Neural Networks- An introduction

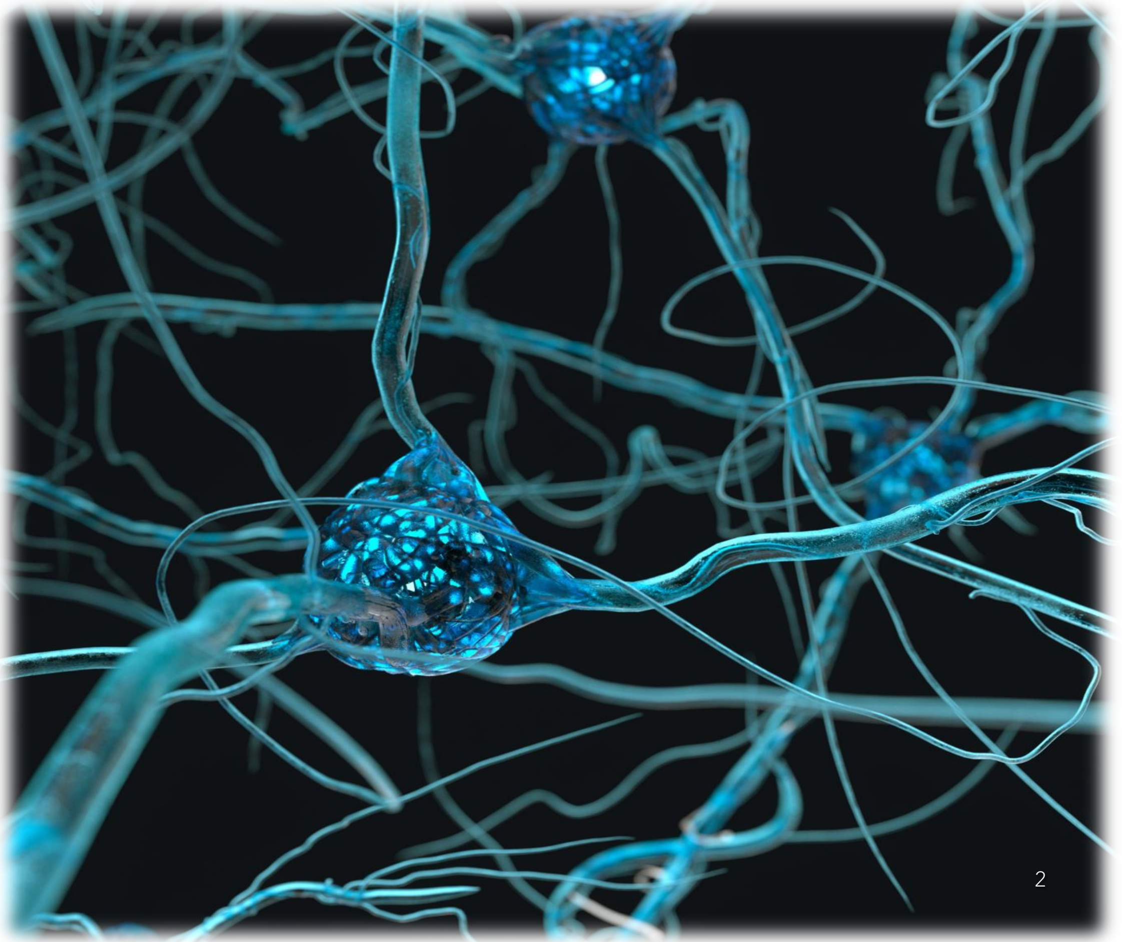
INSTRUCTOR: AVNEET KAUR

EMAIL: A93KAUR@UWATERLOO.CA

Neurons in Biology to Artificial Neural Networks(ANNs)

- Human brain cells, called neurons, form a
  - complex,
  - highly interconnected network and
  - send electrical signals to each other

  to help humans process information.
- Similarly, an ANN is made of artificial neurons that work together to solve a problem.



What are they used for?

Computer vision: Extract information from images and videos

In self-driving cars, to recognize road signs and other road users

Inappropriate content removal from websites

Facial recognition

Image labelling to identify different objects

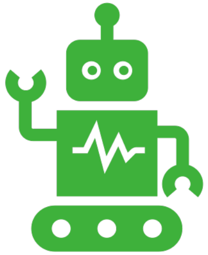
Speech recognition: Analyze human speech despite variations

Automatically classify calls at call centers

Convert clinic calls into documented text

Subtitle generation for videos and meetings

What are they used for? (Contd.)



Natural language processing: Process natural, human-created text

Automation of virtual agents and chatbots

Analysis of emails and long forms

Distinguish positive and negative content

Organize and classify data

Recommendation engines: Develop personalized recommendations based on behaviors

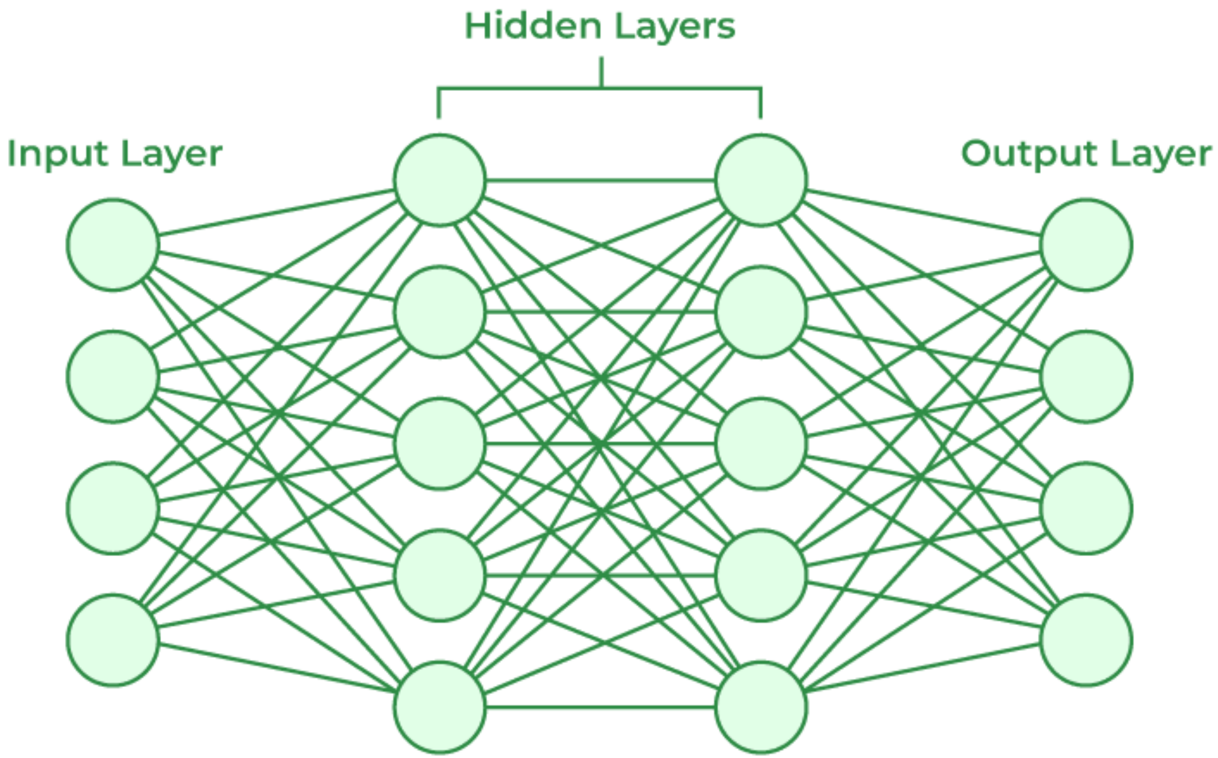
Personalized recommendations on Netflix, YouTube, Google and many other websites

Convert social media posts into sales using automatically tagged products

How do they work?

Input layer

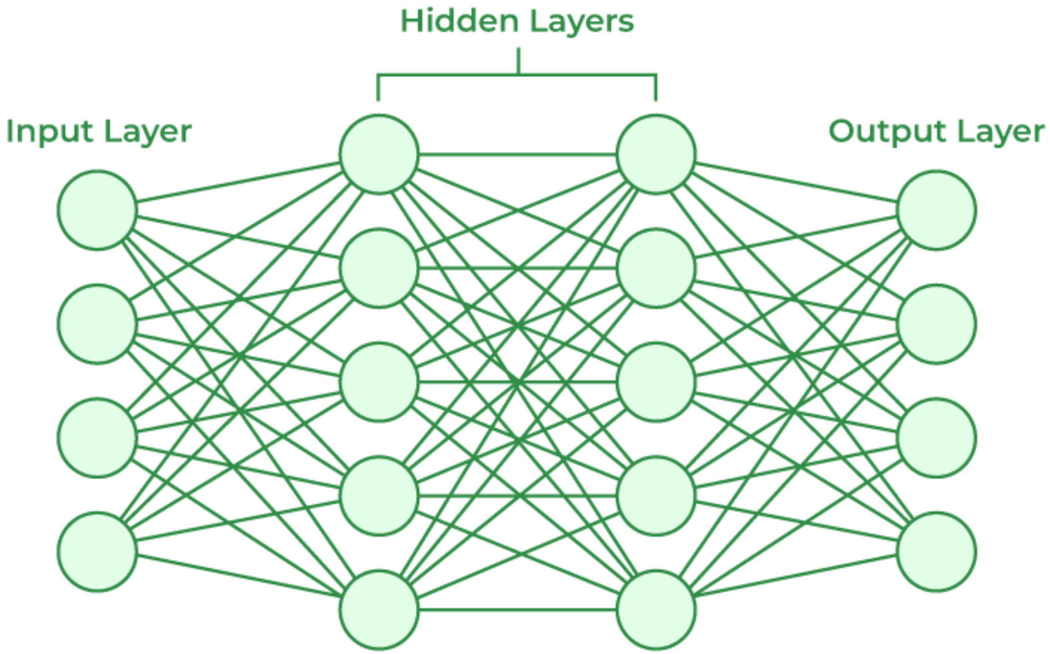
- Information from the outside world enters the artificial neural network from the input layer.
- Input nodes process the data, analyze or categorize it, and pass it on to the next layer.



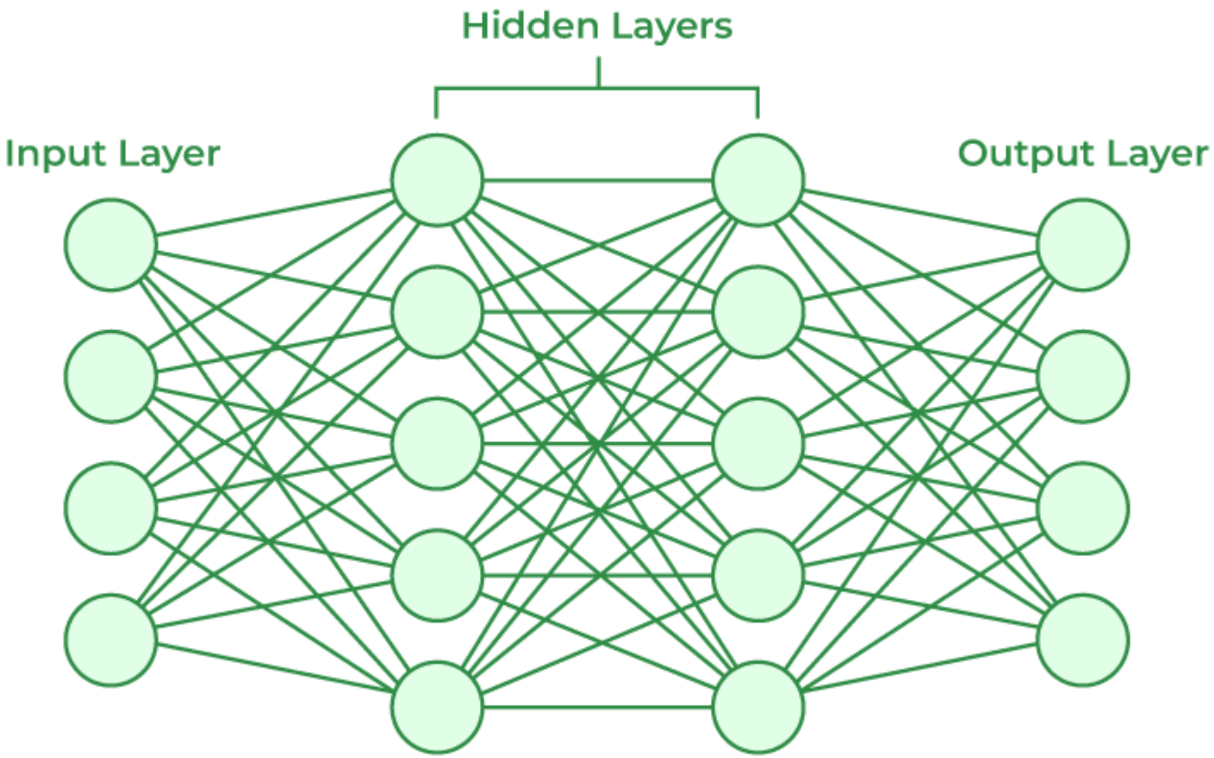
How do they work?

Hidden layers

- Hidden layers take their input from the input layer or other hidden layers.
- Artificial neural networks can have many hidden layers.
- Each hidden layer analyzes the output from the previous layer, processes it further, and passes it on to the next layer.



How do they work?

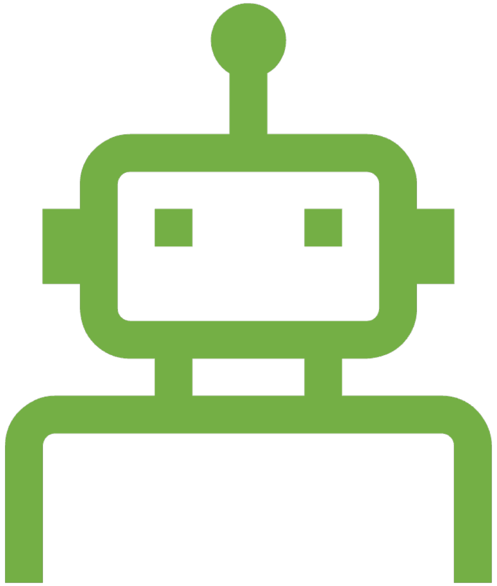


Output Layer

- Gives the result of all the data processed by the ANN.
- May have a single or multiple nodes.
- For instance, if we have a binary (yes/no) classification problem, the output layer will have one output node, which will give the result as 1 or 0. However, if we have a multi-class classification problem, the output layer might consist of more than one output node.

How do they work?

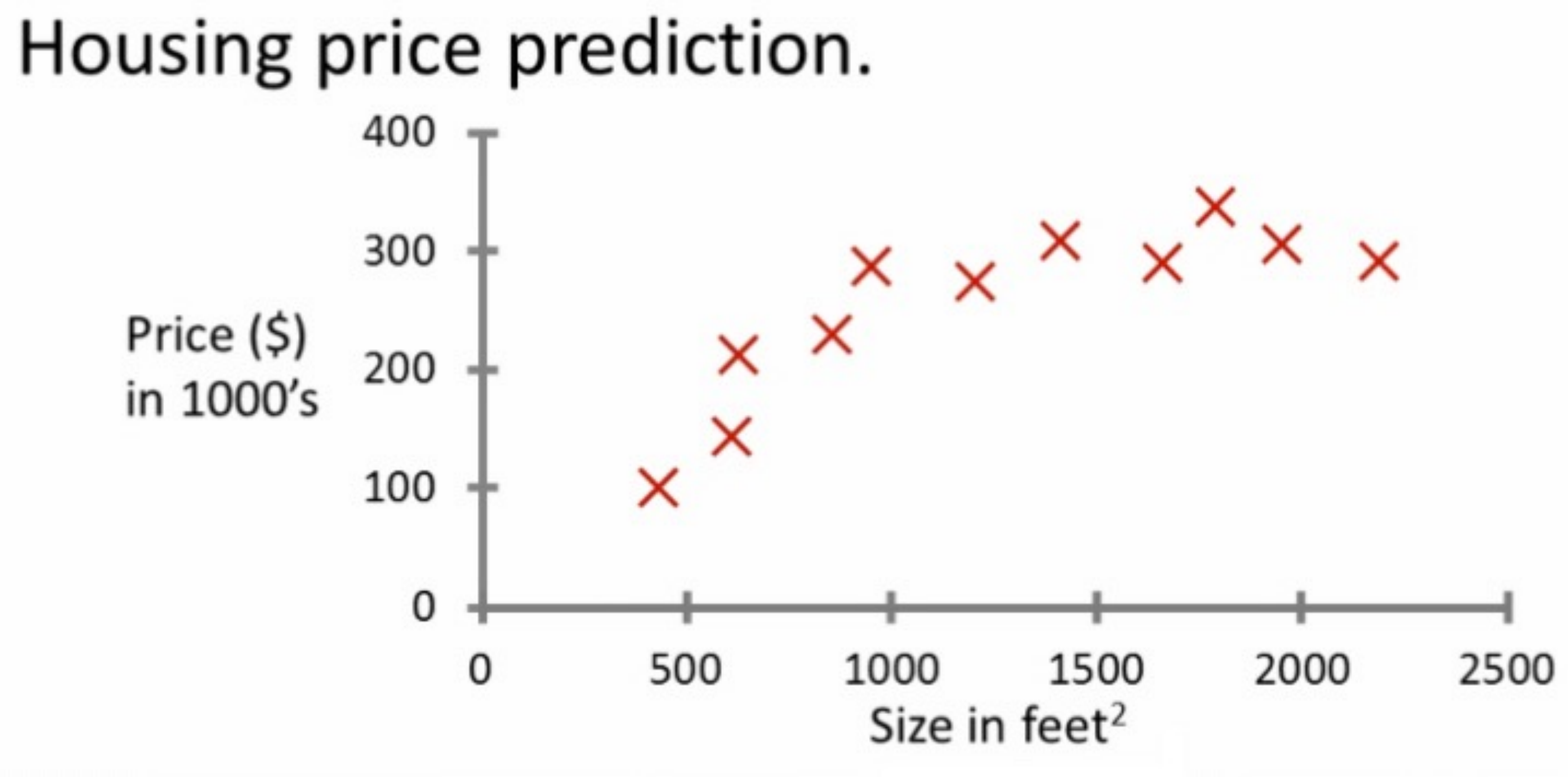
- 3 major parts:
  - Training: Make your neural network fit for the data.
  - Validating: check whether it is sufficiently fit.
  - Testing: Deploy it in the application and test its working.



Training neural networks

- Supervised learning
  - Labelled dataset: The neural network knows the true answer for the purpose of training and validating
  - Classification and Regression
- Unsupervised learning
  - Unlabeled dataset: The neural network does not know the true answers even for training
  - It tries to group data based on similarity
  - Clustering
- Reinforcement learning
  - Generate dataset: The neural network has access to the environment and learns from mistakes with some partial feedback availability
  - Robot navigation

Supervised Learning: The Regression Problem



Regression: To predict a number from infinitely many possibilities

$\hat{y}^{(1)} = x^{(1)} + 2 = 2106$

Data table

	size in feet²	price in $1000's	
$x^{(1)} \rightarrow$	2104	400	$\rightarrow y^{(1)}$
$x^{(2)} \rightarrow$	1416	232	$\rightarrow y^{(2)}$
$x^{(3)} \rightarrow$	1534	315	$\rightarrow y^{(3)}$
	852	178	$\rightarrow y^{(4)}$
	...	...	
	3210	870	

m training examples.

Problem: Given the above data, figure out a rough estimate for the price of a house that has 5000 square feet area.

Linear regression: Consider f(x)=Wx+b and train the neural network to learn W and b based on given training data.

How can I do this? Start with a guess of W and b and try to update them with better W and b values.

How to know which value is better?

Cost function

- Mean square error cost function

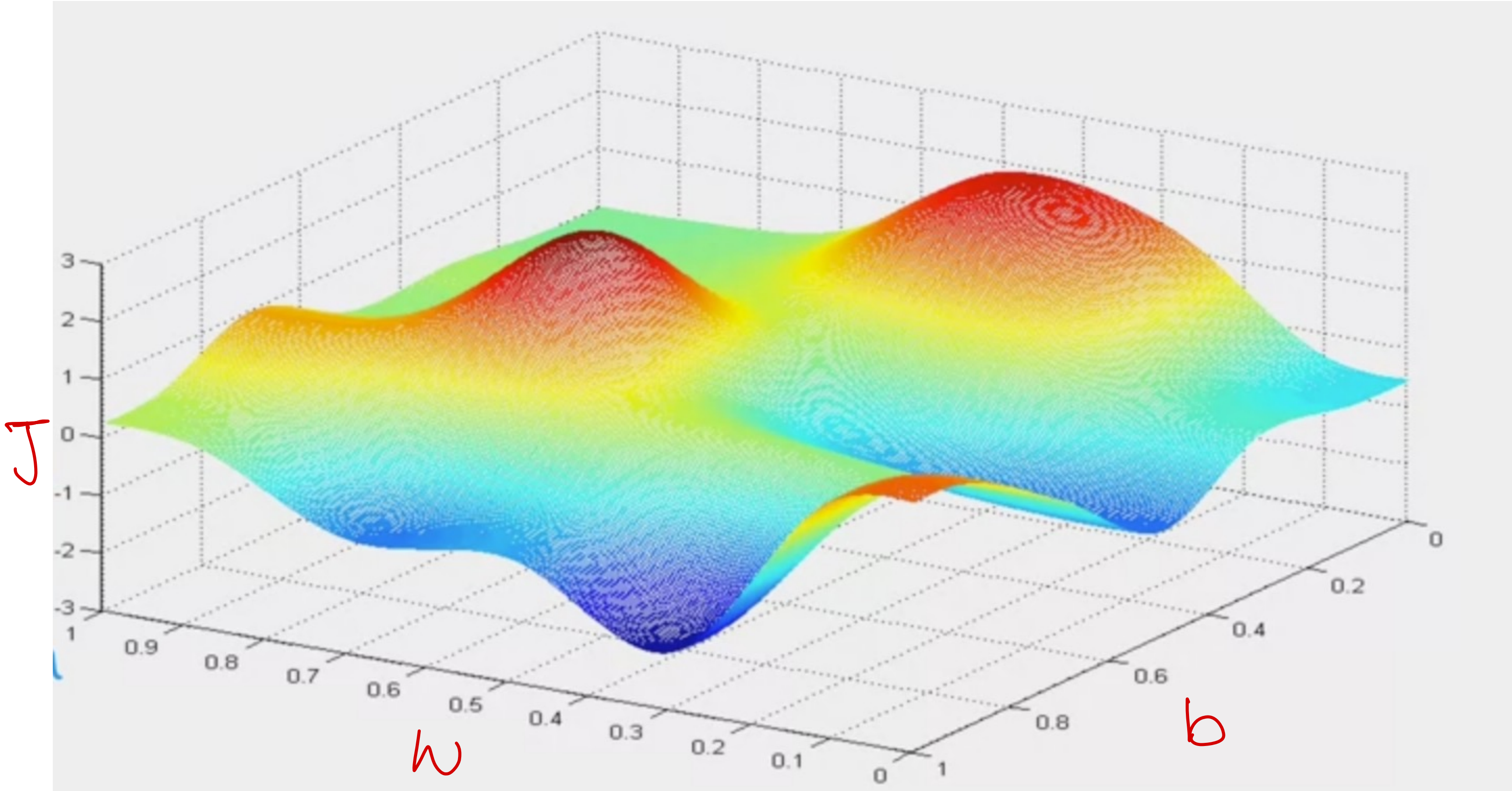
$$J(w,b) = \left(\sum_{i=1}^{m} (\overbrace{\hat{y}^{(i)}}^{Wx^{(i)}+b} - y^{(i)})^2\right)\frac{1}{2m}$$

The closer is the loss value to 0, the better model we get.

What if it's not close to 0?

$$\min_{W,b} J(W,b)$$

Gradient Descent



Gradient descent algorithm

- $w = w - \alpha \dfrac{\partial J(w,b)}{\partial w}$
- $b = b - \alpha \dfrac{\partial J(w,b)}{\partial b}$
- α is the learning rate

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

$$= \frac{1}{2m} \sum_{i=1}^{m} \left(w x^{(i)} + b - y^{(i)}\right)^2$$

Please provide your feedback for the last 3 sessions here:

https://forms.gle/xekicd7ctpVBcdHr8

Have a great rest of the year!

Questions

Q1- Find $\frac{\partial J(w,b)}{\partial w}$.

Q2- Find $\frac{\partial J(w,b)}{\partial b}$.

Q3- Show the first iteration of linear regression for the following data:

	$x^{(i)}$	$y^{(i)}$
$i=1$	0	4
$i=2$	1	7
$i=3$	2	8
$i=4$	3	4

Start with an initial guess of $W=1$ and $b=3$.

Solutions

Sol. 1 → $$\frac{\partial J}{\partial w} = \frac{1}{m}\sum_{i=1}^{m}\left(Wx^{(i)} + b - y^{(i)}\right) \cdot x^{(i)}$$

Sol. 2 → $$\frac{\partial J}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}\left(Wx^{(i)} + b - y^{(i)}\right)$$

Sol. 3 → $W = 1, \; b = 3, \; \alpha = \frac{1}{9}$

$$\hat{y}^{(i)} = Wx^{(i)} + b = x^{(i)} + 3$$

$$\hat{y}^{(1)} = x^{(1)} + 3 = 0 + 3 = 3$$
$$\hat{y}^{(2)} = x^{(2)} + 3 = 1 + 3 = 4$$
$$\hat{y}^{(3)} = x^{(3)} + 3 = 2 + 3 = 5$$
$$\hat{y}^{(4)} = x^{(4)} + 3 = 3 + 3 = 6$$

$$J(W,b) = J(1,3) = \frac{1}{2(4)}\left[\left(\hat{y}^{(1)} - y^{(1)}\right)^2 + \left(\hat{y}^{(2)} - y^{(2)}\right)^2 + \left(\hat{y}^{(3)} - y^{(3)}\right)^2 + \left(\hat{y}^{(4)} - y^{(4)}\right)^2\right]$$

$$= \frac{1}{8}\left((3-4)^2 + (4-7)^2 + (5-8)^2 + (6-9)^2\right)$$

$$= \frac{1}{8}\left(1^2 + 3^2 + 3^2 + 3^2\right)$$

$$= \frac{1}{8}(28) > 0.5$$

$$W = 1 - \frac{1}{9}\left(\frac{\partial J}{\partial W}\right) = 1 - \frac{1}{9}\left(\frac{-9}{2}\right) = 1 + \frac{1}{2} = 1.5$$

$$b = 3 - \frac{1}{9}\left(\frac{\partial J}{\partial b}\right) = 3 - \frac{1}{9}\left(\frac{-5}{2}\right) = 3 + \frac{5}{18} \approx 3.28$$

Gradient descent algorithm

- $w = w - \alpha \frac{\partial J(w,b)}{\partial w}$
- $b = b - \alpha \frac{\partial J(w,b)}{\partial b}$
- α is the learning rate

Please leave feedback at this link:

https://forms.gle/xekicd7ctpVBcdHr8