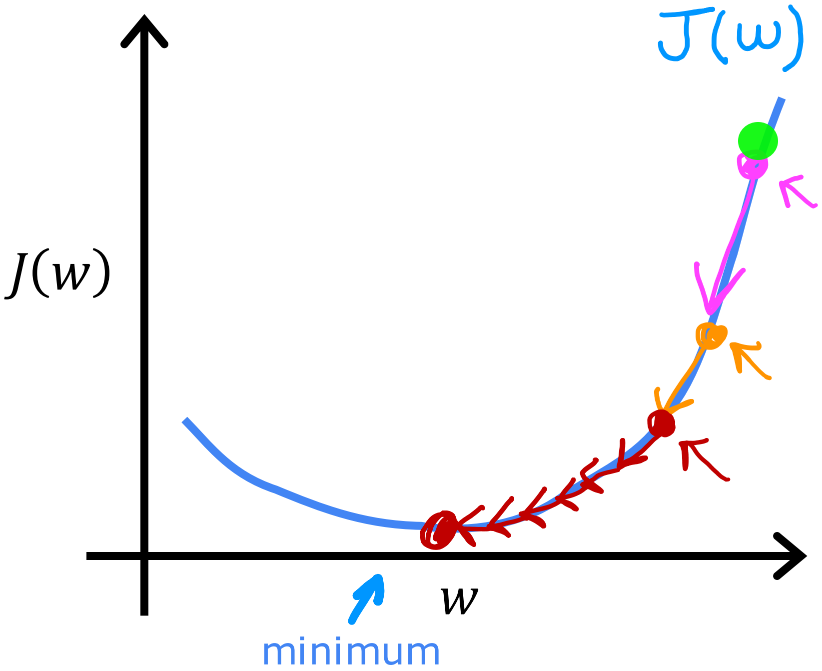
smaller

not as large

large

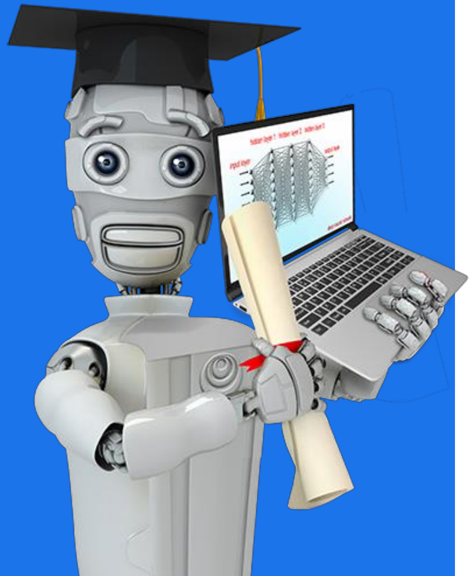
Near a local minimum,
- Derivative becomes smaller
- Update steps become smaller

Can reach minimum without decreasing learning rate α



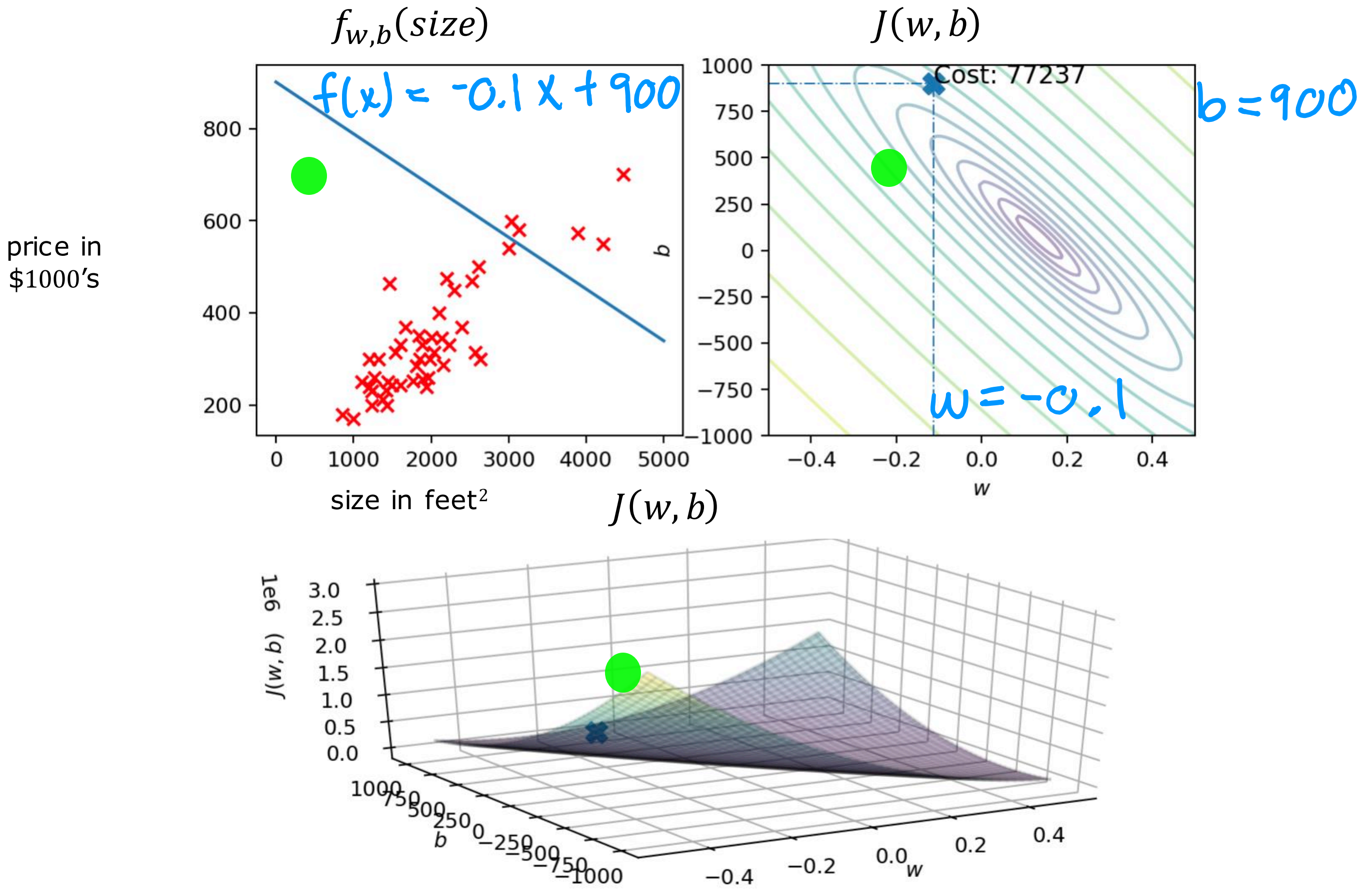
Stanford ONLINE

DeepLearning.AI



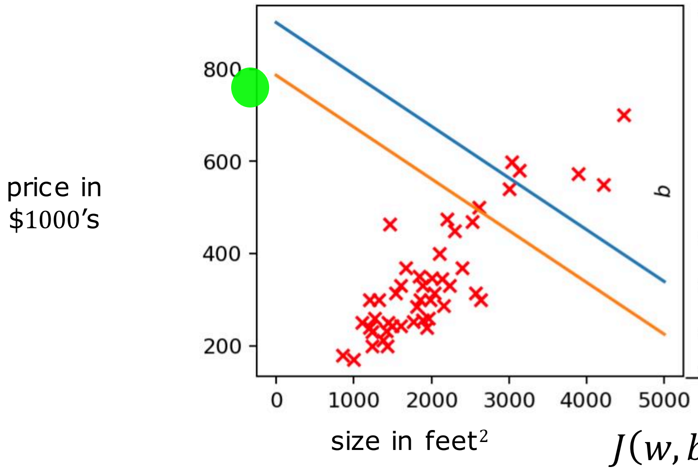
Training Linear Regression

Running Gradient Descent

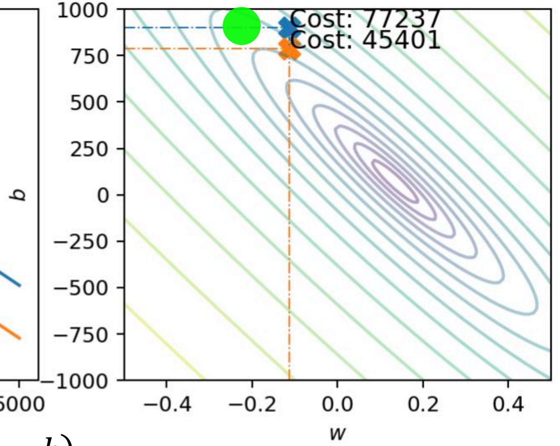


$f_{w,b}(size)$
$J(w,b)$
$f(x) = -0.1x + 900$
Cost: 77237
$b = 900$
$w = -0.1$
price in $1000's
size in feet^2
b
w
$J(w,b)$
1e6
$J(w,b)$

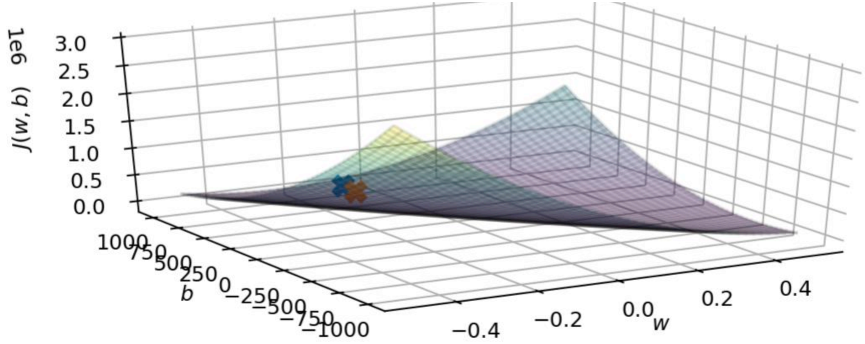
$f_{w,b}(size)$

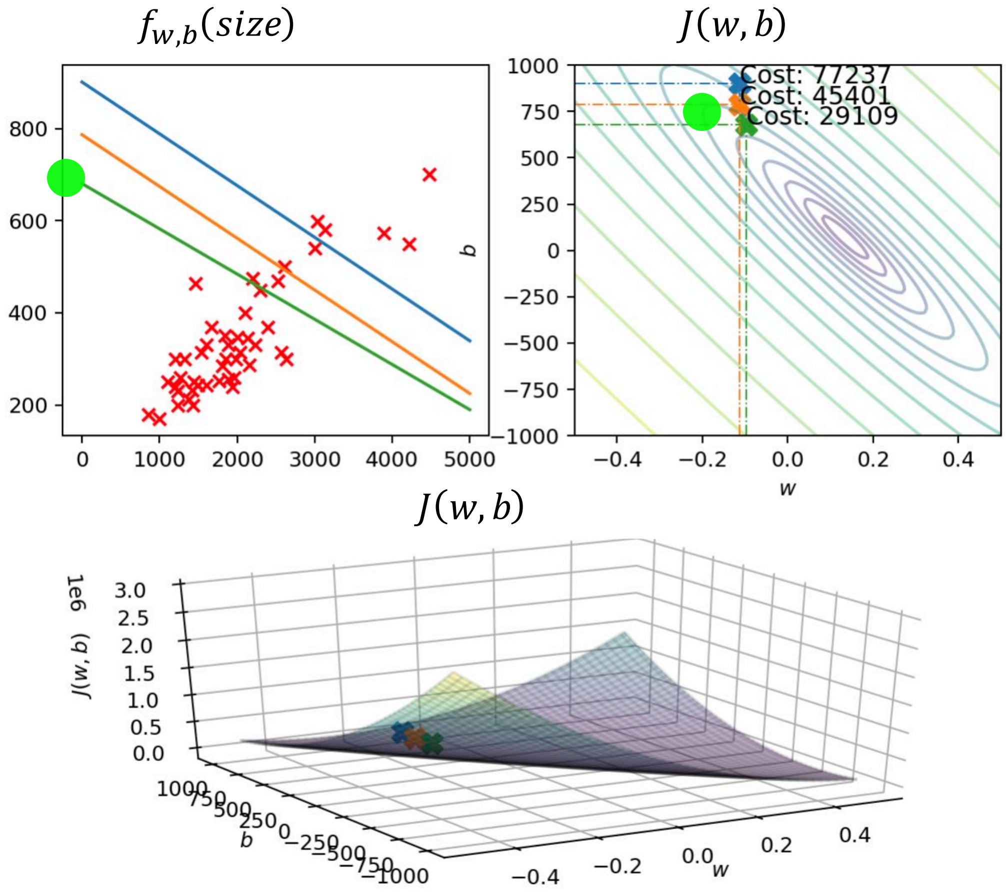


$J(w,b)$

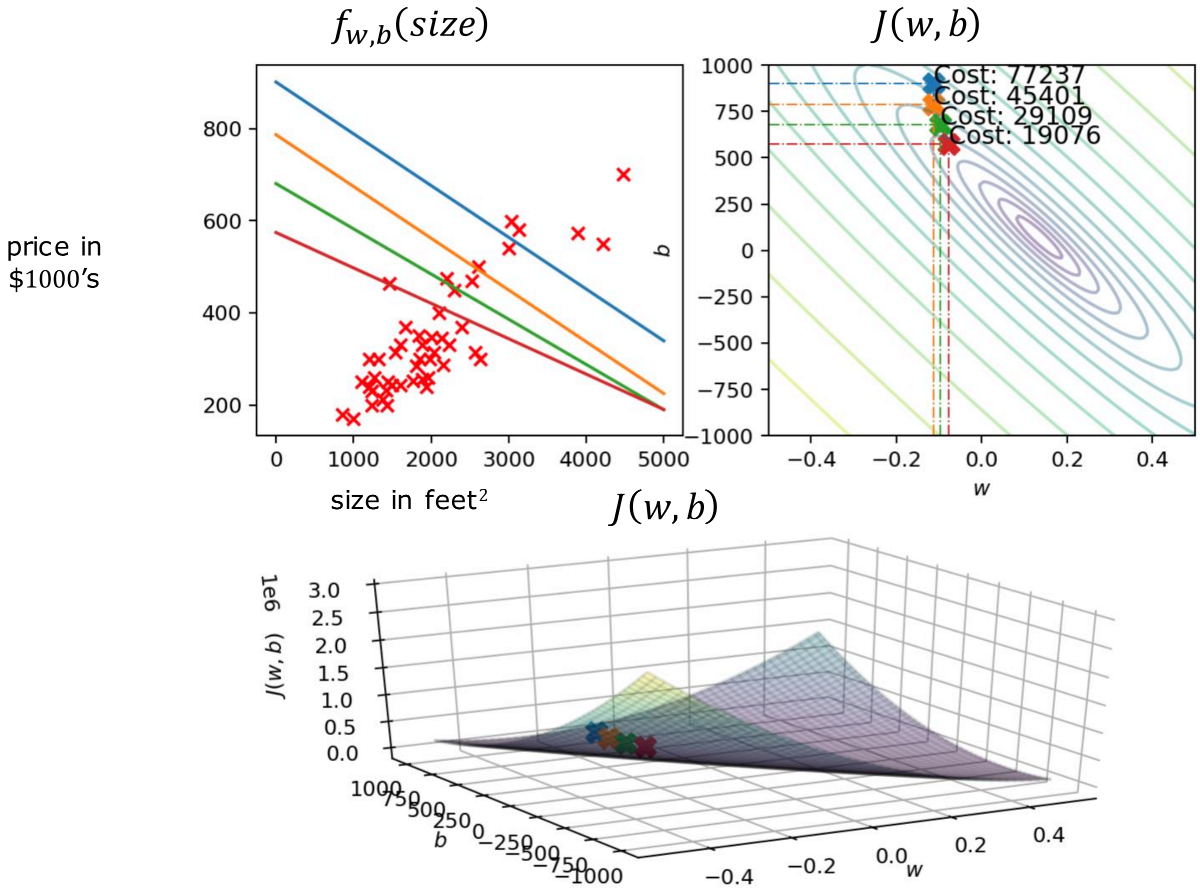


$J(w,b)$

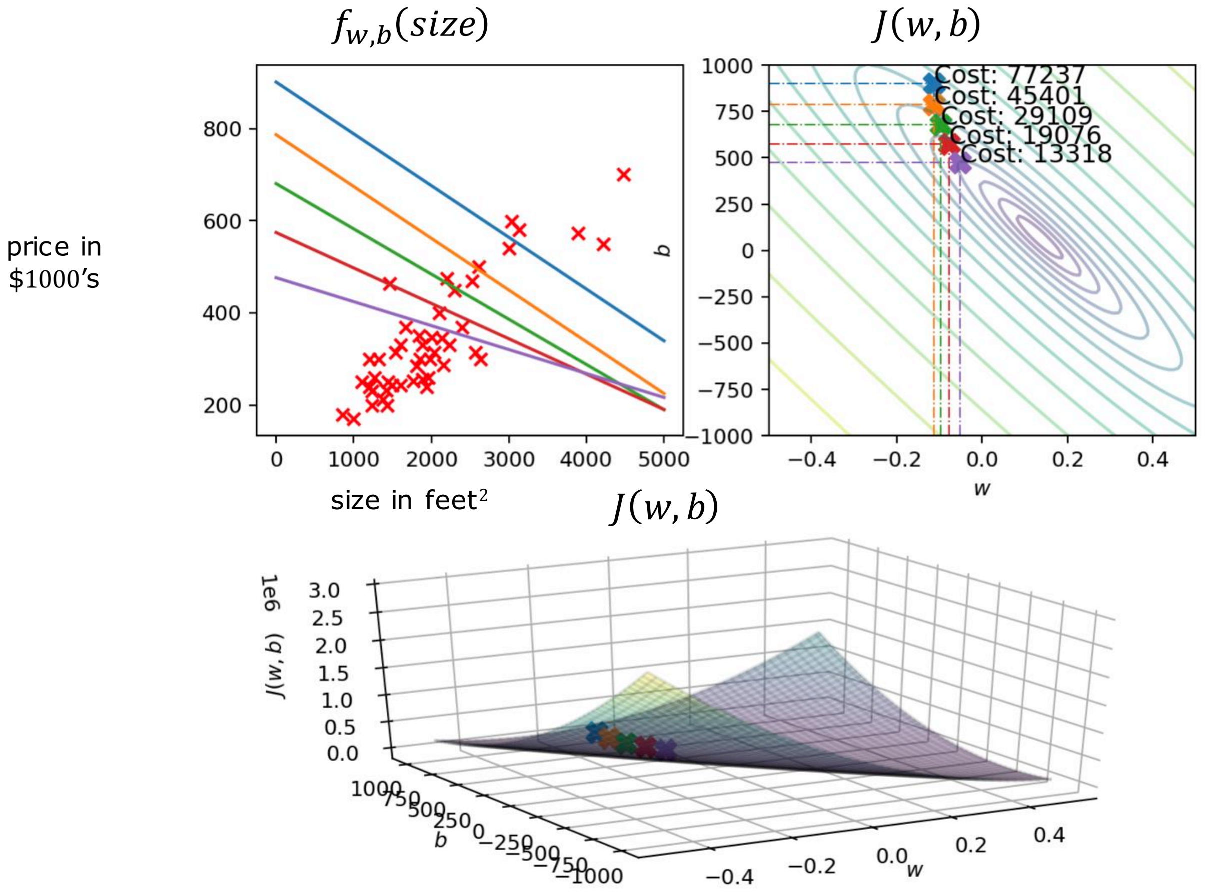




$f_{w,b}(size)$
$J(w,b)$
Cost: 77237
Cost: 45401
Cost: 29109
b
w
$J(w,b)$
1e6
J(w, b)
b
w

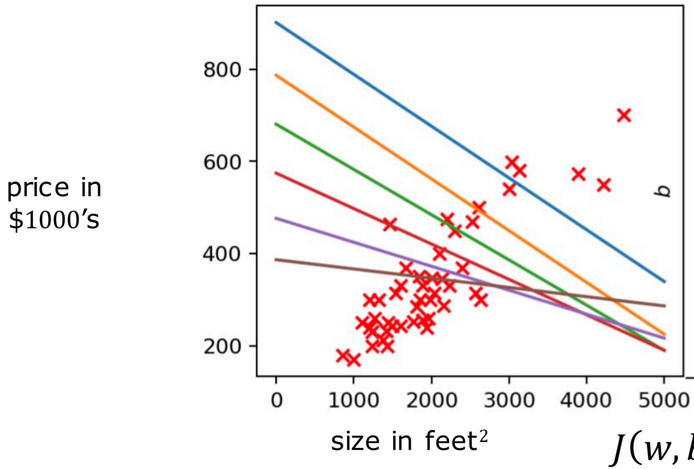


$f_{w,b}(size)$
$J(w,b)$
price in
$1000's
size in feet^2
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
b
w
$J(w,b)$
1e6
J(w, b)

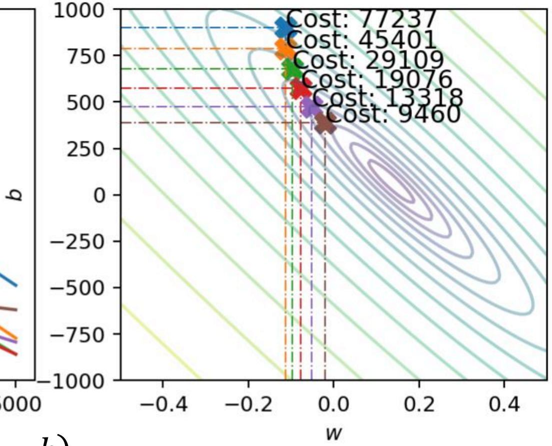


$f_{w,b}(size)$
price in $1000's
size in feet^2
$J(w,b)$
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
Cost: 13318
b
w
$J(w,b)$
1e6
J(w, b)

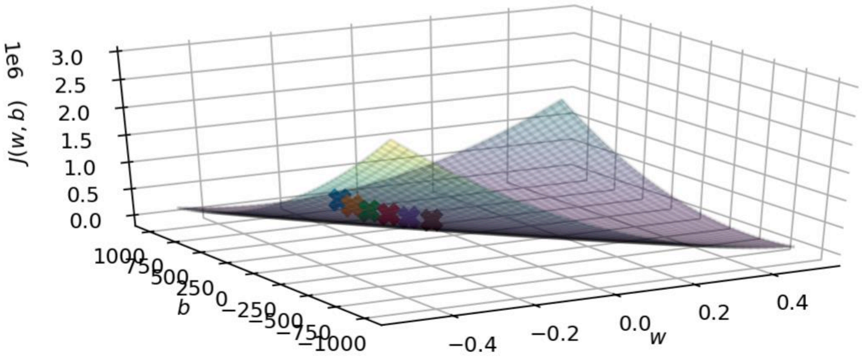
$f_{w,b}(size)$

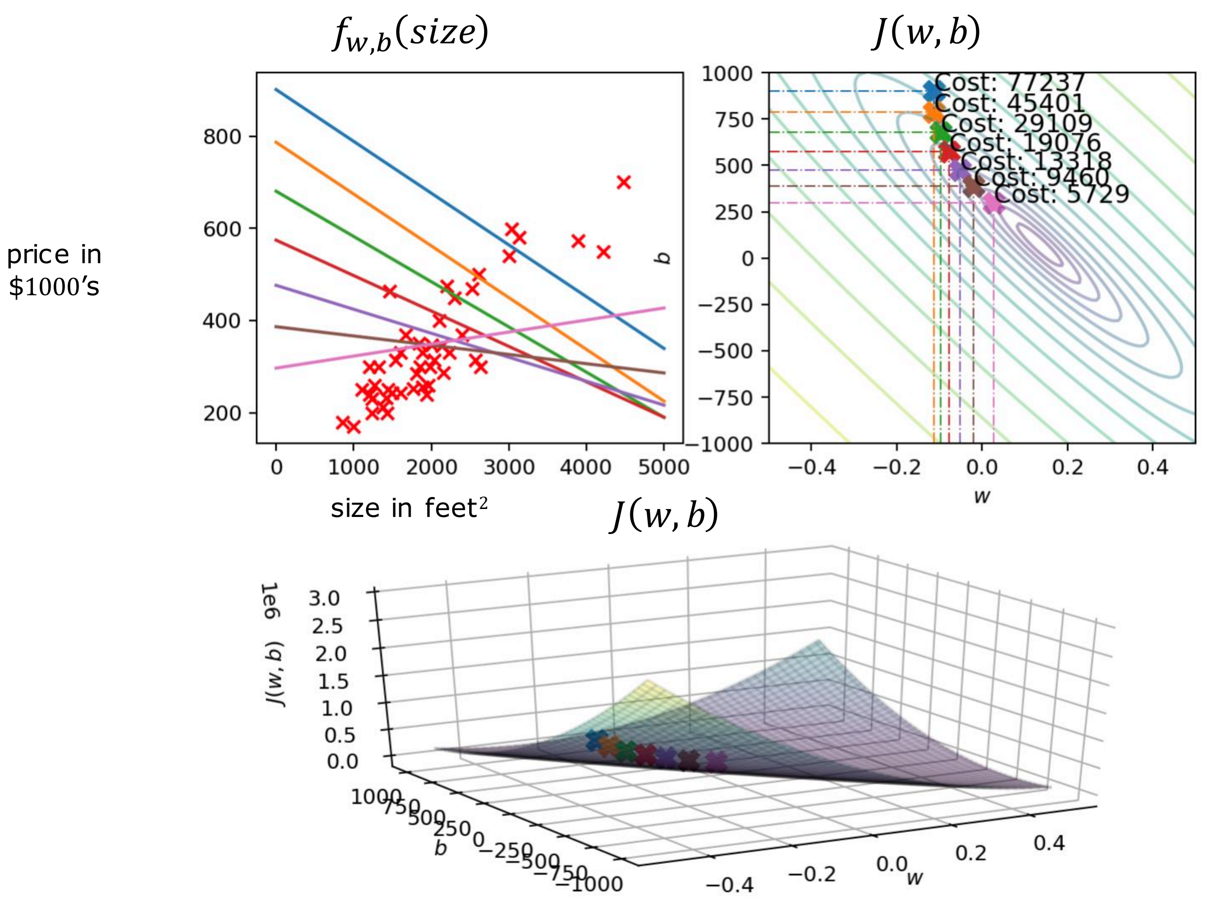


$J(w,b)$

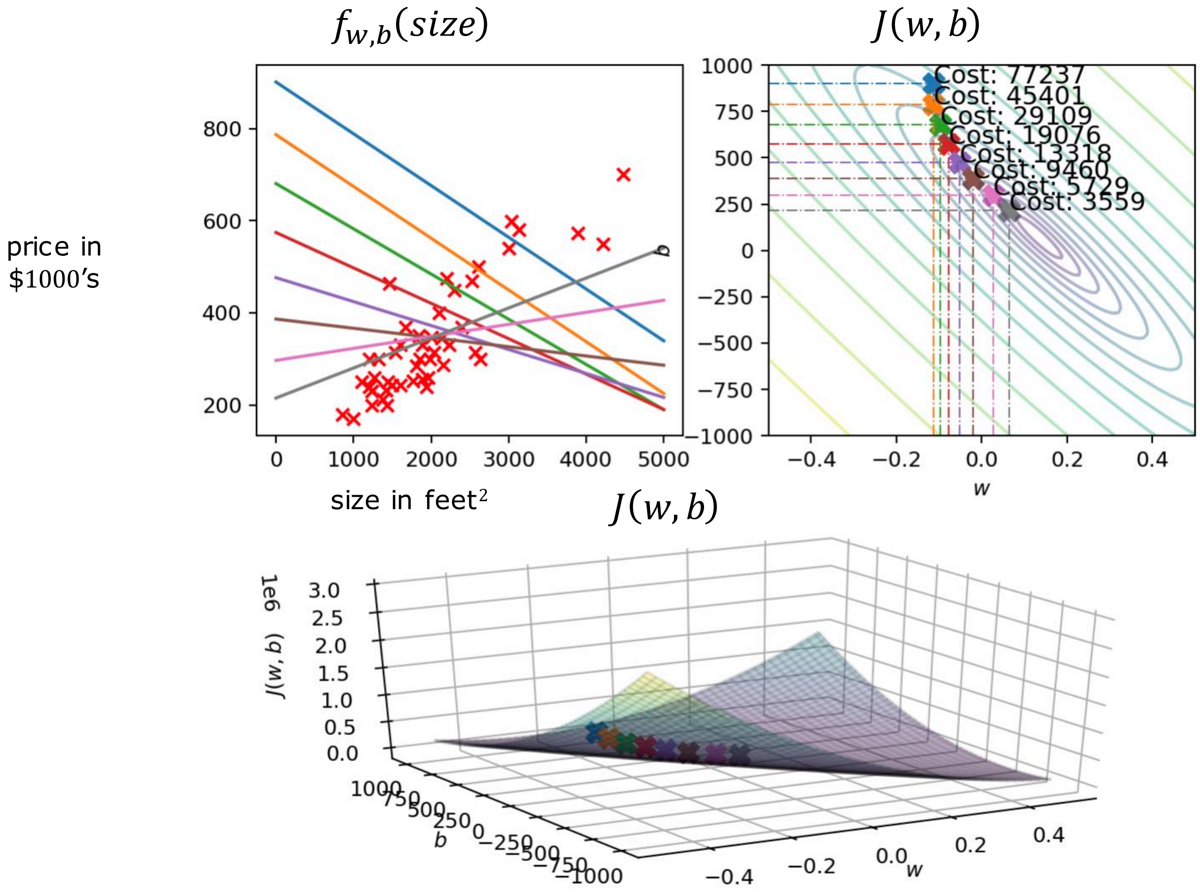


$J(w,b)$

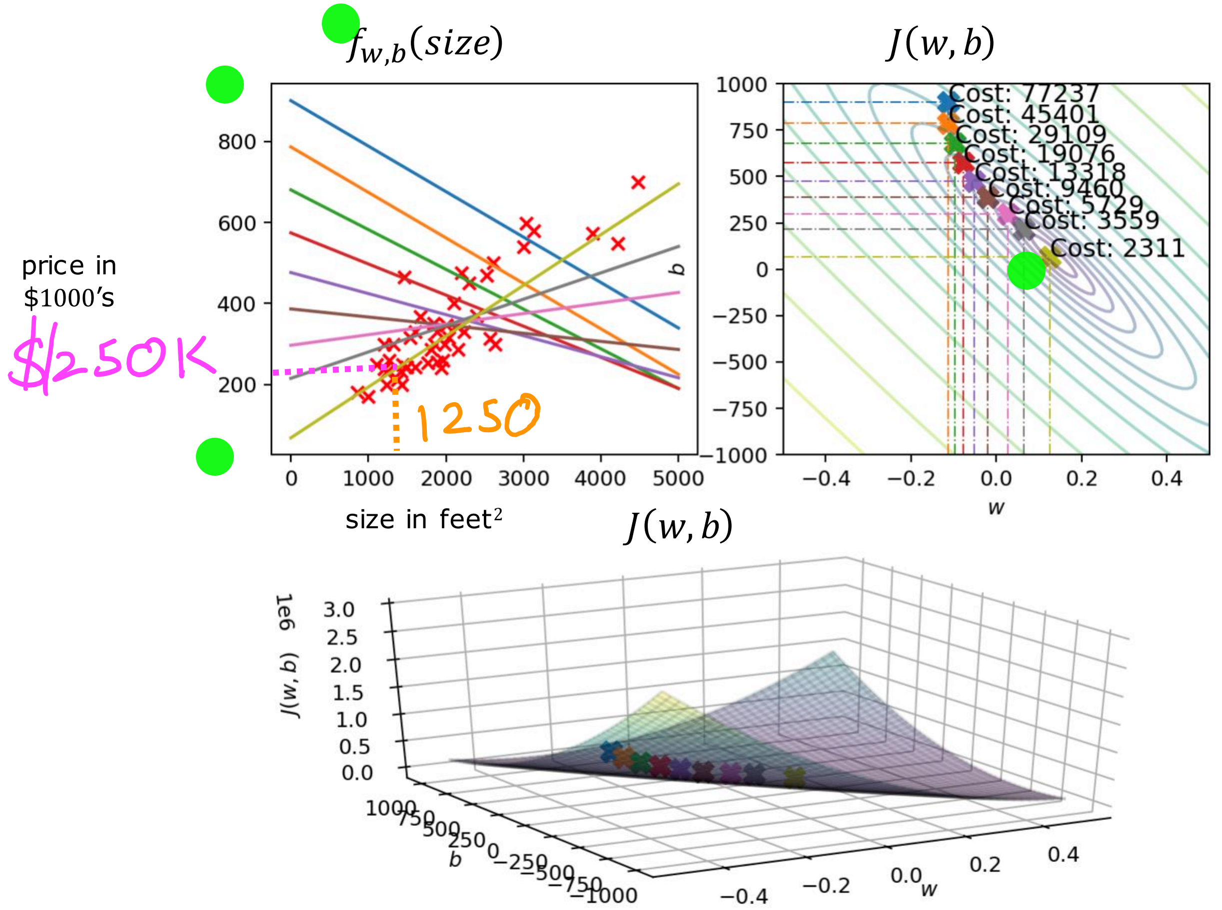




$f_{w,b}(size)$
price in
$1000's
size in feet^2
$J(w,b)$
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
Cost: 13318
Cost: 9460
Cost: 5729
$J(w,b)$



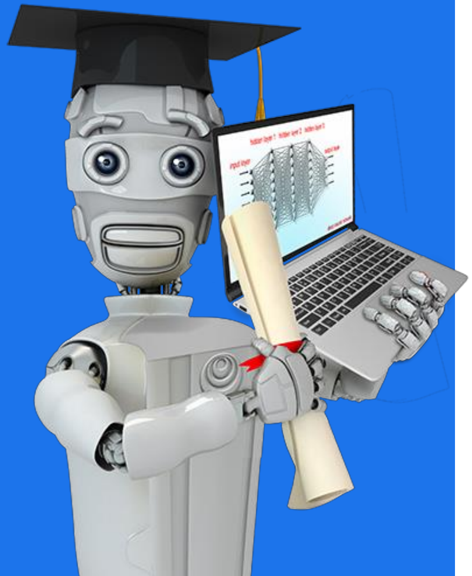
$f_{w,b}(size)$
$J(w,b)$
price in
$1000's
size in feet^2
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
Cost: 13318
Cost: 9460
Cost: 5729
Cost: 3559
w
$J(w,b)$
1e6
$J(w,b)$
b
w



$f_{w,b}(size)$
$J(w,b)$
price in
$1000's
$250K
1250
size in feet^2
Cost: 77237
Cost: 45401
Cost: 29109
Cost: 19076
Cost: 13318
Cost: 9460
Cost: 5729
Cost: 3559
Cost: 2311
b
w
$J(w,b)$
1e6
$J(w,b)$

Stanford ONLINE

DeepLearning.AI



Linear Regression with Multiple Variables

Multiple Features

Multiple features (variables)

one feature

Size in feet^2 (x)	Price (\$) in 1000's (y)
2104	400
1416	232
1534	315
852	178
…	…

$$f_{w,b}(x) = wx + b$$

Multiple features (variables)

Size in feet² x_1	Number of bedrooms x_2	Number of floors x_3	Age of home in years x_4	Price ($) in $1000's
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

$j = 1 \ldots 4$

$n = 4$

$i = 2$

x_j = j^{th} feature

n = number of features

$\vec{x}^{(i)}$ = features of i^{th} training example

$x_j^{(i)}$ = value of feature j in i^{th} training example

$\vec{x}^{(2)} = [1416 \ 3 \ 2 \ 40]$

$x_3^{(2)} = 2$

Model:

Previously: $f_{w,b}(x) = wx + b$

$f_{w,b}(X) = w_1 X_1 + w_2 X_2 + w_3 X_3 + w_4 X_4 + b$

example

$f_{w,b}(X) = 0.1 X_1 + 4 X_2 + 10 X_3 + -2 X_4 + 80$

size #bedrooms #floors years base price

$f_{w,b}(\mathrm{x}) = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$

$$f_{\vec{w},b}(\vec{x}) = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$$

$\vec{w} = [w_1 \ w_2 \ w_3 \ \ldots \ w_n]$ parameters of the model

b is a number

vector $\vec{x} = [x_1 \ x_2 \ x_3 \ \ldots \ x_n]$

$$f_{\vec{w},b}(\vec{x}) = \vec{w} \cdot \vec{x} + b = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n + b$$

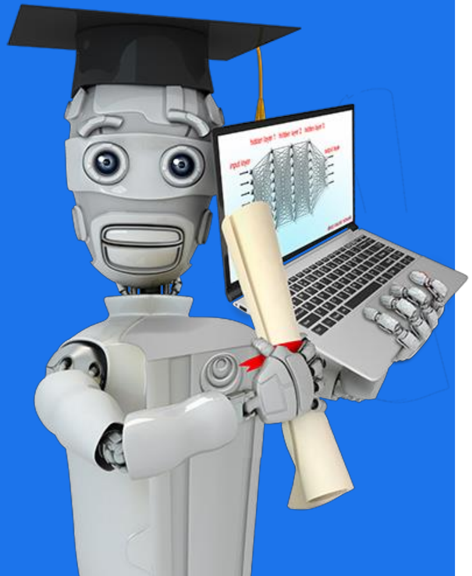
dot product

multiple linear regression

(not multivariate regression)

Stanford ONLINE

DeepLearning.AI



Classification

Motivations

Classification

Question	Answer "y"	
Is this email spam?	no	yes
Is the transaction fraudulent?	no	yes
Is the tumor malignant?	no	yes

y can only be one of two values

"binary classification"

class = category

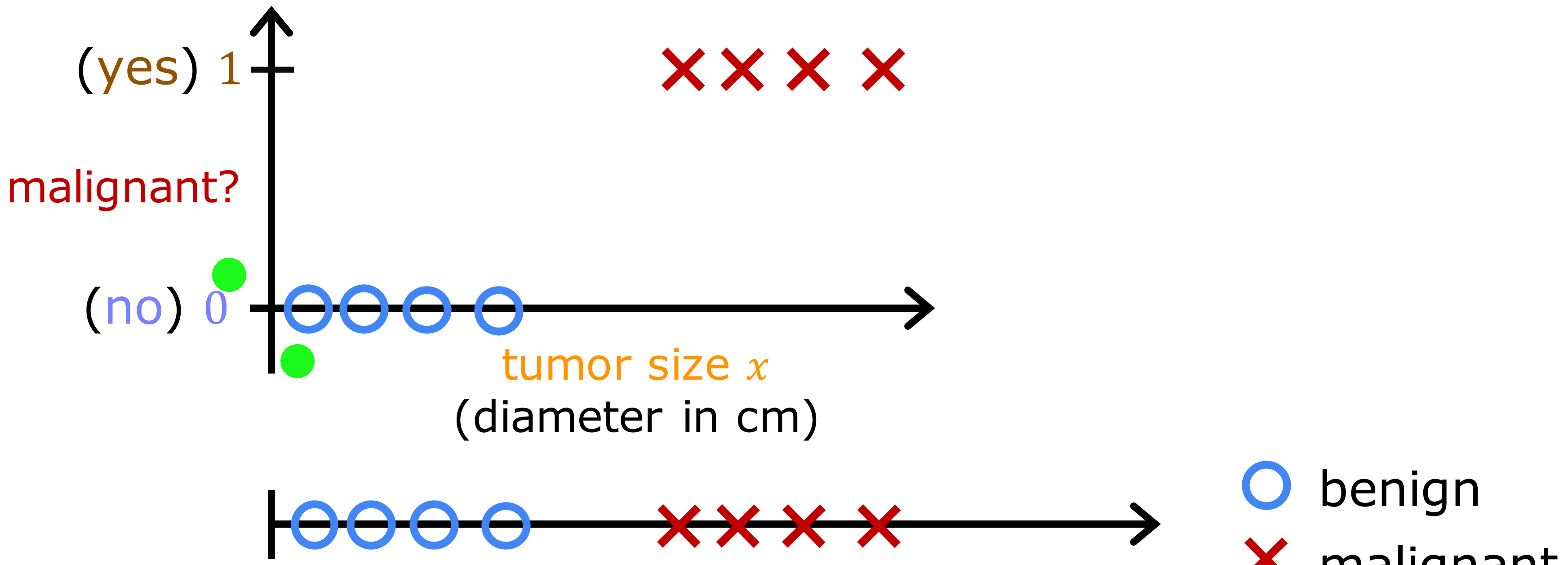
false true

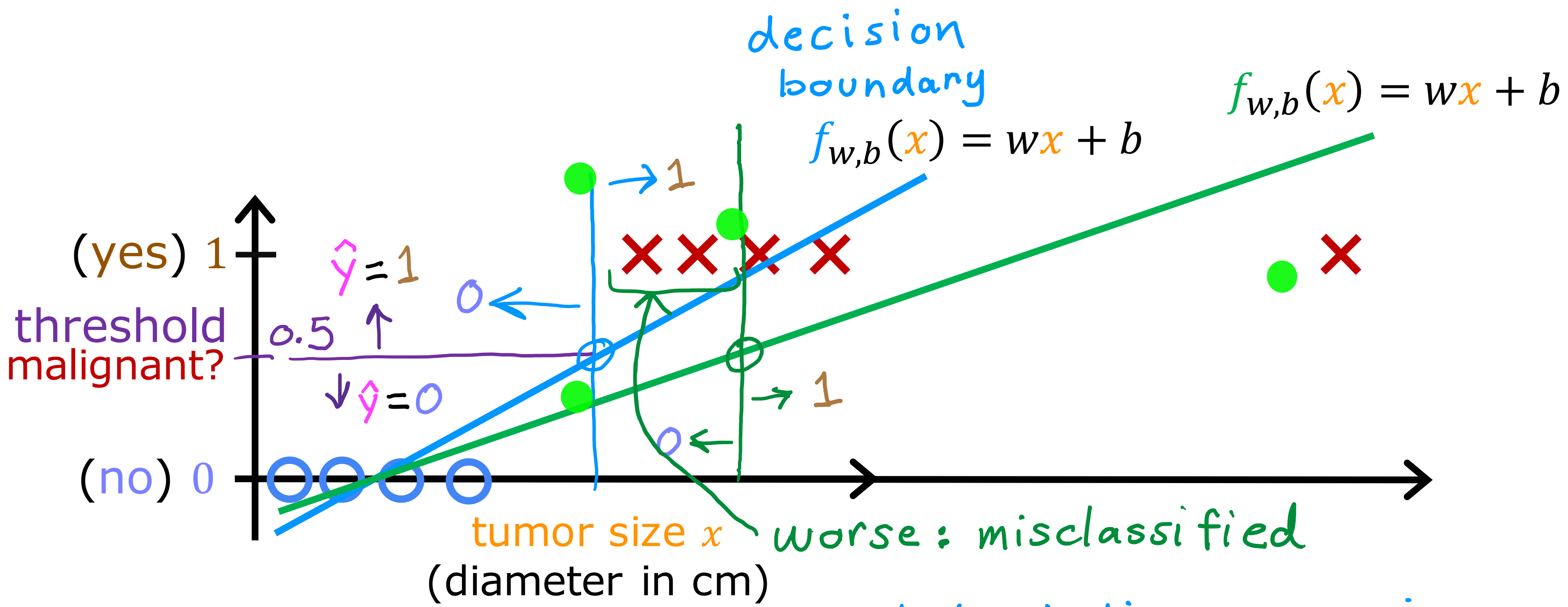
0 1

useful for classification

"negative class" ≠ "bad" absence

"positive class" ≠ "good" presence


(yes) 1
malignant?
(no) 0
tumor size x
(diameter in cm)
benign
malignant



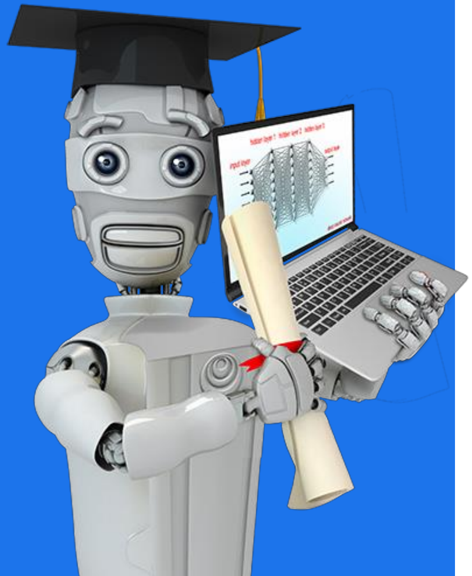
if $f_{w,b}(x) < 0.5 \rightarrow \hat{y} = 0$

if $f_{w,b}(x) \geq 0.5 \rightarrow \hat{y} = 1$

next: $\underbrace{\text{logistic regression}}_{\text{classification}}$

Stanford ONLINE

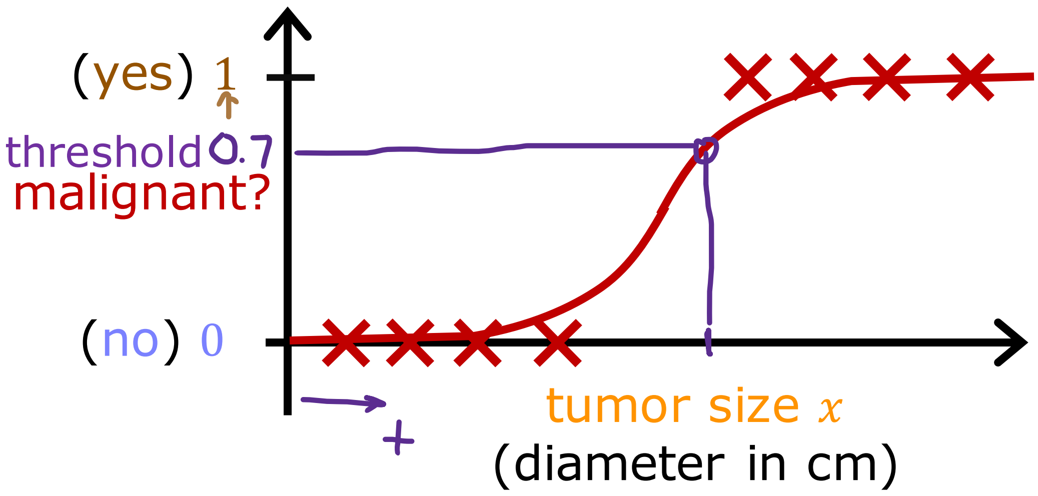
DeepLearning.AI

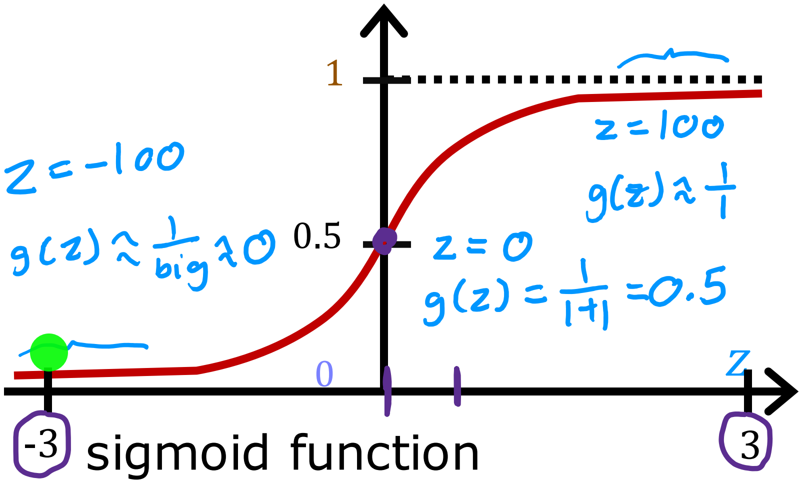


Classification

Logistic Regression

Want outputs between 0 and 1





sigmoid function

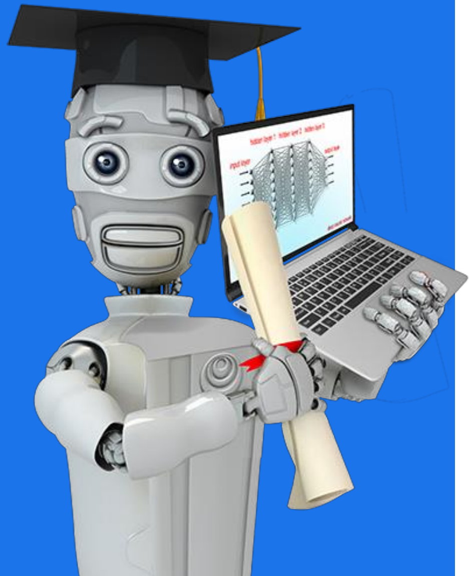
logistic function

outputs between 0 and 1

$$g(z) = \frac{1}{1+e^{-z}} \qquad 0 < g(z) < 1$$

Stanford ONLINE

DeepLearning.AI



Cost Function

Simplified Cost Function for Logistic Regression

Simplified cost function

loss

$$L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = -\underbrace{y^{(i)}\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) - \left(1-y^{(i)}\right)\log\left(1-f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right)}$$

cost

$$J(\vec{w},b) = \frac{1}{m}\sum_{i=1}^{m}\left[\underbrace{L\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right), y^{(i)}\right)}\right]$$

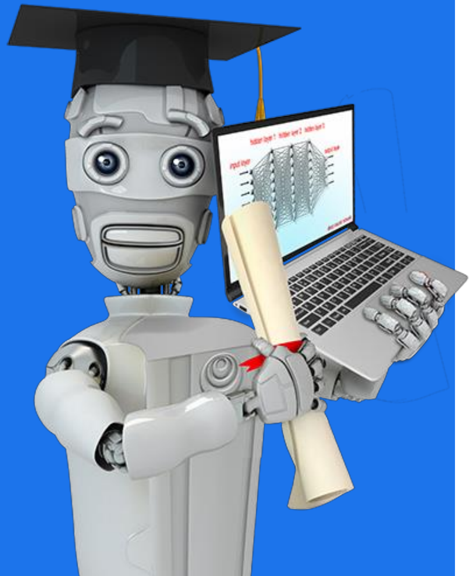
convex
(single global minimum)

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right) + \left(1-y^{(i)}\right)\log\left(1-f_{\vec{w},b}\left(\vec{x}^{(i)}\right)\right)\right]$$

maximum likelihood
(don't worry about it!)

Stanford
ONLINE

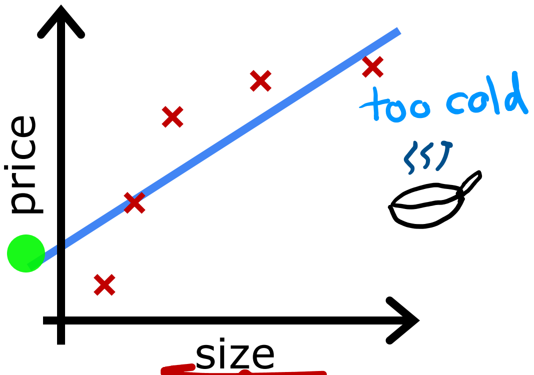
DeepLearning.AI



Regularization to Reduce Overfitting

The Problem of Overfitting

Regression example

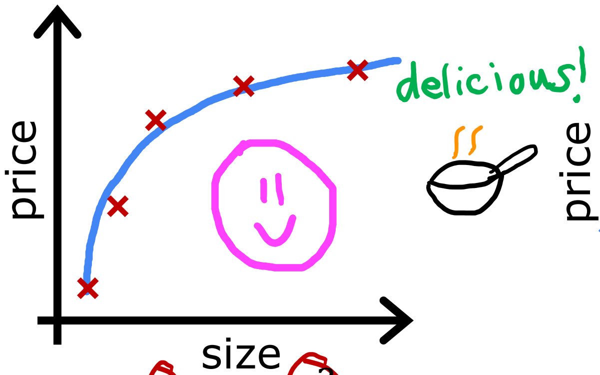


$$w_1 x + b$$

underfit

- Does not fit the training set well

high bias

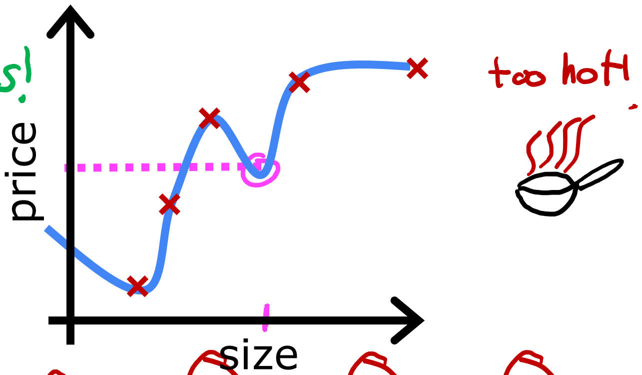


$$w_1 x + w_2 x^2 + b$$

just right

- Fits training set pretty well

generalization



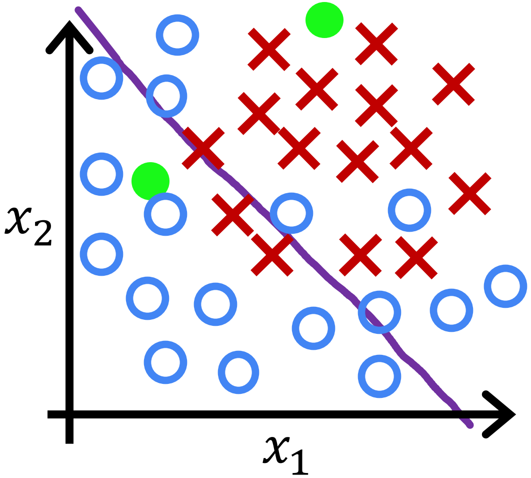
$$w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$$

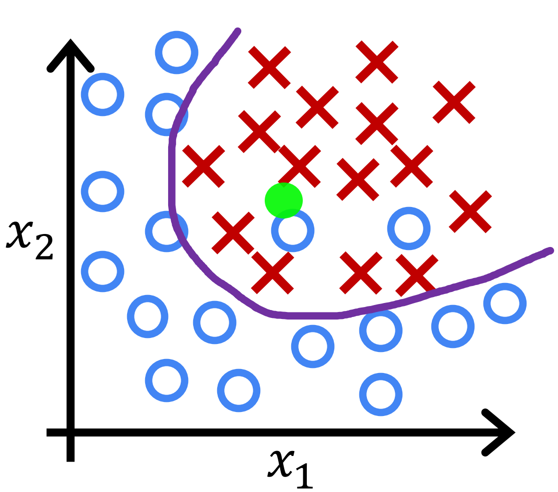
overfit

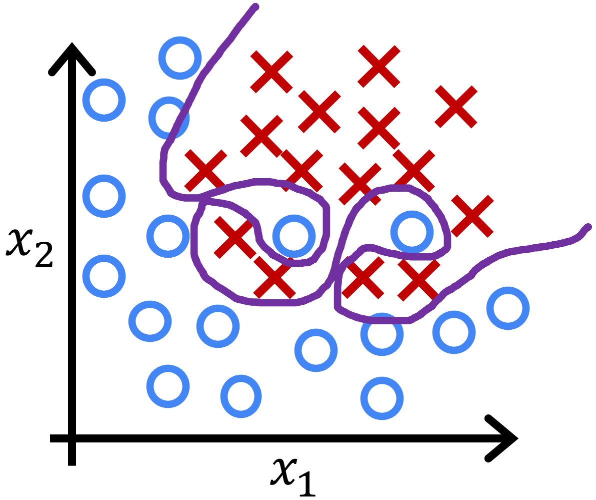
- Fits the training set extremely well

high variance

Classification







$$z = w_1 x_1 + w_2 x_2 + b$$

$$f_{\vec{w},b}(\vec{x}) = g(z)$$

g is the sigmoid function

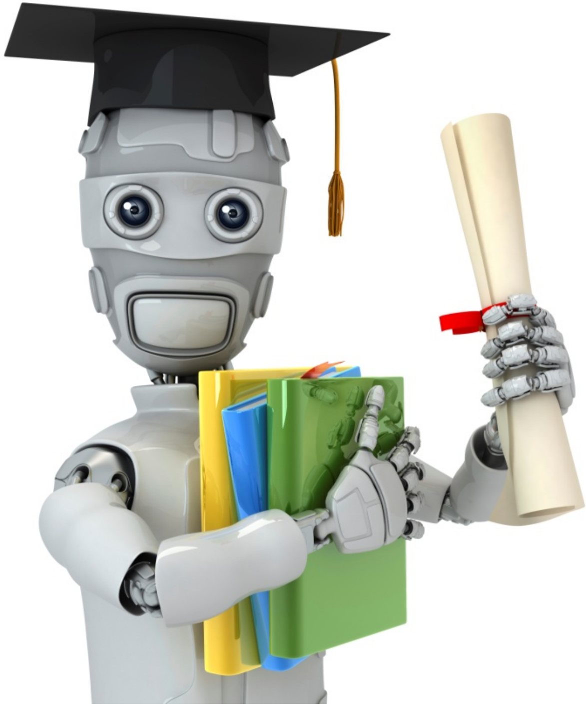
underfit high bias

$$z = w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 + w_5 x_1 x_2 + b$$

just right

$$z = w_1 x_1 + w_2 x_2 + w_3 x_1^2 x_2 + w_4 x_1^2 x_2^2 + w_5 x_1^2 x_2^3 + w_6 x_1^3 x_2 + \cdots + b$$

overfit

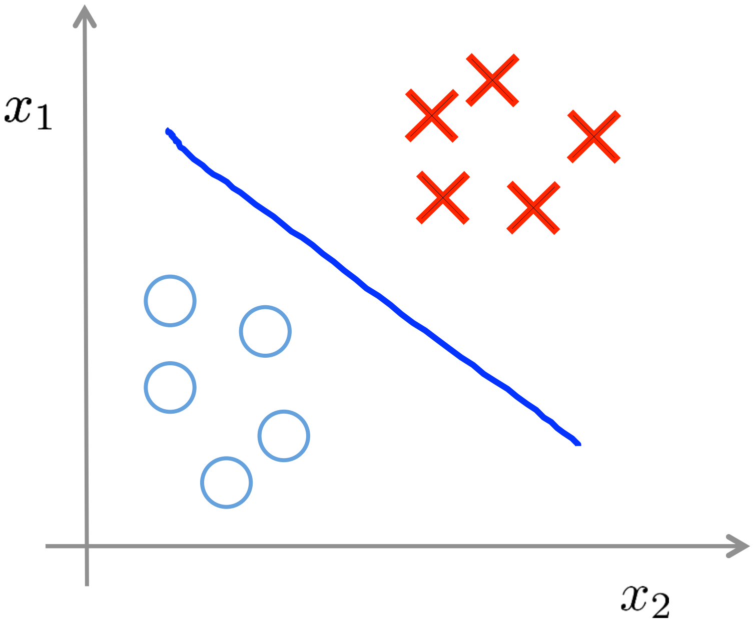


Machine Learning

Clustering

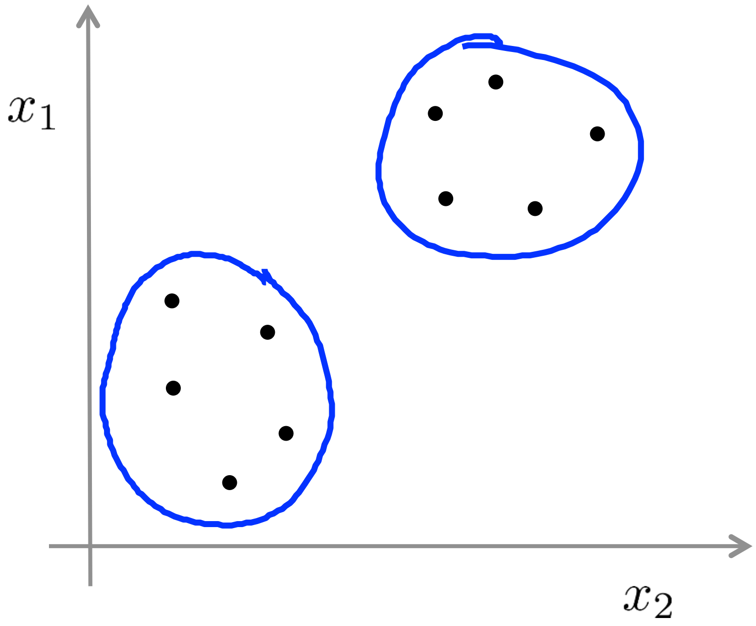
Unsupervised learning introduction

Supervised learning



Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots, (x^{(m)}, y^{(m)})\}$

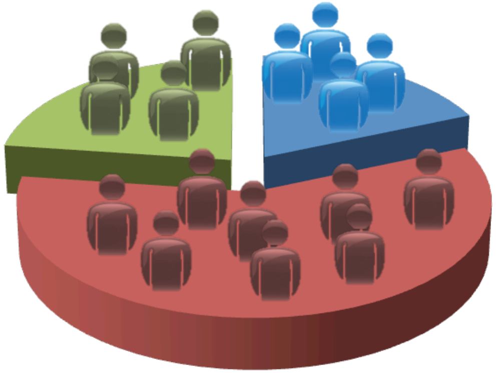
Unsupervised learning



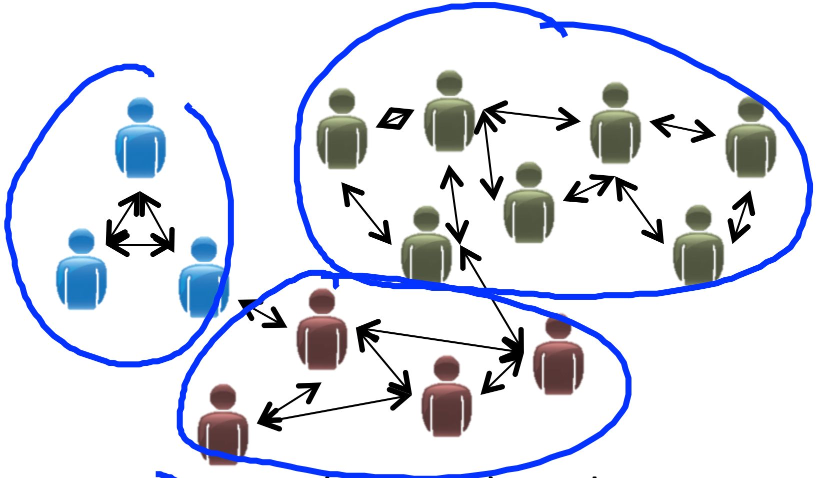
Clustering algorithm

Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$ ←

Applications of clustering

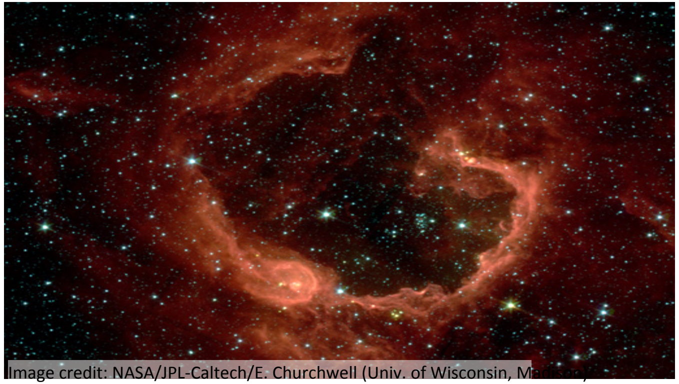


Market segmentation



Social network analysis

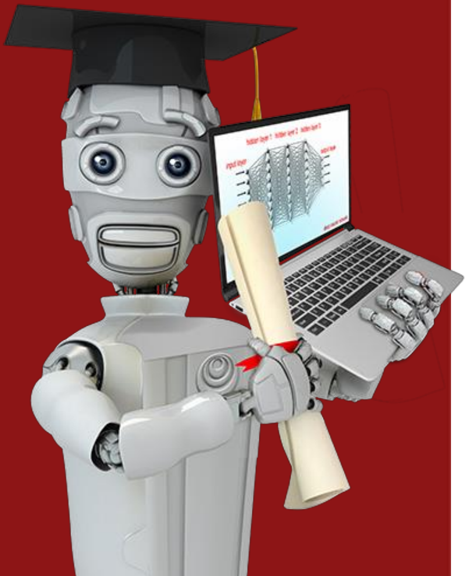
Organize computing clusters



Astronomical data analysis

DeepLearning.AI

Stanford ONLINE



Neural Networks Intuition

Neurons and the brain

Neural networks

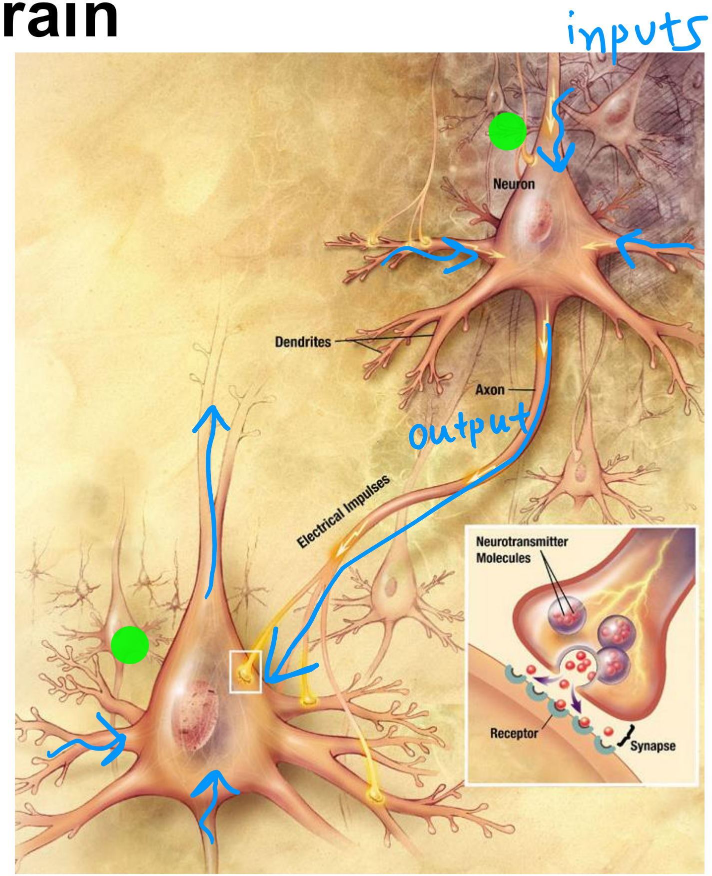
Origins: Algorithms that try to mimic the brain.

Used in the 1980's and early 1990's.
Fell out of favor in the late 1990's.

Resurgence from around 2005.

speech → images → text (NLP) → ...

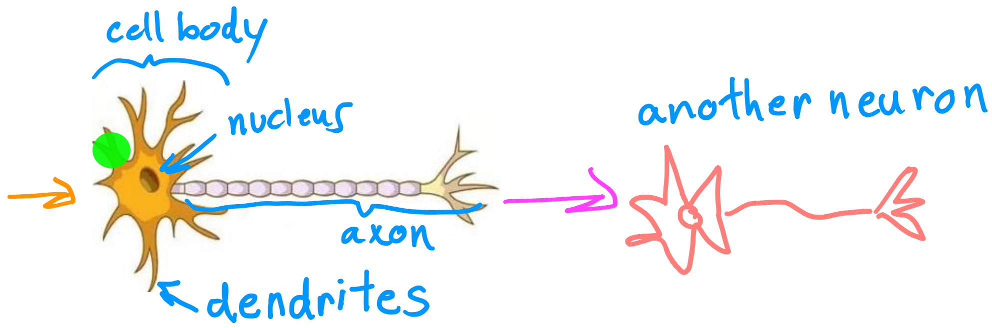
Neurons in the brain



[Credit: US National Institutes of Health, National Institute on Aging]

Biological neuron

inputs outputs



Simplified mathematical model of a neuron

inputs outputs

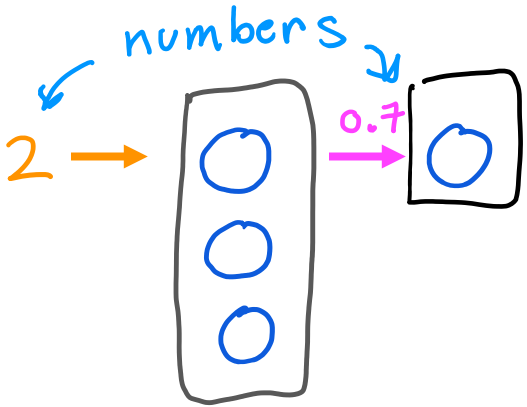
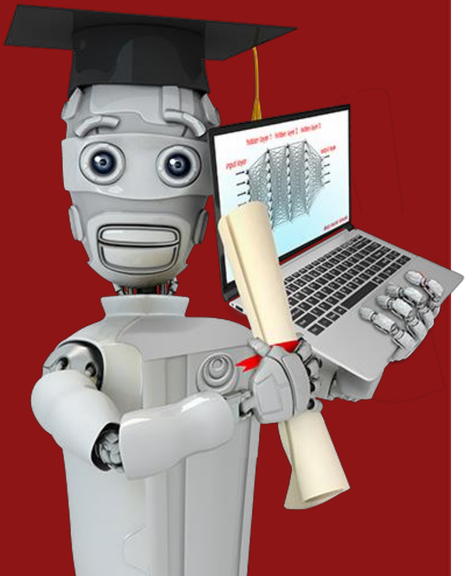


image source: https://biologydictionary.net/sensory-neuron/

DeepLearning.AI

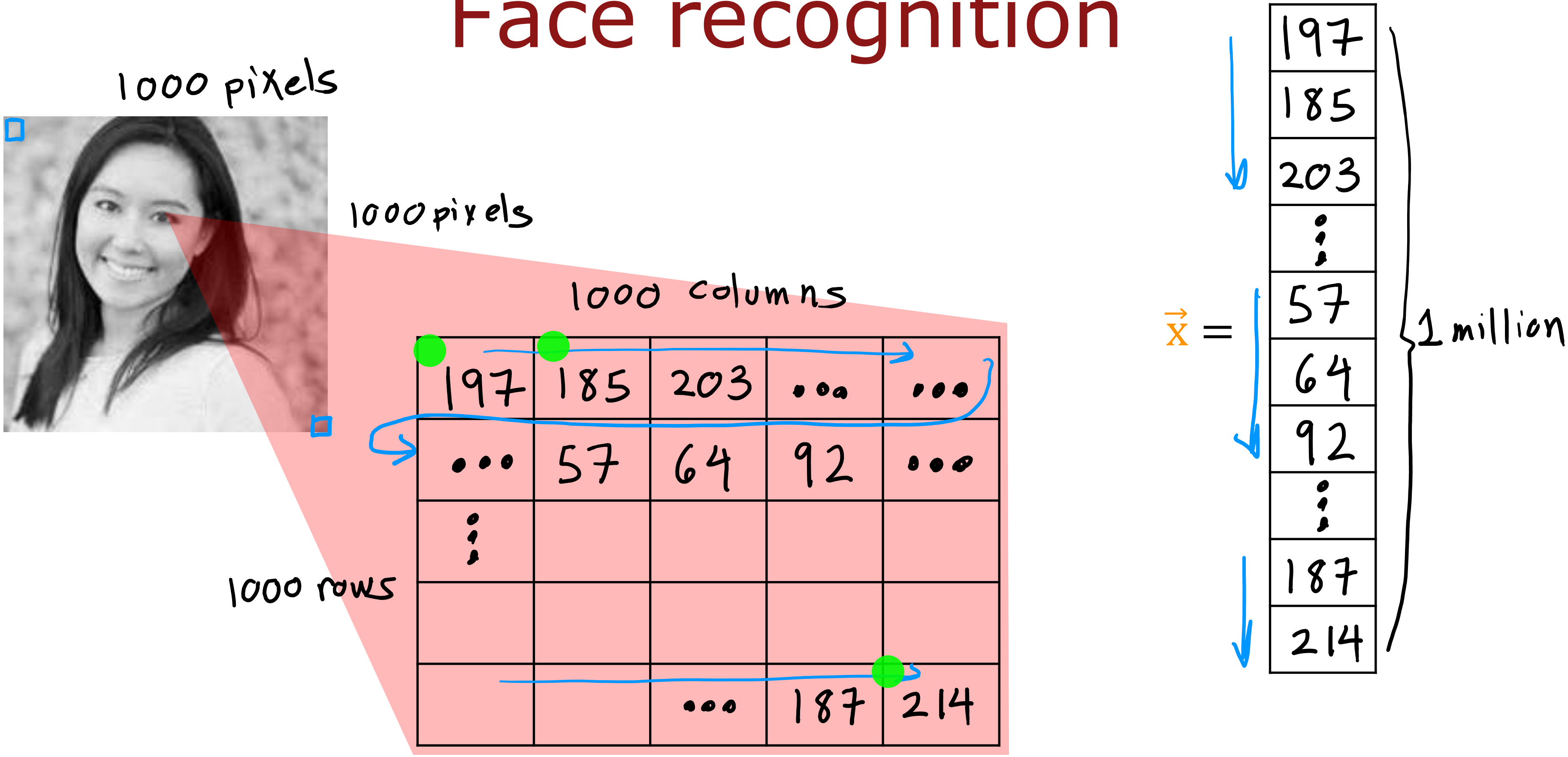
Stanford
ONLINE

Neural Networks Intuition

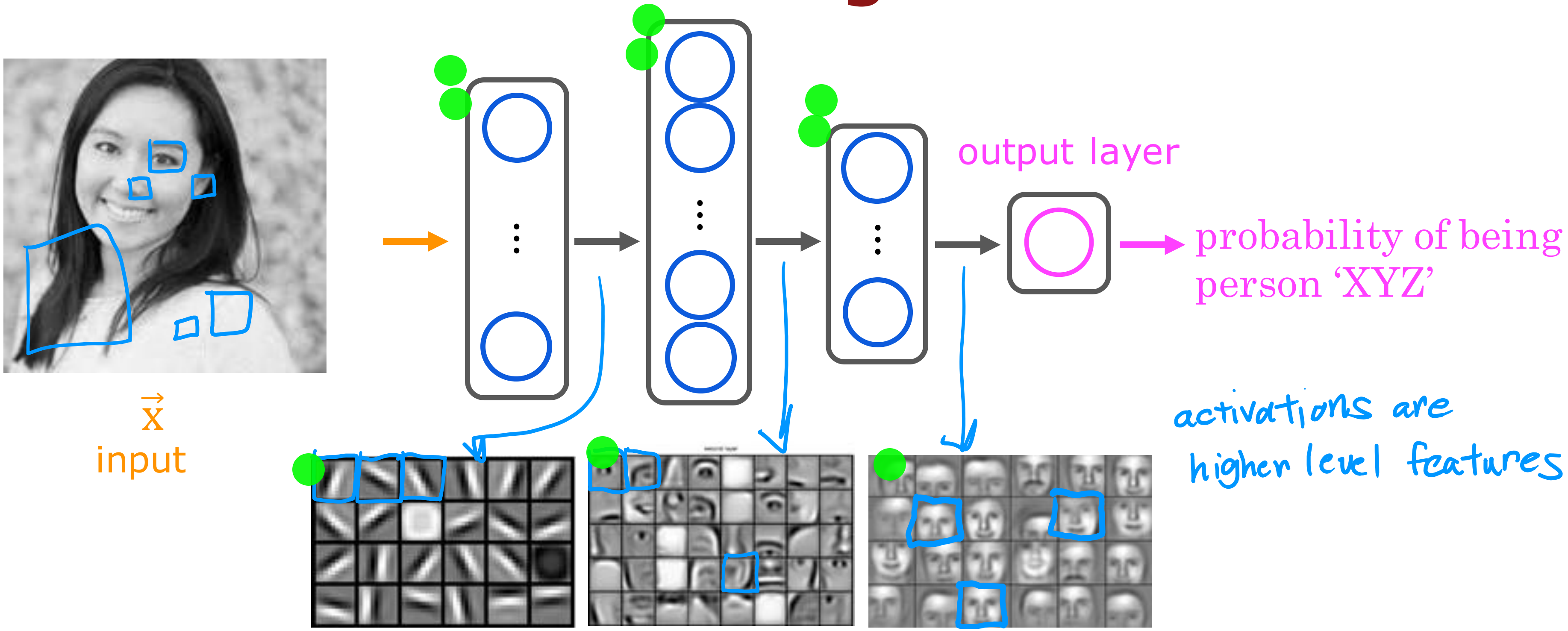


Example:
Recognizing Images

Face recognition

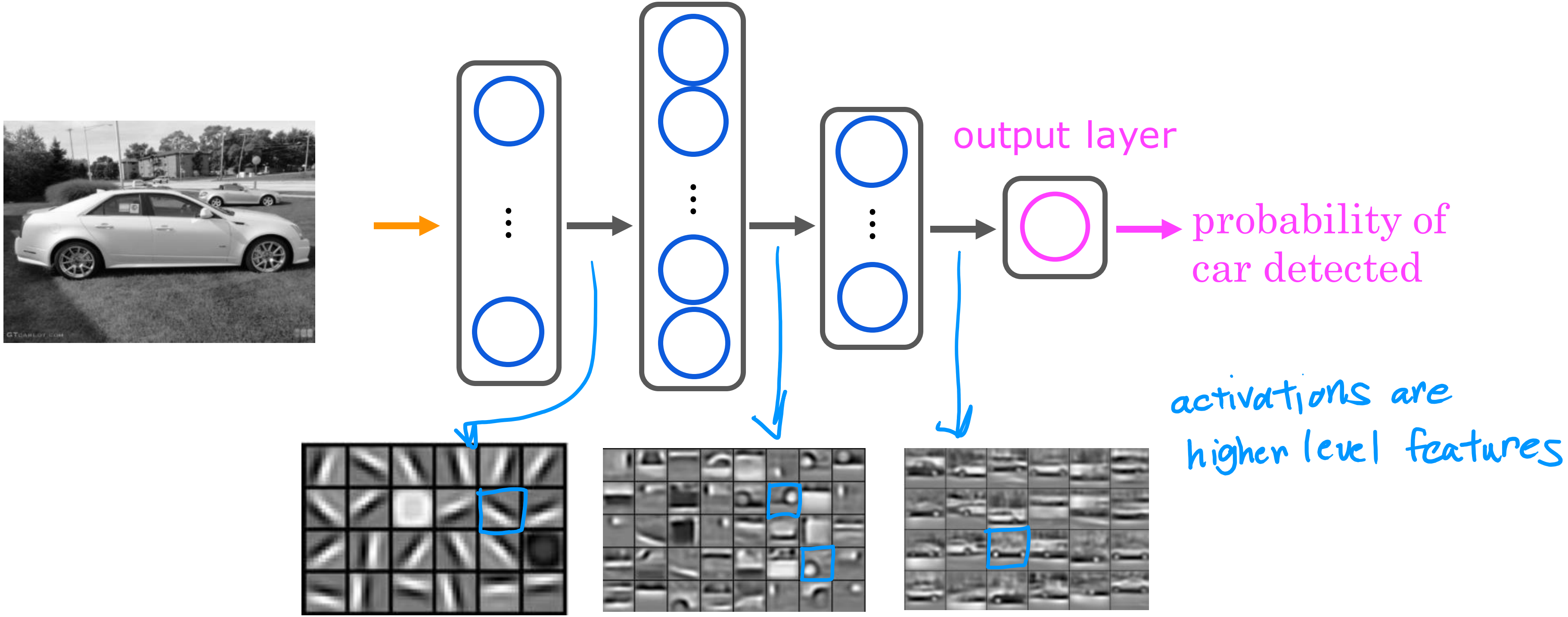


Face recognition



source: Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations by Honglak Lee, Roger Grosse, Ranganath Andrew Y. Ng

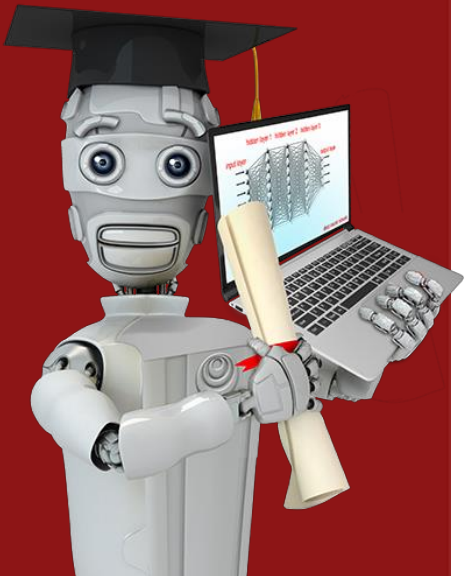
Car classification



source: Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations by Honglak Lee, Roger Grosse, Ranganath Andrew Y. Ng

DeepLearning.AI

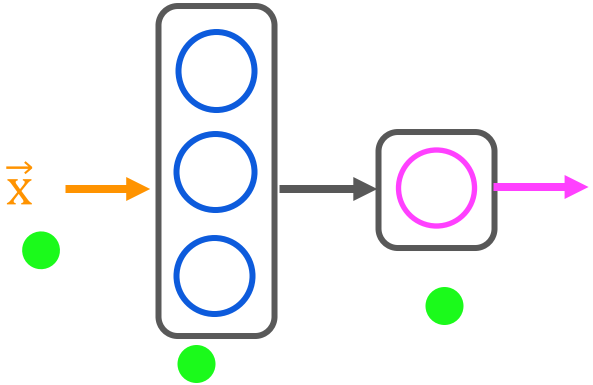
Stanford ONLINE



Neural network model

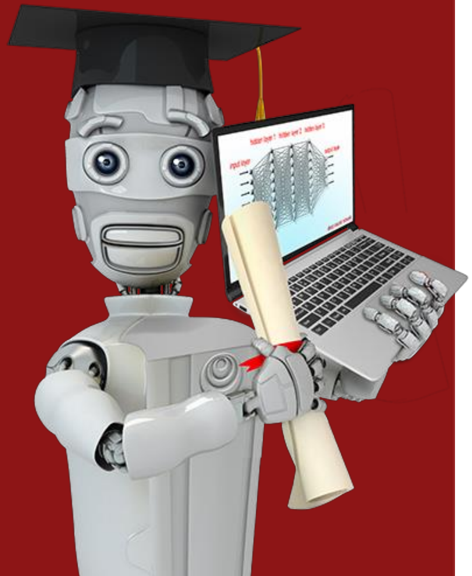
Neural network layer

Neural network layer



DeepLearning.AI

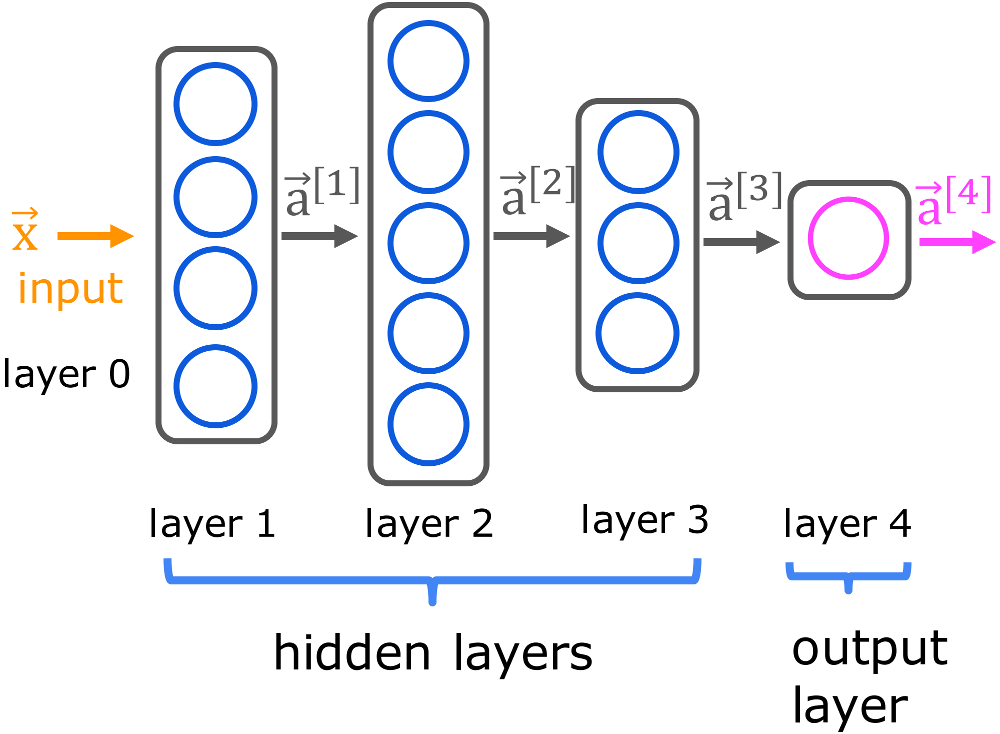
Stanford ONLINE



Neural Network Model

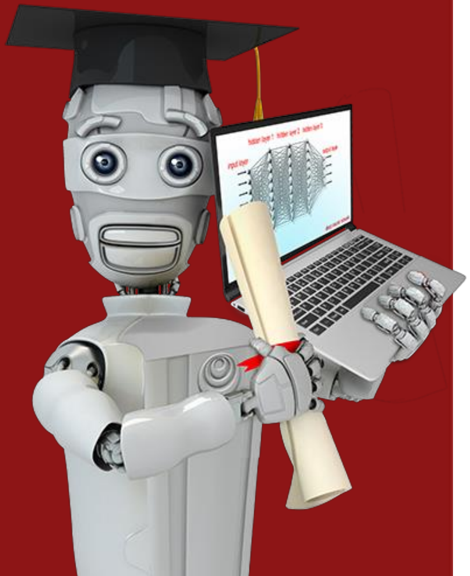
More complex neural networks

More complex neural network



DeepLearning.AI

Stanford ONLINE



Speculations on artificial general intelligence (AGI)

Is there a path to AGI?

AI

ANI
(artificial narrow intelligence)

E.g., smart speaker,
self-driving car, web search,
AI in farming and factories

AGI
(artificial general intelligence)

Do anything a human can do