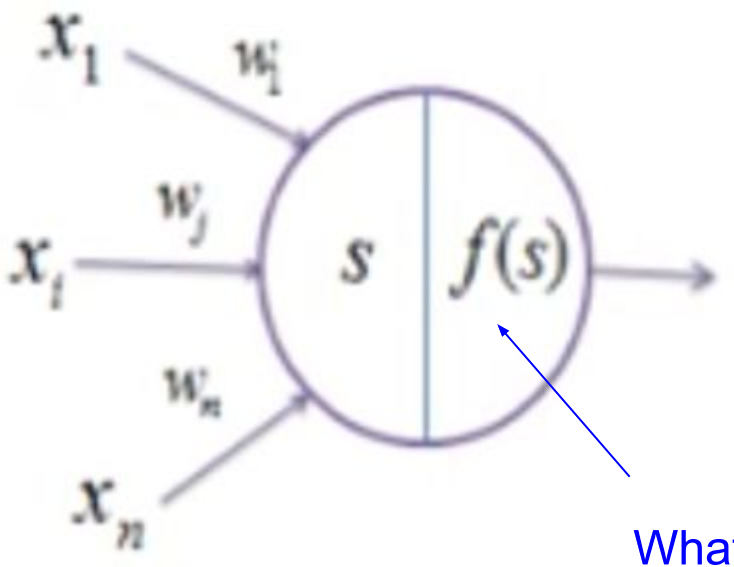
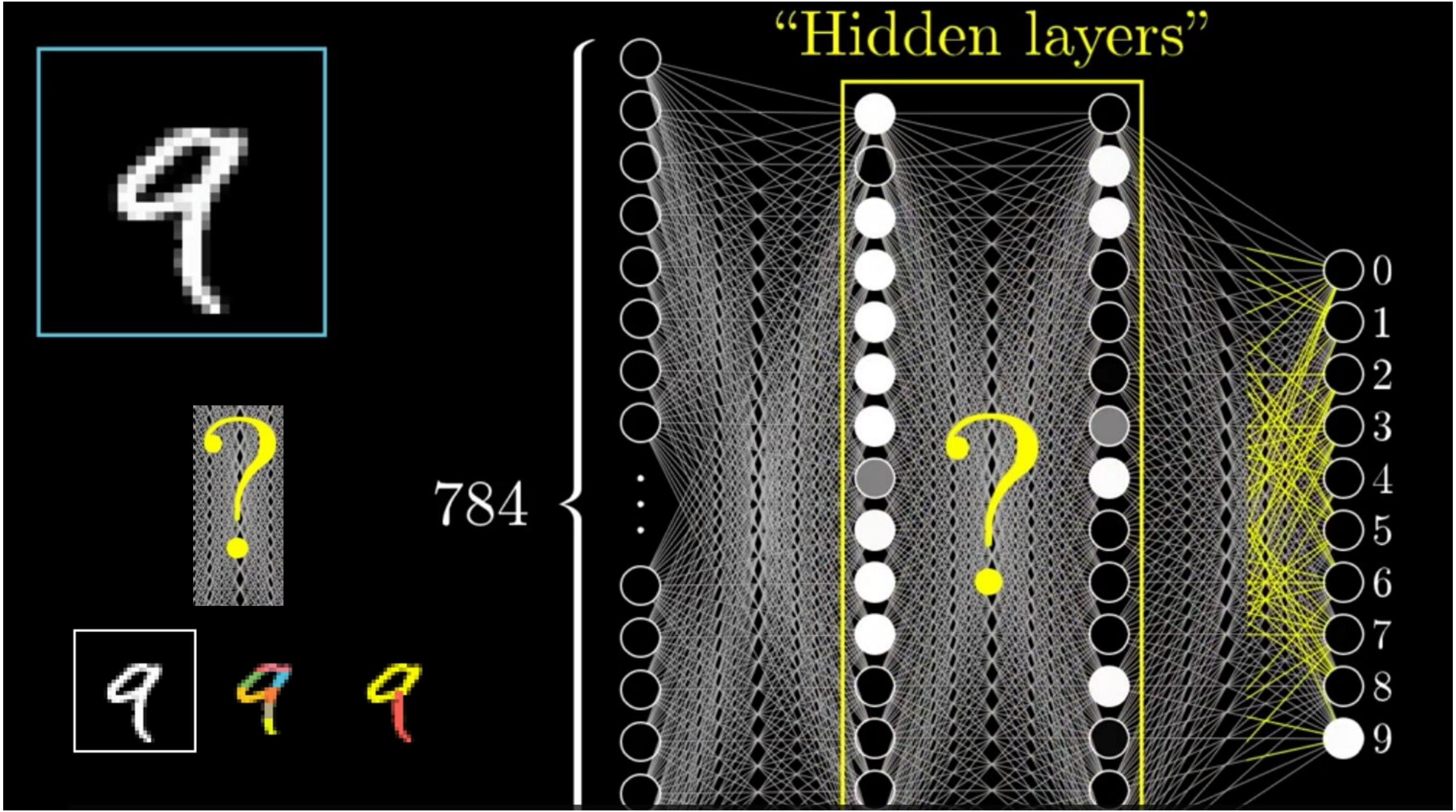
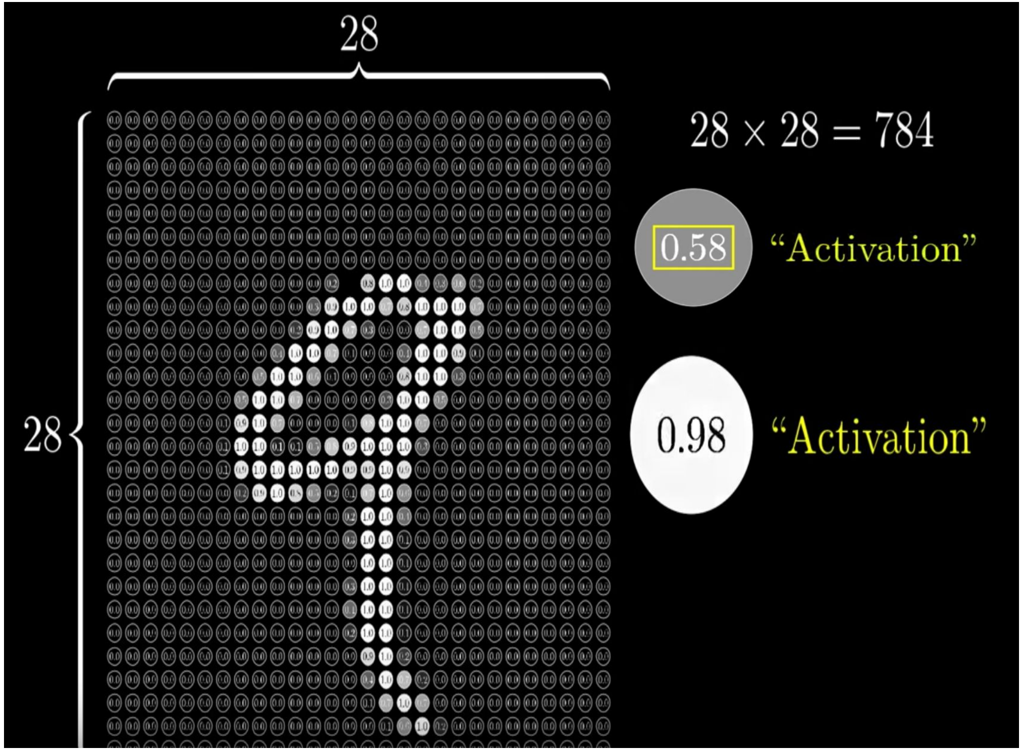
What is Perceptron Activation Function ?

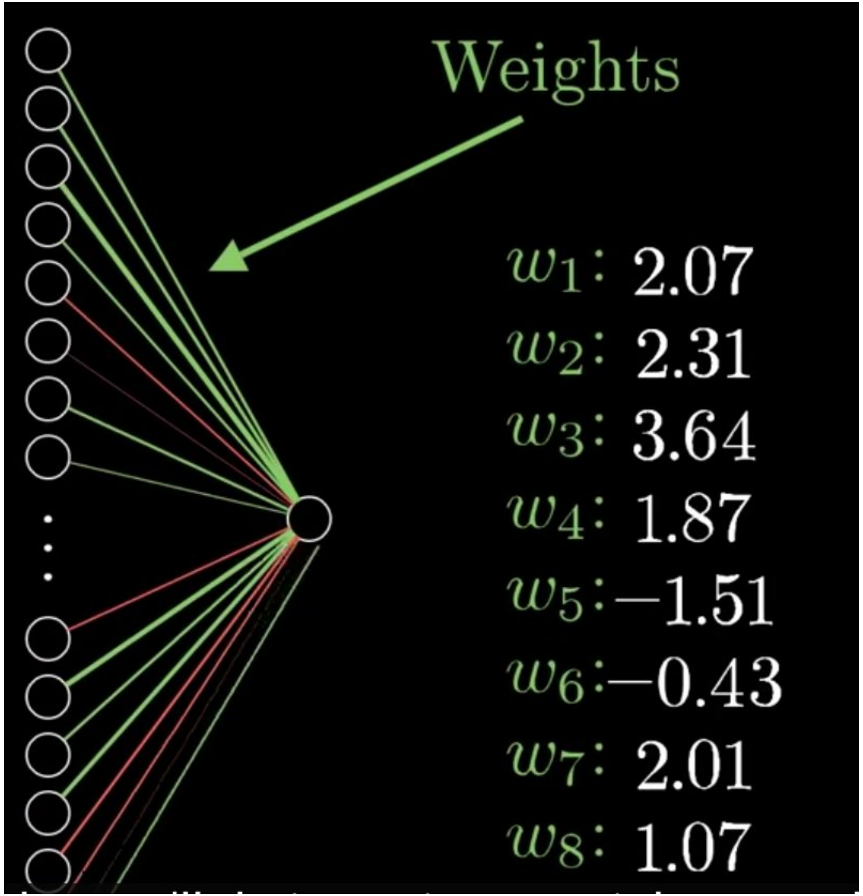
Created and Presented By:
Dr.Mydhili K Nair
Professor
Dept of ISE, MSRIT
For: Sem 6 Elective Class on Machine Learning(Jan-May 2019)



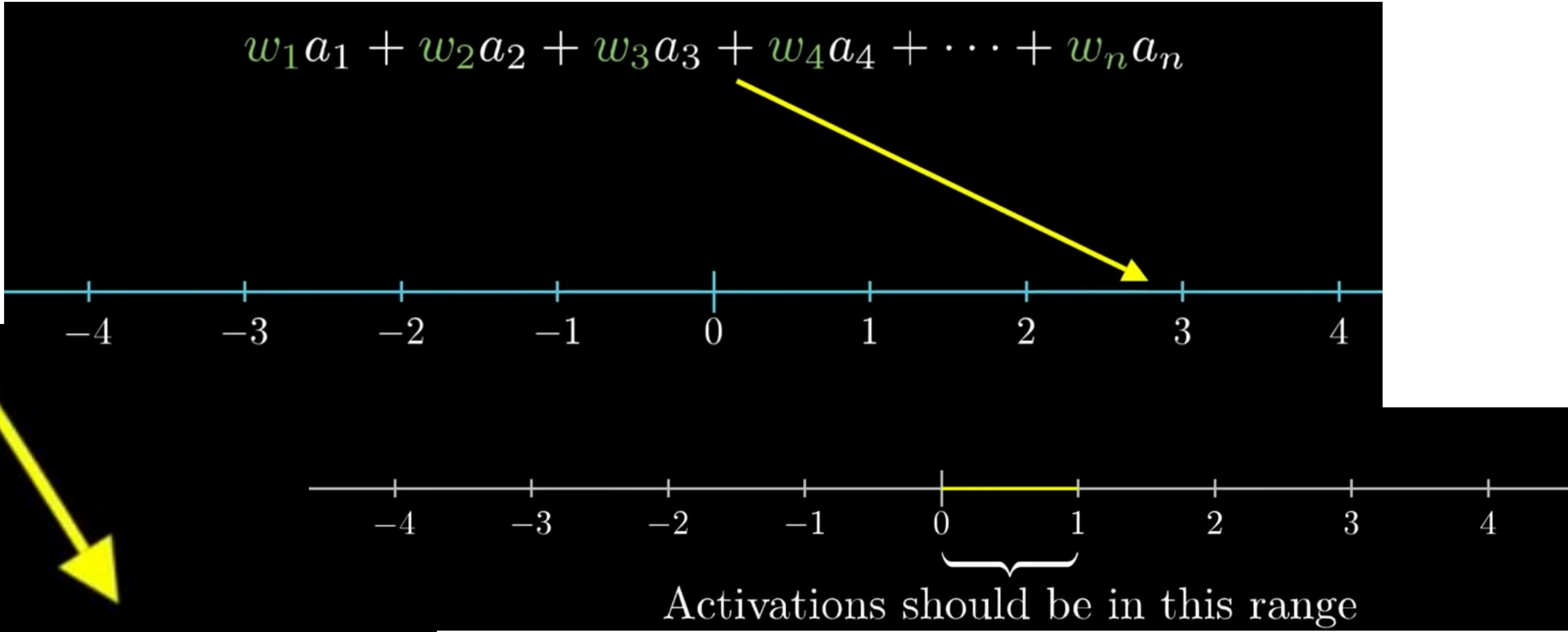
What is this?
Why do we need this?

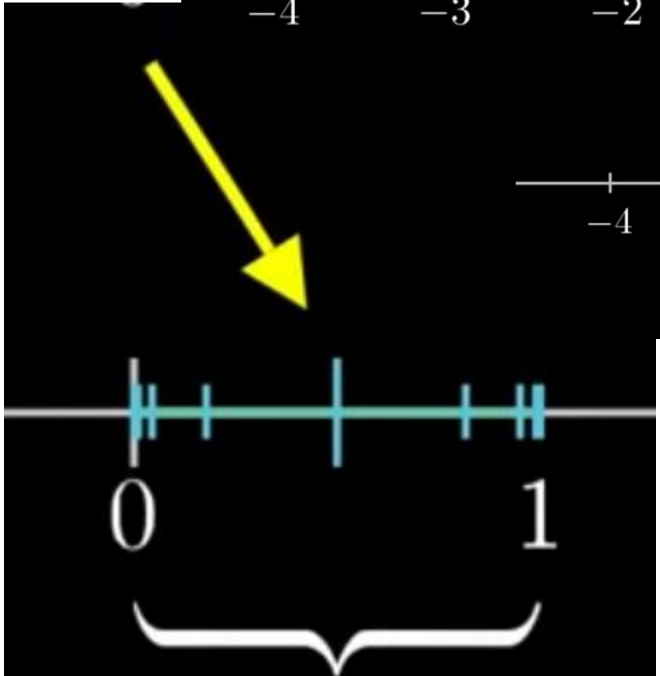

"Hidden layers"
784
0
1
2
3
4
5
6
7
8
9


28
28
28 × 28 = 784
0.58
“Activation”
0.98
“Activation”

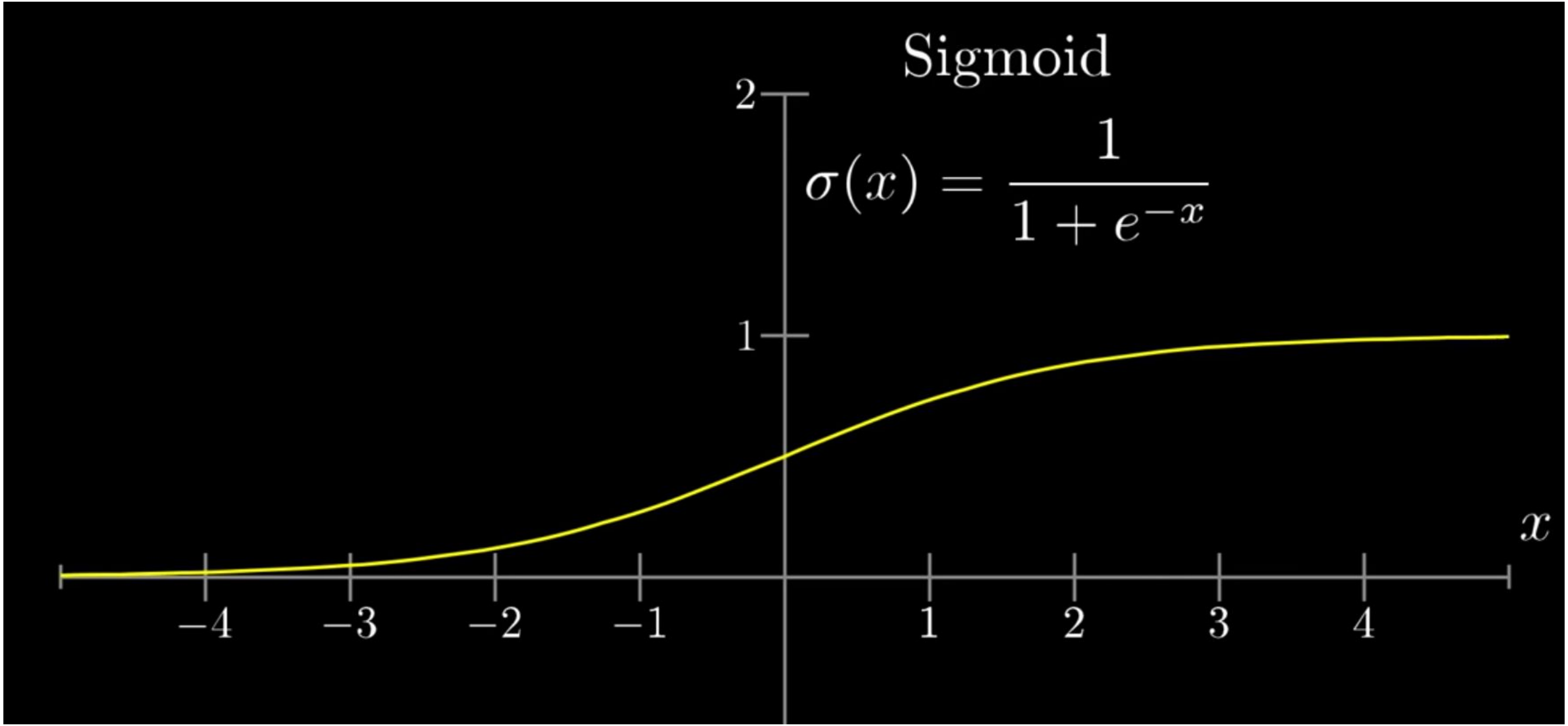


“Weights” can be equated to the “strength” between the neuron and the inputs connected to it.



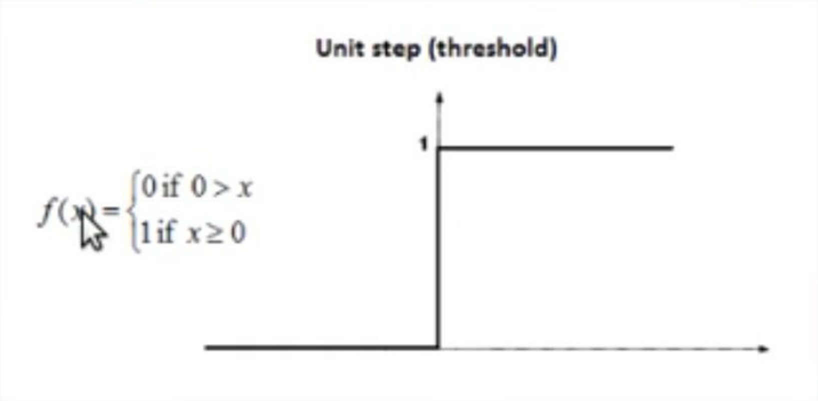


Activation Function helps to **squeeze** the values got within the range from 0 to 1.


Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$
2
1
x
−4
−3
−2
−1
1
2
3
4

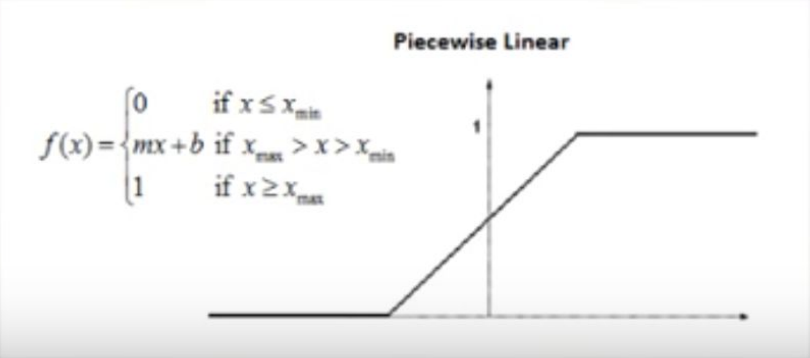
Unit step (threshold)

- The output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value.



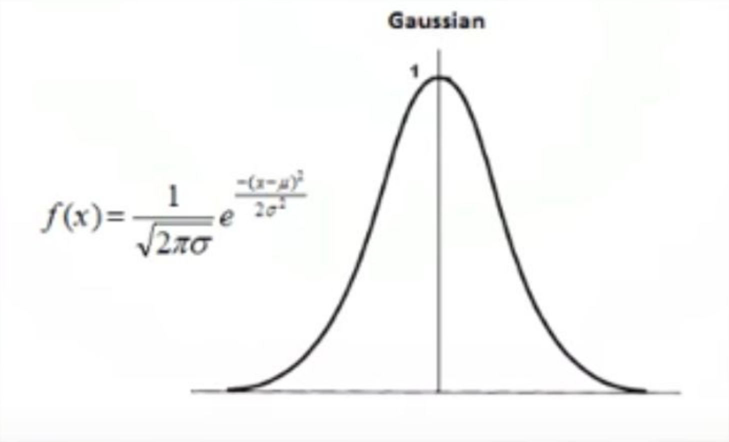
Piecewise Linear

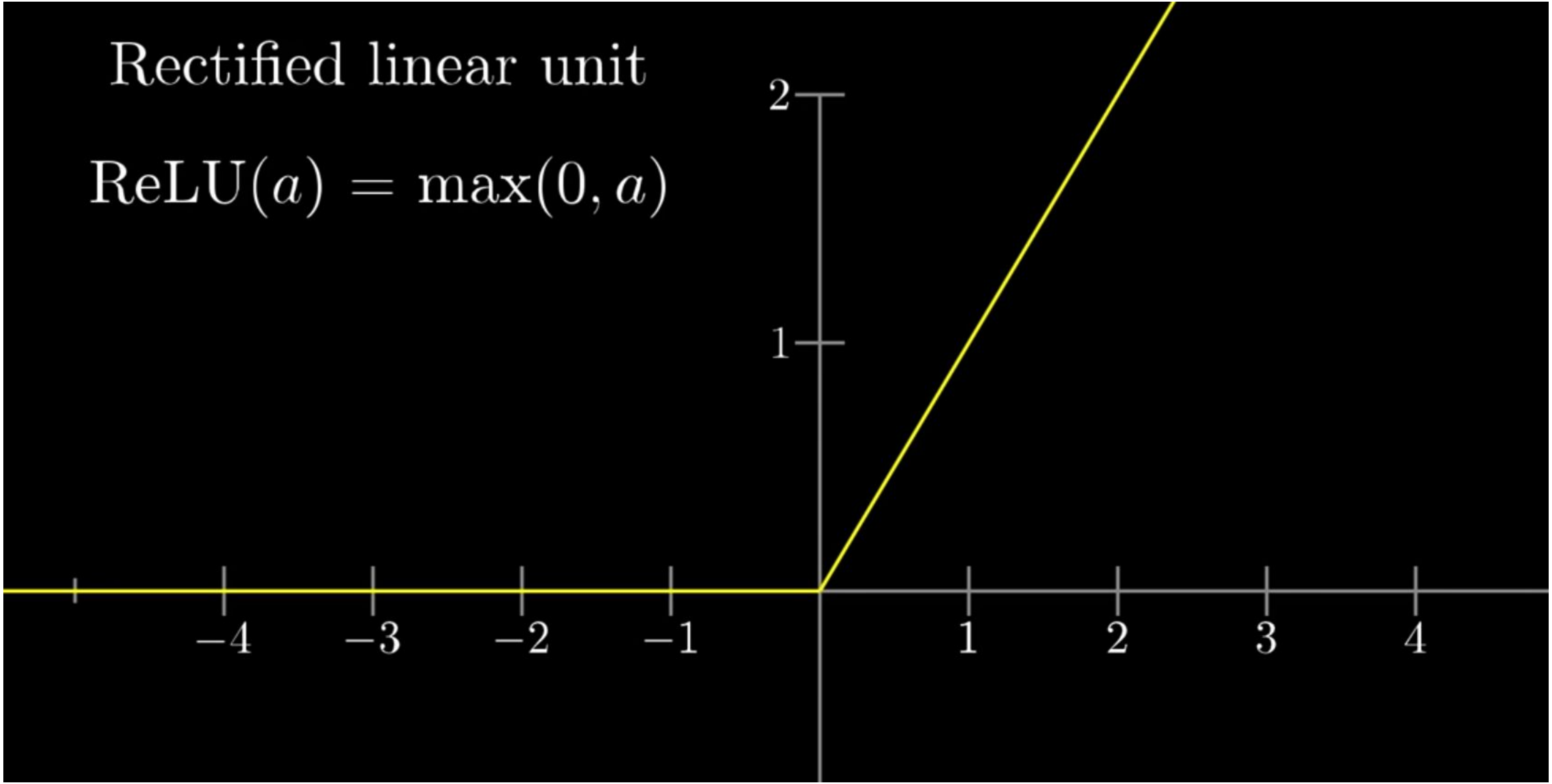
- The output is proportional to the total weighted output.

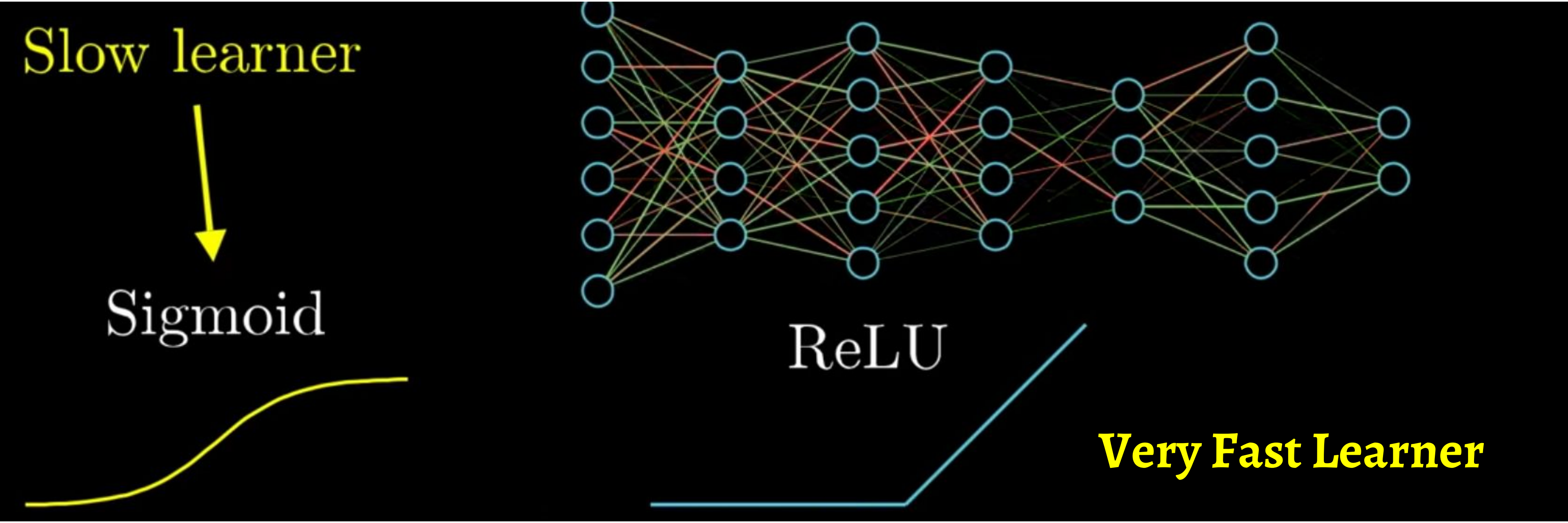


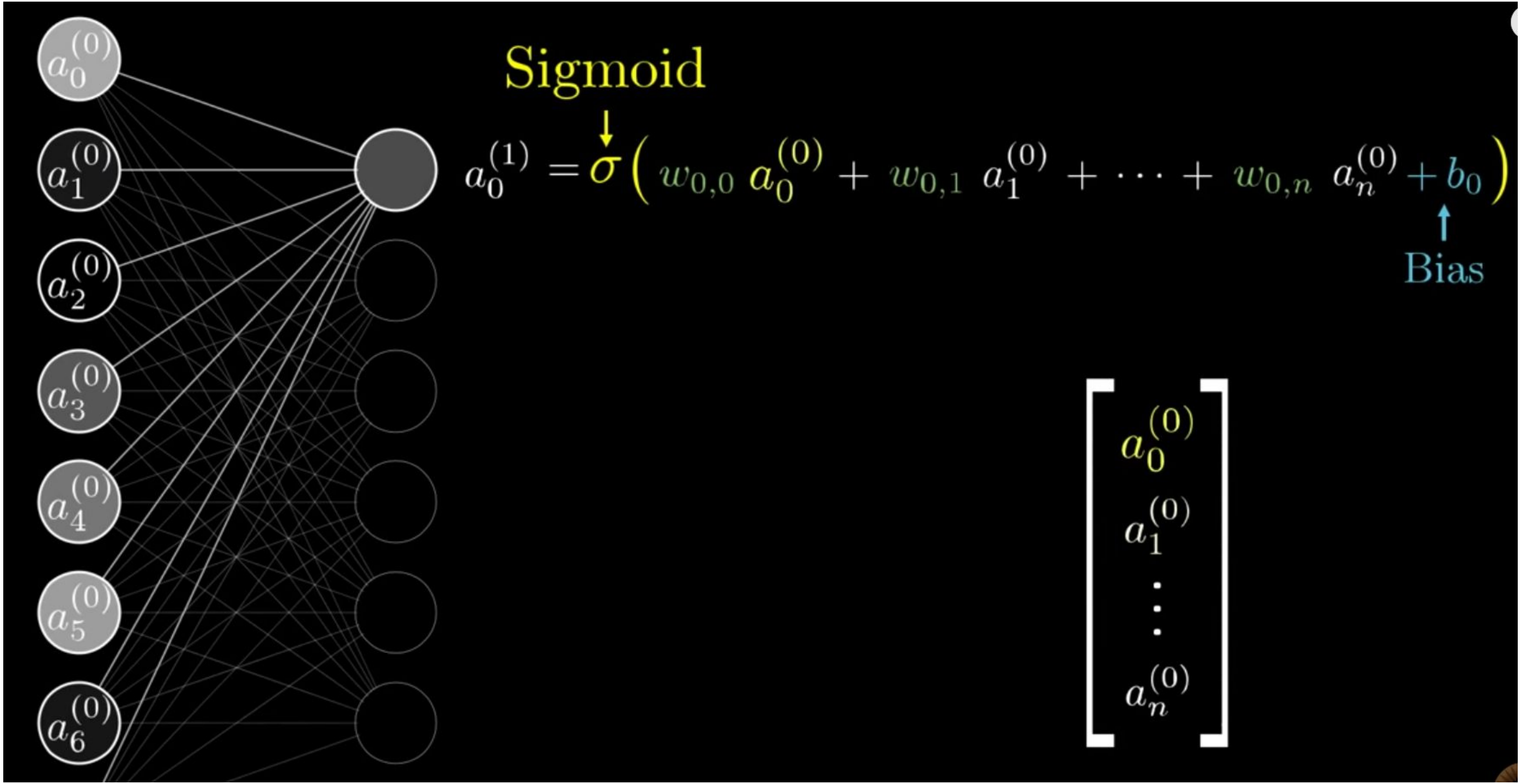
Gaussian

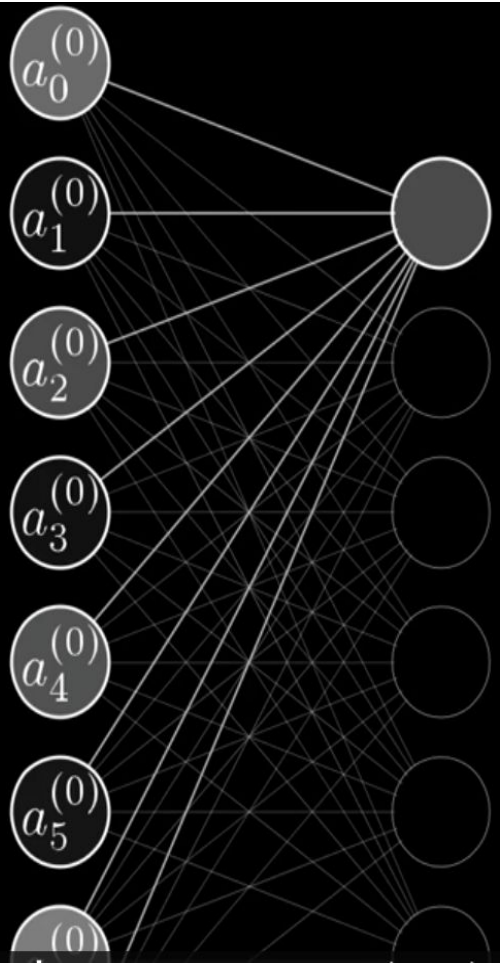
- Gaussian functions are bell-shaped curves that are continuous. The node output (high/low) is interpreted in terms of class membership (1/0), depending on how close the net input is to a chosen value of average.




Rectified linear unit
ReLU(a) = max(0, a)
2
1
−4
−3
−2
−1
1
2
3
4


Slow learner
Sigmoid
ReLU
Very Fast Learner


$a_0^{(0)}$
$a_1^{(0)}$
$a_2^{(0)}$
$a_3^{(0)}$
$a_4^{(0)}$
$a_5^{(0)}$
$a_6^{(0)}$
Sigmoid
$a_0^{(1)} = \sigma\left(w_{0,0}\, a_0^{(0)} + w_{0,1}\, a_1^{(0)} + \cdots + w_{0,n}\, a_n^{(0)} + b_0 \right)$
Bias
$\begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix}$

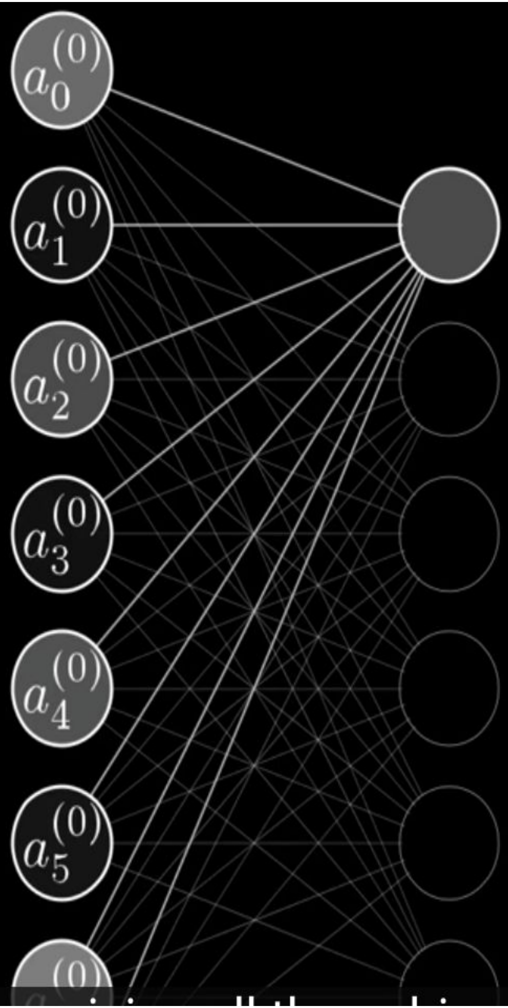


Sigmoid

$$a_0^{(1)} = \sigma\left(w_{0,0}\, a_0^{(0)} + w_{0,1}\, a_1^{(0)} + \cdots + w_{0,n}\, a_n^{(0)} \boxed{+ b_0} \right)$$

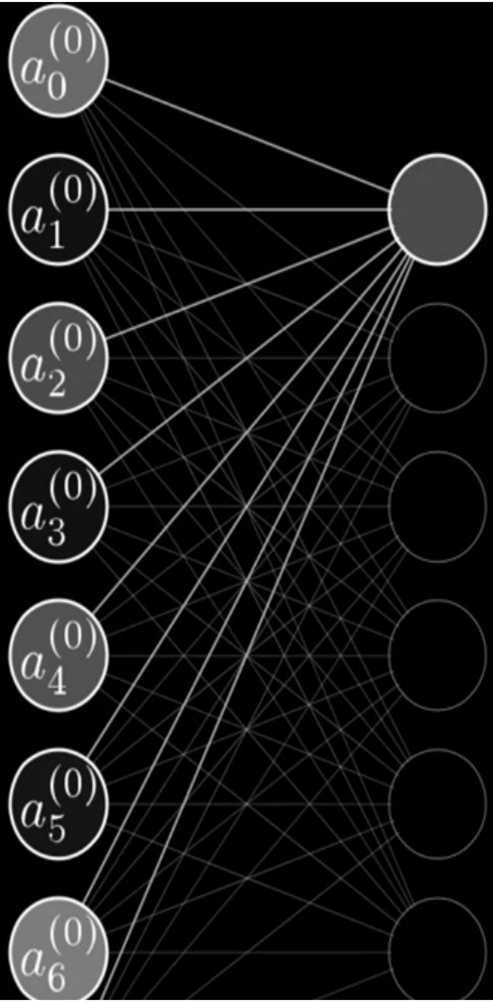
Bias

$$\begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,n} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \cdots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix}$$



$$a_0^{(1)} = \overset{\text{Sigmoid}}{\sigma}\left(w_{0,0}\, a_0^{(0)} + w_{0,1}\, a_1^{(0)} + \cdots + w_{0,n}\, a_n^{(0)} + \underset{\text{Bias}}{b_0} \right)$$

$$\begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,n} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \cdots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}$$



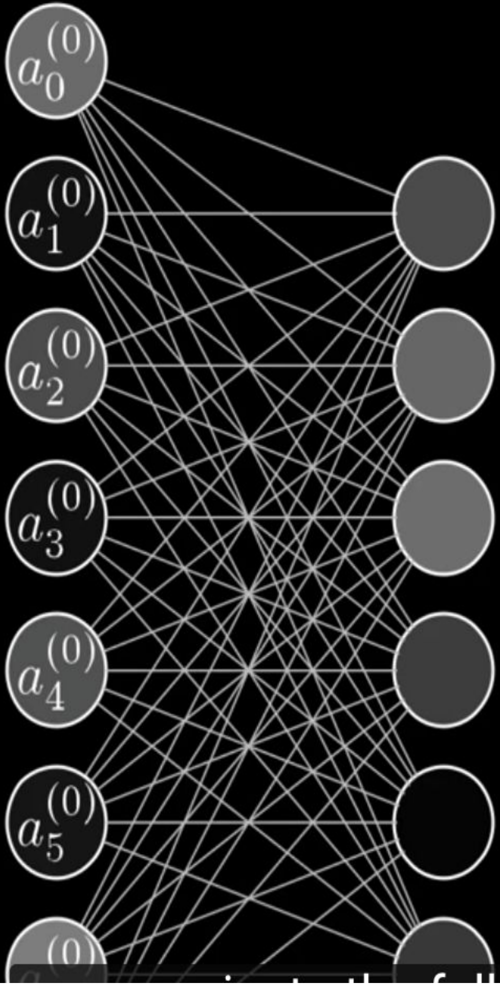
Sigmoid

$$a_0^{(1)} = \underset{\uparrow\ \text{Sigmoid}}{\sigma}\left(w_{0,0}\, a_0^{(0)} + w_{0,1}\, a_1^{(0)} + \cdots + w_{0,n}\, a_n^{(0)} + \underset{\uparrow\ \text{Bias}}{b_0} \right)$$

Bias

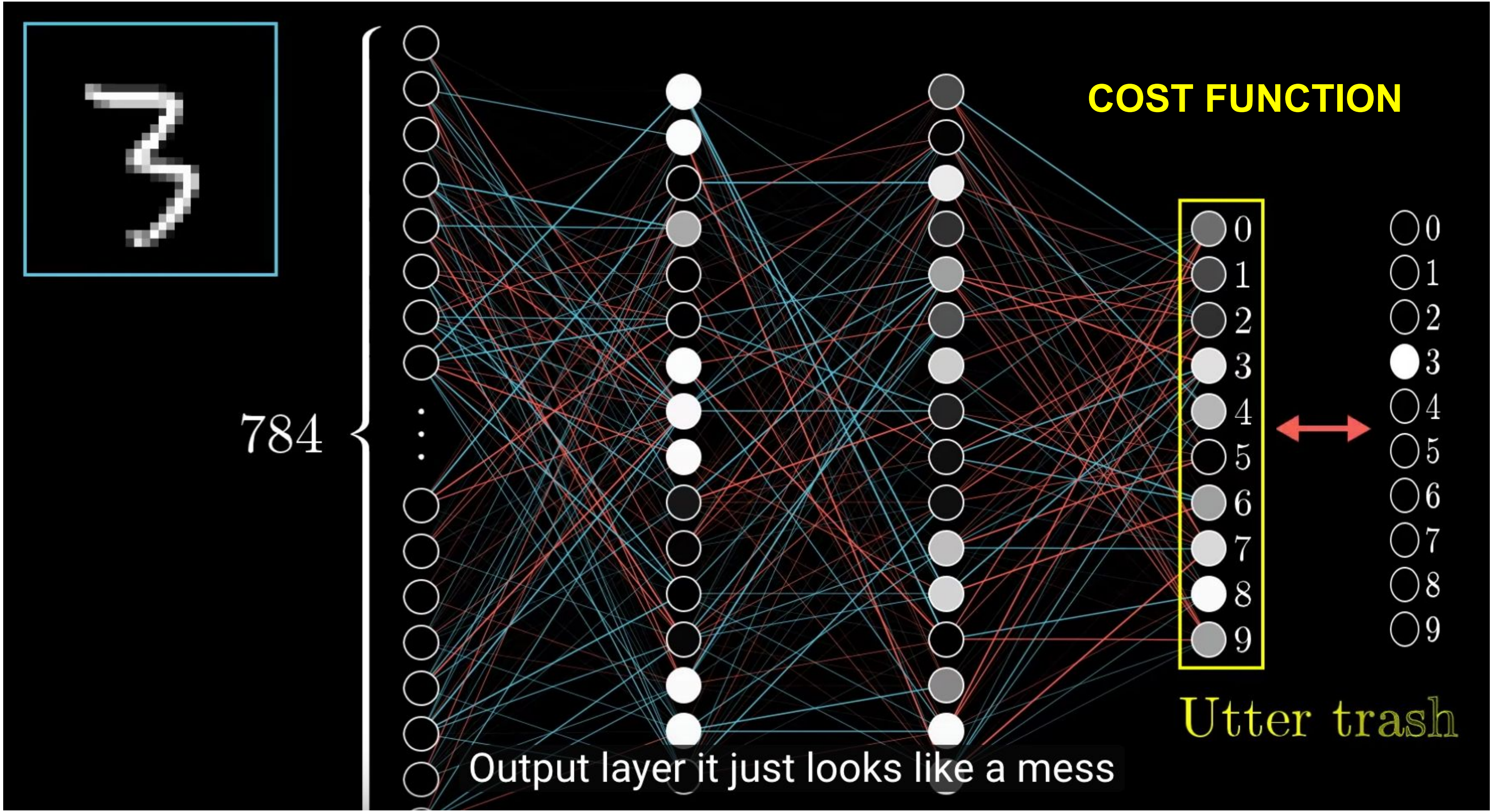
$$\sigma\left(\begin{bmatrix} w_{0,0} & w_{0,1} & \dots & w_{0,n} \\ w_{1,0} & w_{1,1} & \dots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \dots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} \right)$$

$$\sigma\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} \sigma(x) \\ \sigma(y) \\ \sigma(z) \end{bmatrix}$$



$$\mathbf{a}^{(1)} = \sigma\left(\mathbf{W}\mathbf{a}^{(0)} + \mathbf{b}\right)$$

$$\sigma\left(\begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,n} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \cdots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}\right)$$

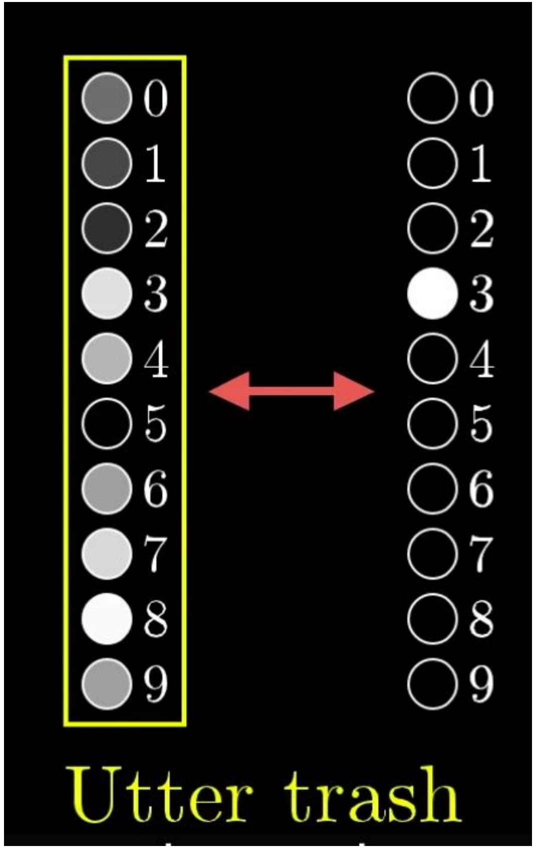


COST FUNCTION
784
0
1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
6
7
8
9
Utter trash
Output layer it just looks like a mess

Least Square Error as Cost Function

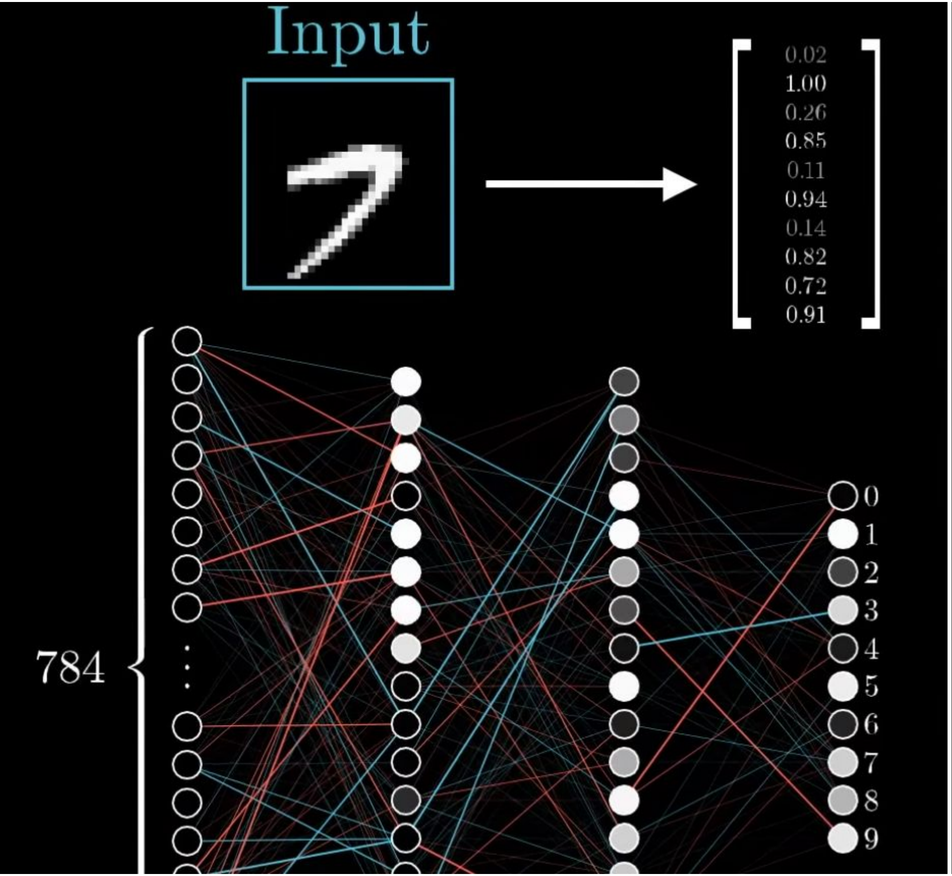
Cost of 

$$
3.37 \left\{
\begin{array}{lcl}
0.1863 & \leftarrow & (0.43 - 0.00)^2 + \\
0.0809 & \leftarrow & (0.28 - 0.00)^2 + \\
0.0357 & \leftarrow & (0.19 - 0.00)^2 + \\
0.0138 & \leftarrow & (0.88 - 1.00)^2 + \\
0.5242 & \leftarrow & (0.72 - 0.00)^2 + \\
0.0001 & \leftarrow & (0.01 - 0.00)^2 + \\
0.4079 & \leftarrow & (0.64 - 0.00)^2 + \\
0.7388 & \leftarrow & (0.86 - 0.00)^2 + \\
0.9817 & \leftarrow & (0.99 - 0.00)^2 + \\
0.3998 & \leftarrow & (0.63 - 0.00)^2
\end{array}
\right.
$$



Cost of 

$$
0.03 \left\{
\begin{array}{lcl}
0.0006 & \leftarrow & (0.02 - 0.00)^2 + \\
0.0007 & \leftarrow & (0.03 - 0.00)^2 + \\
0.0039 & \leftarrow & (0.06 - 0.00)^2 + \\
0.0009 & \leftarrow & (0.97 - 1.00)^2 + \\
0.0055 & \leftarrow & (0.07 - 0.00)^2 + \\
0.0004 & \leftarrow & (0.02 - 0.00)^2 + \\
0.0022 & \leftarrow & (0.05 - 0.00)^2 + \\
0.0033 & \leftarrow & (0.06 - 0.00)^2 + \\
0.0072 & \leftarrow & (0.08 - 0.00)^2 +
\end{array}
\right.
$$



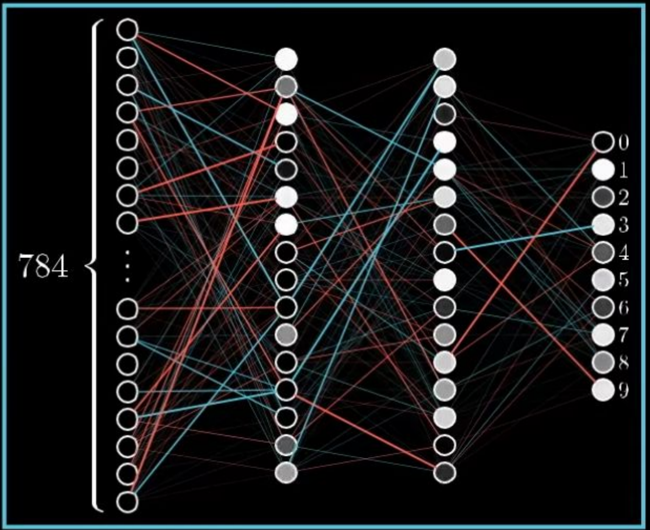
Neural network function

Input: 784 numbers (pixels)

Output: 10 numbers

Parameters: 13,002 weights/biases

Input



Cost: 5.4

Cost function

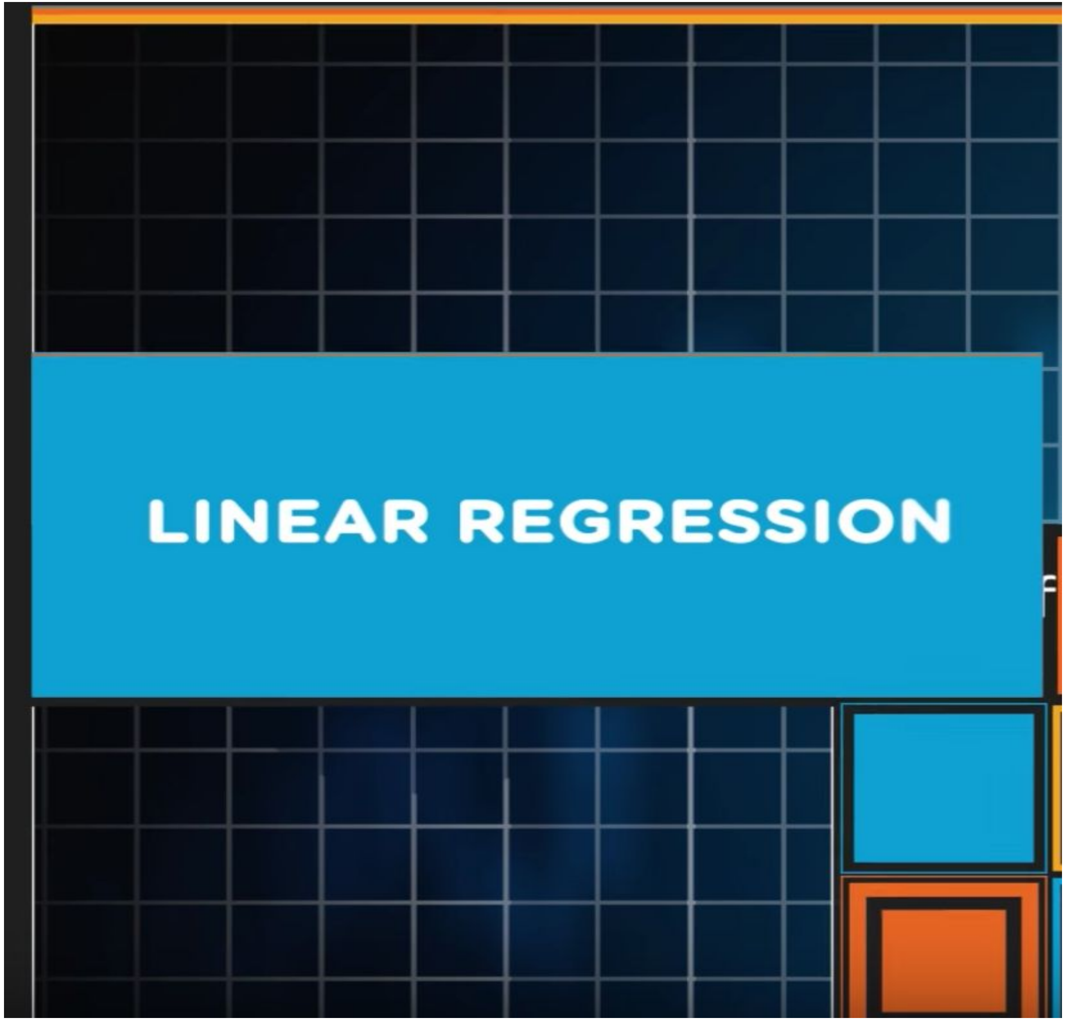
Input: 13,002 weights/biases

Output: 1 number (the cost)

Parameters: Many, many, many training examples

( ,7)

That's a lot to think about


LINEAR REGRESSION

Profit Estimation of a Company

A Venture Capital firm is trying to understand which companies should they invest

Venture Capital firm

Use Case #1

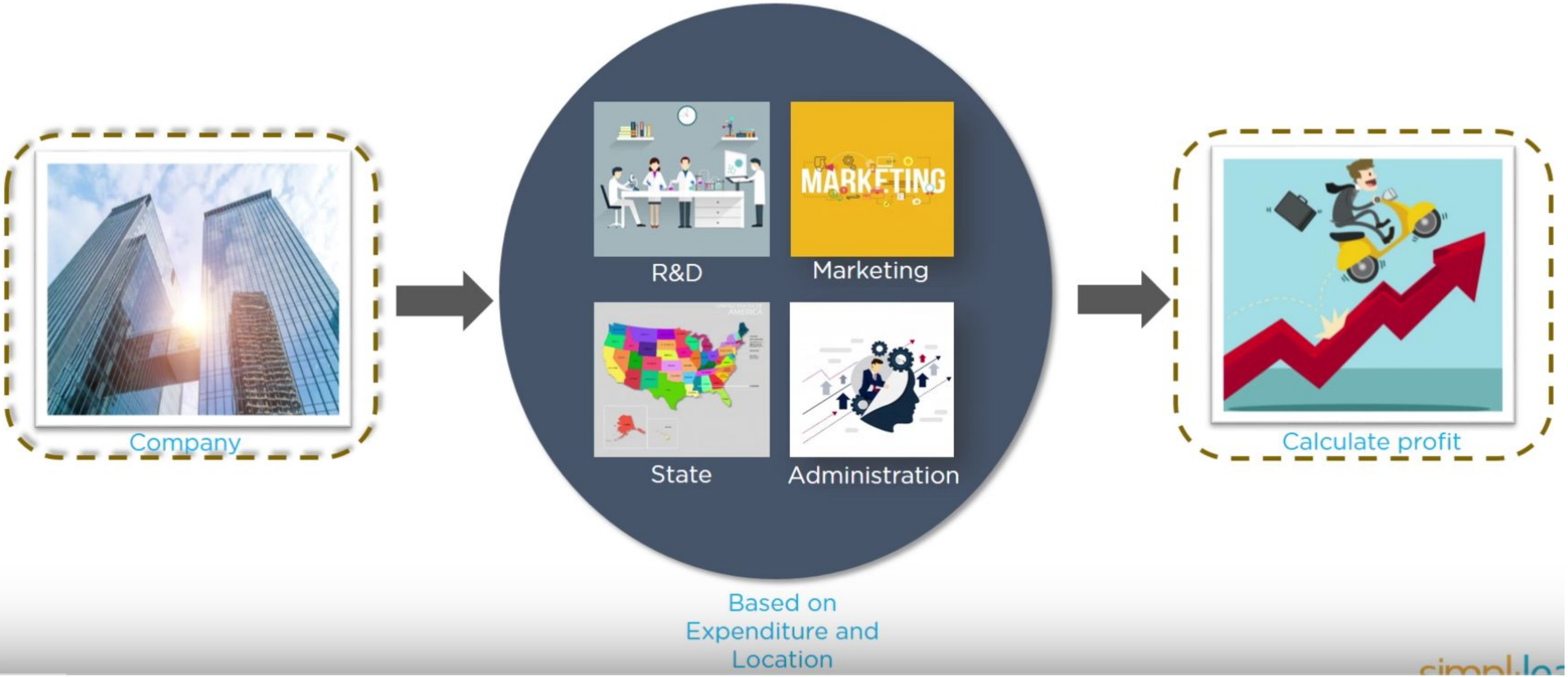
Profit Estimation of a Company

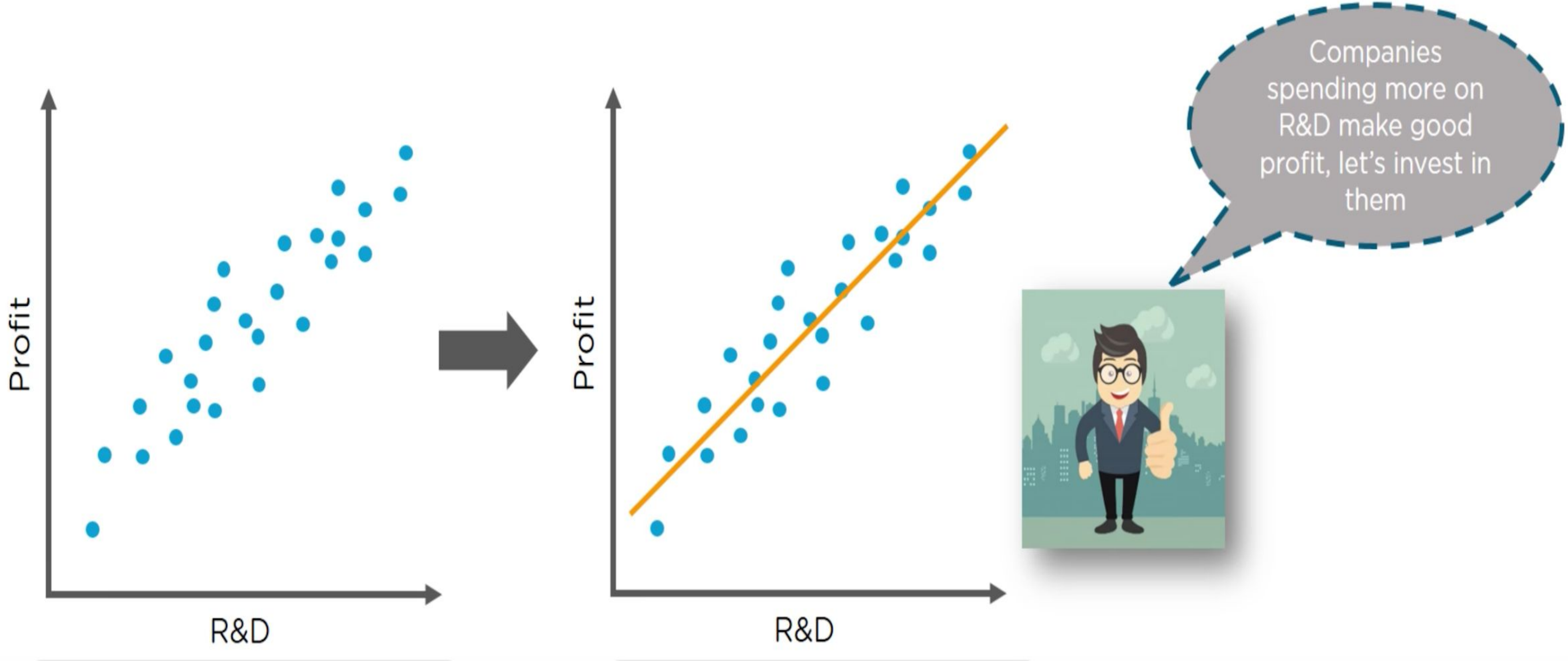
Decide companies to invest

Predict the profit companies make

Based on companies expenses

Profit Estimation of a Company





Plot: Profit Earned Based on R & D Expenditure

Plot: Prediction Line to estimate profit

Crop Field

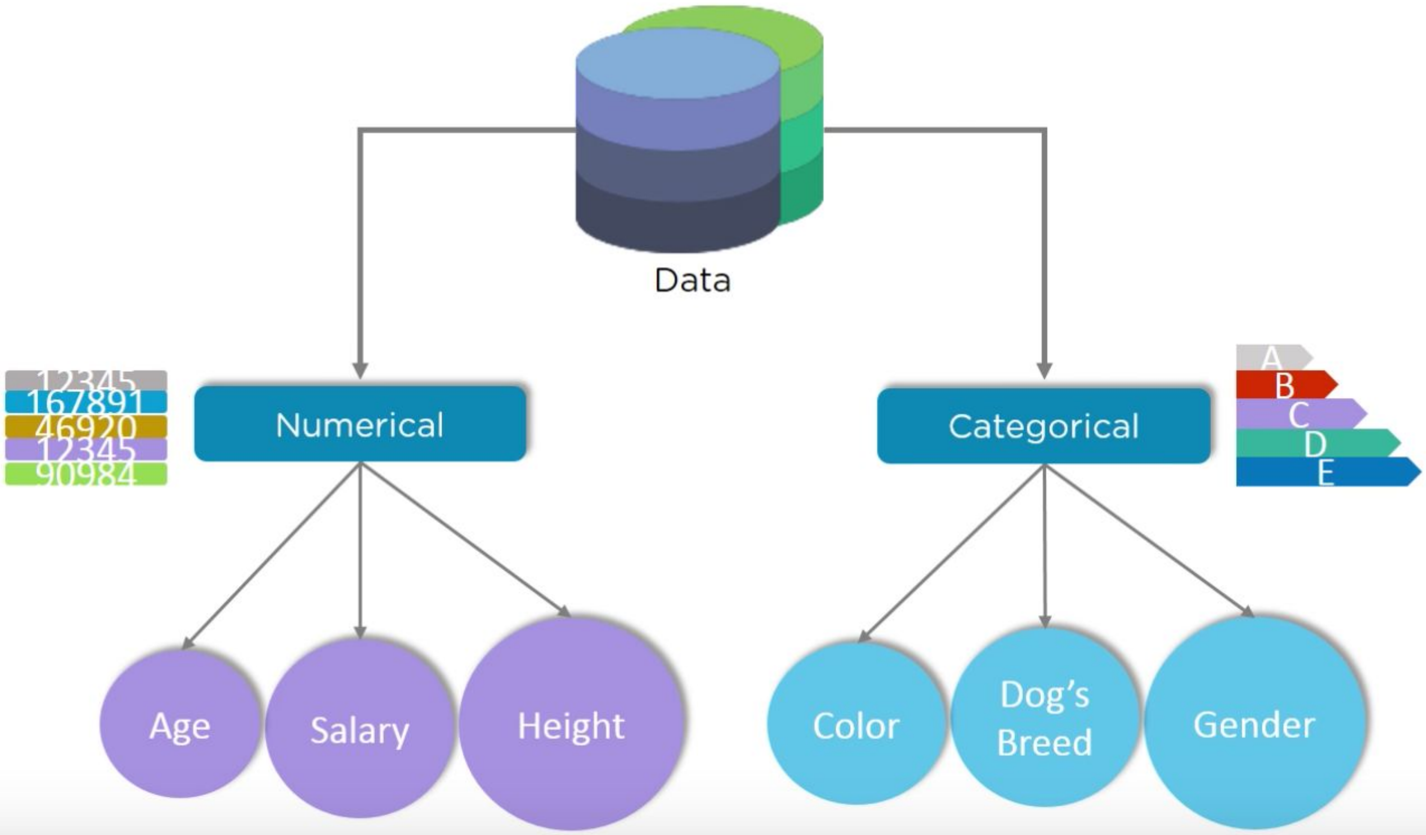
Based on Rainfall

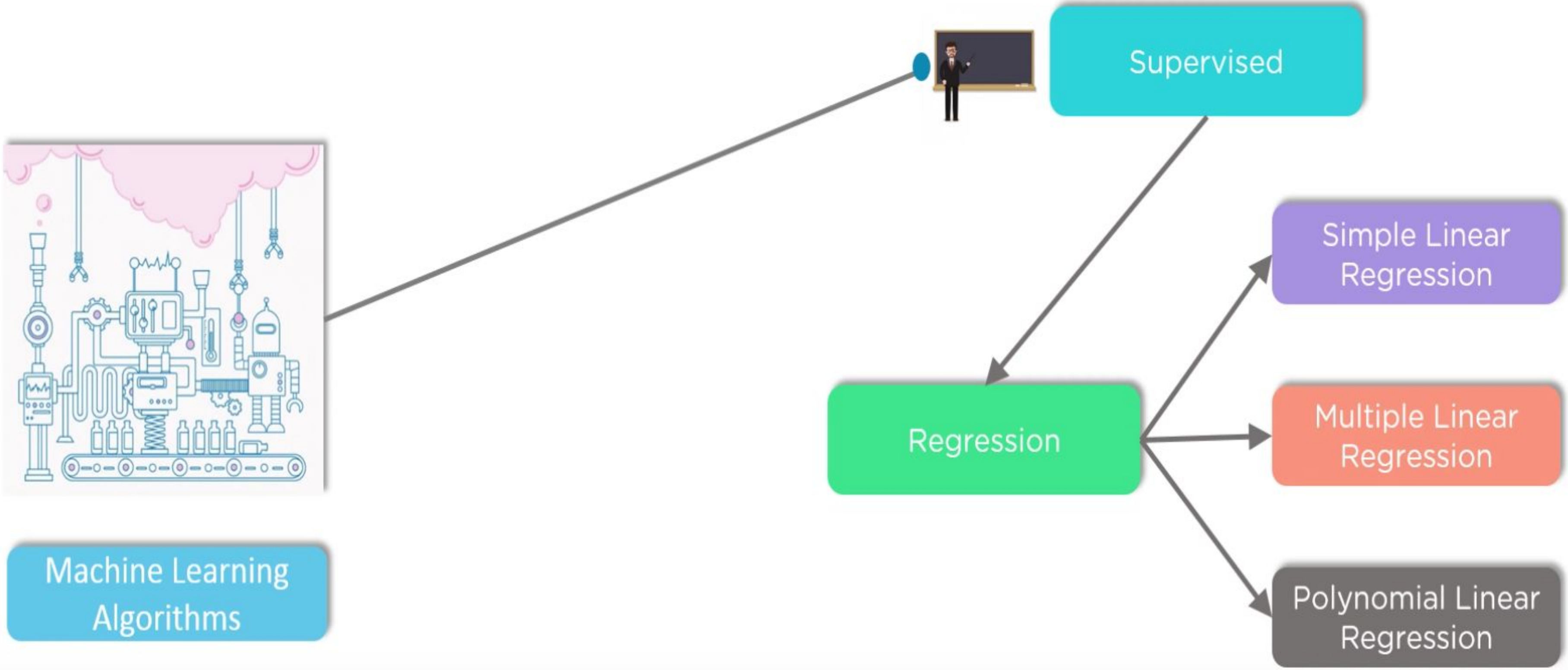
Predict crop yield

Based on the amount of rainfall, how much would be the crop yield?

Use Case #2

Numerical and Categorical Values




Supervised
Regression
Simple Linear Regression
Multiple Linear Regression
Polynomial Linear Regression
Machine Learning Algorithms

Applications of Linear Regression

Score Prediction

To predict the number of runs a player would score in the coming matches based on previous performance

Applications of Linear Regression

Economic Growth

Used to determine the Economic Growth of a country or a state in the coming quarter, can also be used to predict the GDP of a country

Applications of Linear Regression

Product price

Can be used to predict what would be the price of a product in the future

Applications of Linear Regression

Housing sales

To estimate the number of houses a builder would sell and at what price in the coming months

Linear Regression is a statistical model used to predict the relationship between independent and dependent variables.

Examine 2 factors

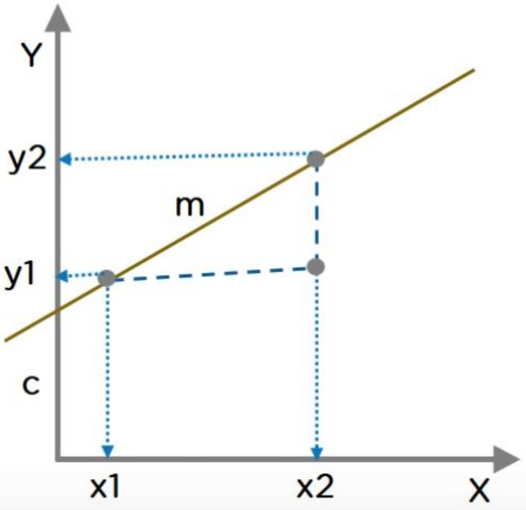
1. Which variables in particular are significant predictors of the outcome variables?

2. How significant is the Regression line to make predictions with highest possible accuracy

Regression Equation

The simplest form of a simple linear regression equation with one dependent and one independent variable is represented by:

$$y = m * x + c$$



y ---> Dependent Variable

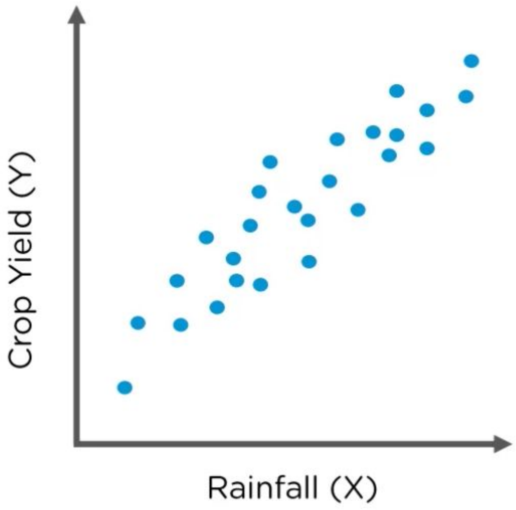
x ---> Independent Variable

m ---> Slope of the line

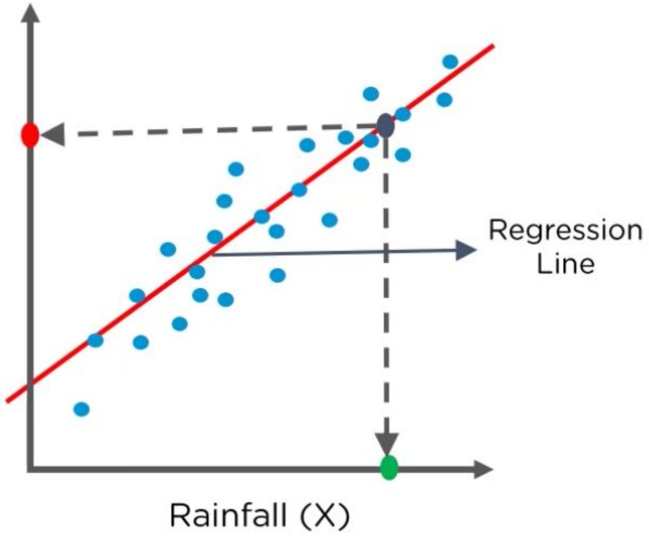
$$m = \frac{y2 - y1}{x2 - x1}$$

c ---> Coefficient of the line

Prediction using the Regression line

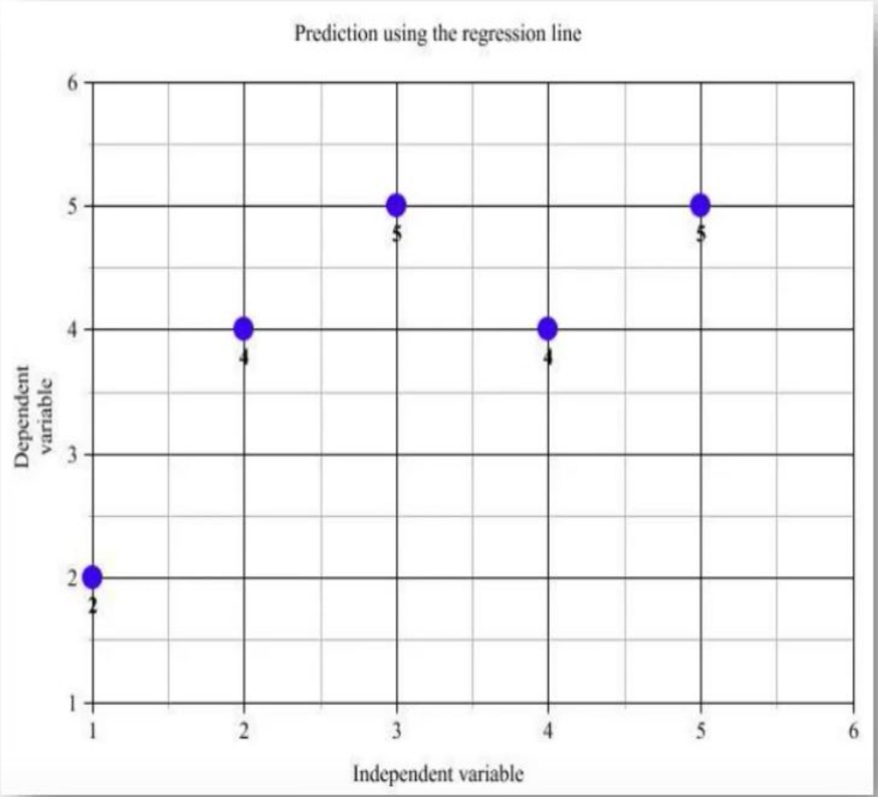


Plotting the amount of Crop Yield based on the amount of Rainfall



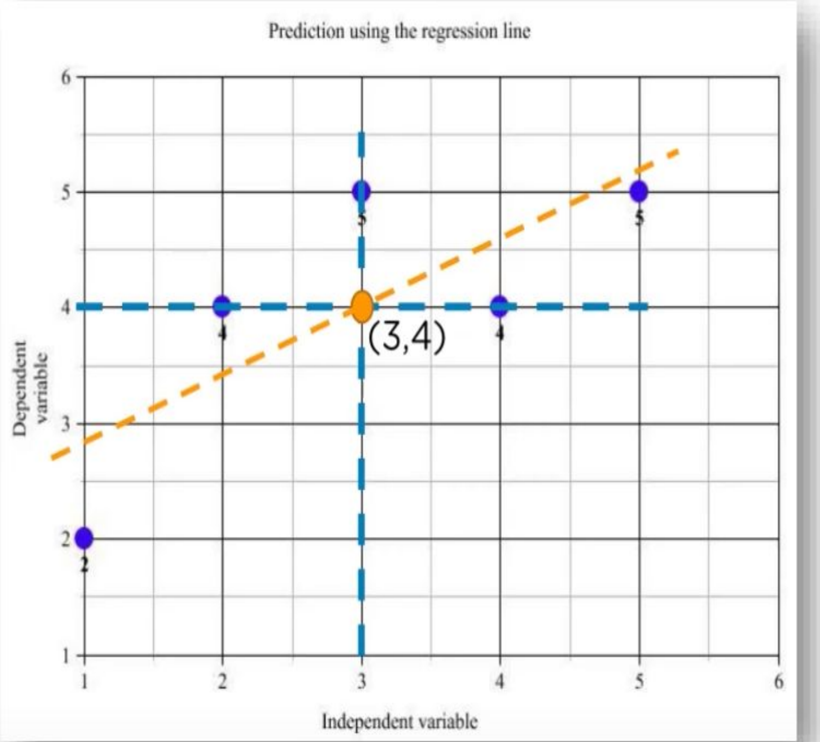
Red point on the Y axis is the amount of Yield you can expect for some amount Rainfall (X) represented by Green dot

Independent variable X	Dependent variable Y
1	2
2	4
3	5
4	4
5	5



Plotting the data points

Independent variable X	Dependent variable Y
1	2
2	4
3	5
4	4
5	5
Mean: 3	4



Regression line

Drawing the equation of the Regression line

X	Y	(X^2)	(Y^2)	$(X*Y)$
1	2	1	4	2
2	4	4	16	8
3	5	9	25	15
4	4	16	16	16
5	5	25	25	25
$\sum = 15$	$\sum = 20$	$\sum = 55$	$\sum = 86$	$\sum = 66$

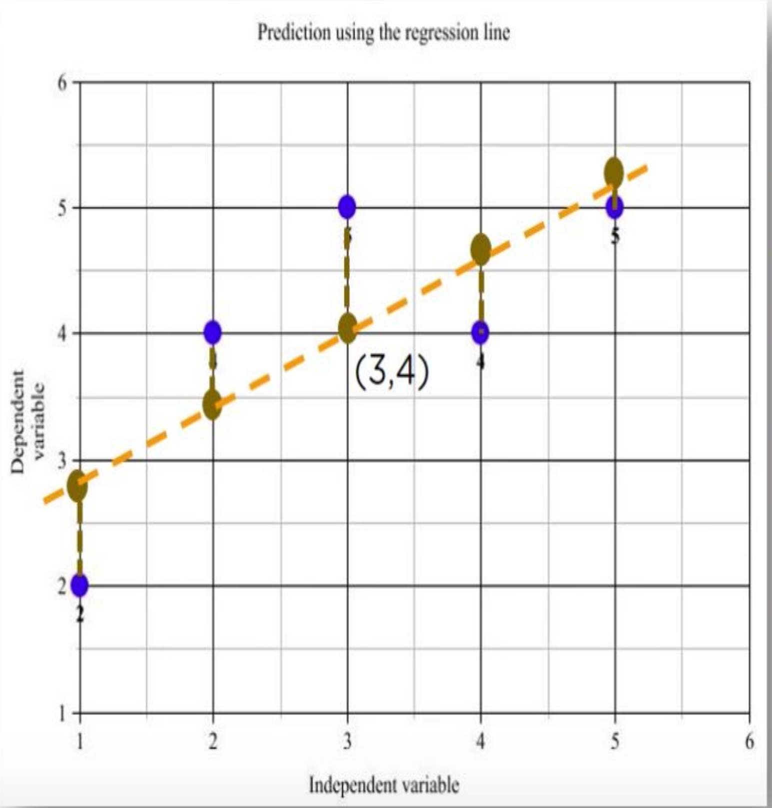
$$Y = m * X + c$$
$$= 0.6 * 3 + 2.2$$
$$= 4$$

Linear equation is represented as $Y = m * X + c$

$$m = \frac{((n * \sum(X*Y)) - (\sum(X) * \sum(Y))}{((n * \sum(X^2)) - (\sum(X)^2)} = \frac{((5*66)-(15*20))}{((5*55))-(225)} = 0.6$$

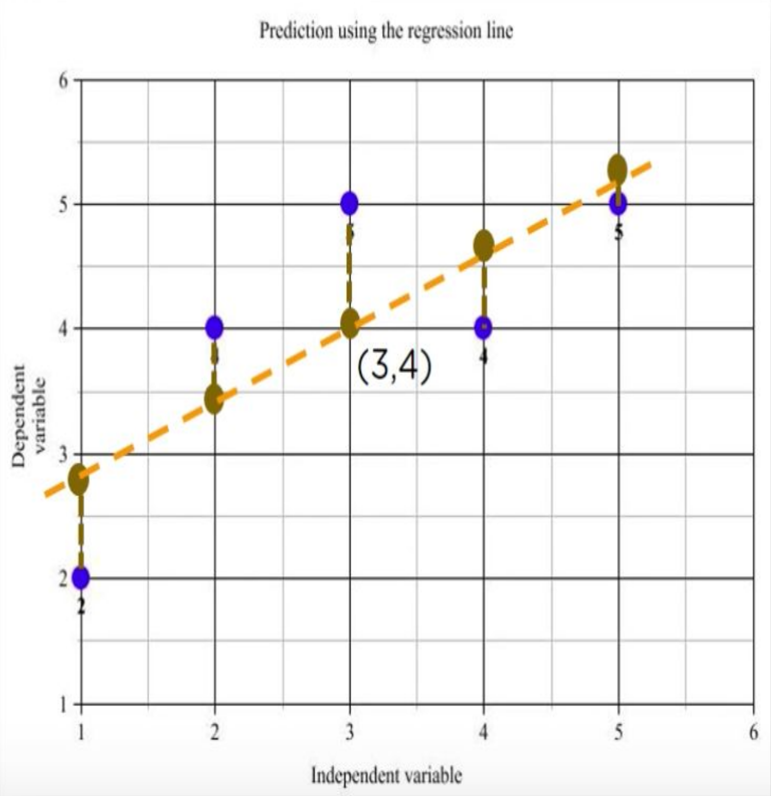
$$c = \frac{((\sum(Y) * \sum(X^2)) - (\sum(X) * \sum(X*Y)}{((n * \sum(X^2)) - (\sum(X)^2)} = 2.2$$

225 = 15 x 15



Y_{pred}
$Y=0.6*1+2.2=2.8$
$Y=0.6*2+2.2=3.4$
$Y=0.6*3+2.2=4$
$Y=0.6*4+2.2=4.6$
$Y=0.6*5+2.2=5.2$

Here the blue points represent the **actual Y values** and the brown points represent the **predicted Y values**. The distance between the actual and predicted values are known as ***residuals or errors***. The best fit line should have the least sum of squares of these errors also known as ***e square.***

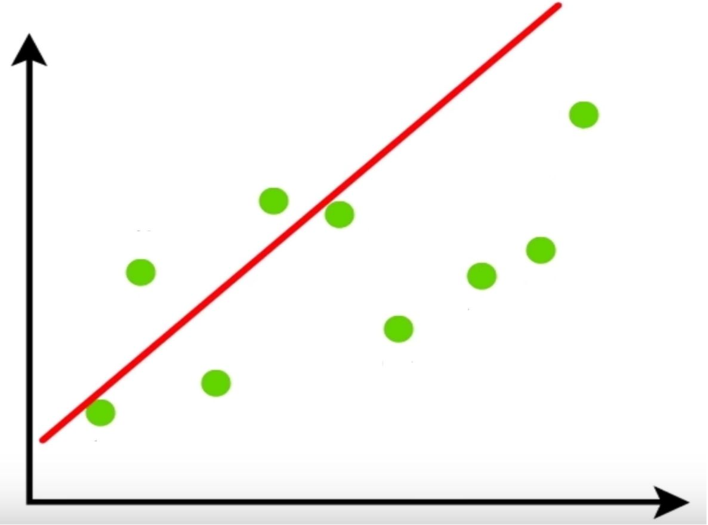


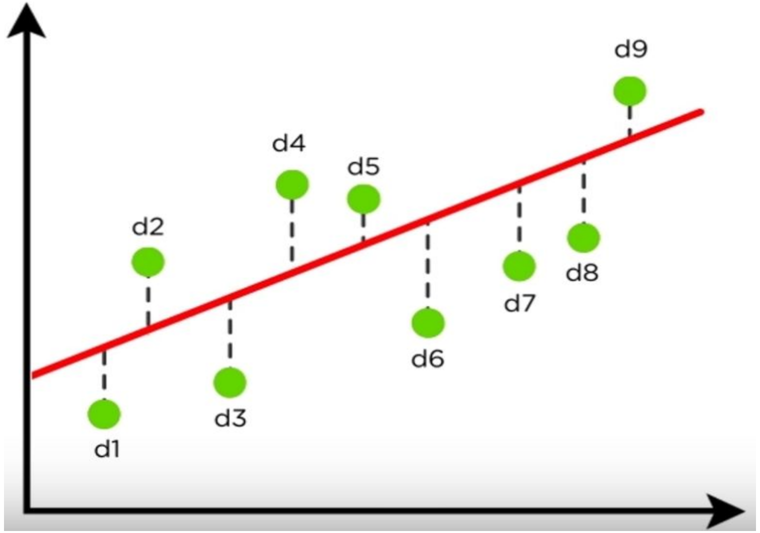
X	Y	Y_{pred}	$(Y-Y_{pred})$	$(Y-Y_{pred})^2$
1	2	2.8	-0.8	0.64
2	4	3.4	0.6	0.36
3	5	4	1	1
4	4	4.6	-0.6	0.36
5	5	5.2	-0.2	0.04

$\sum = 2.4$

The sum of squared errors for this regression line is 2.4. We check this error for each line and conclude the best fit line having the least e square value.

Sum of “Least Square Error”



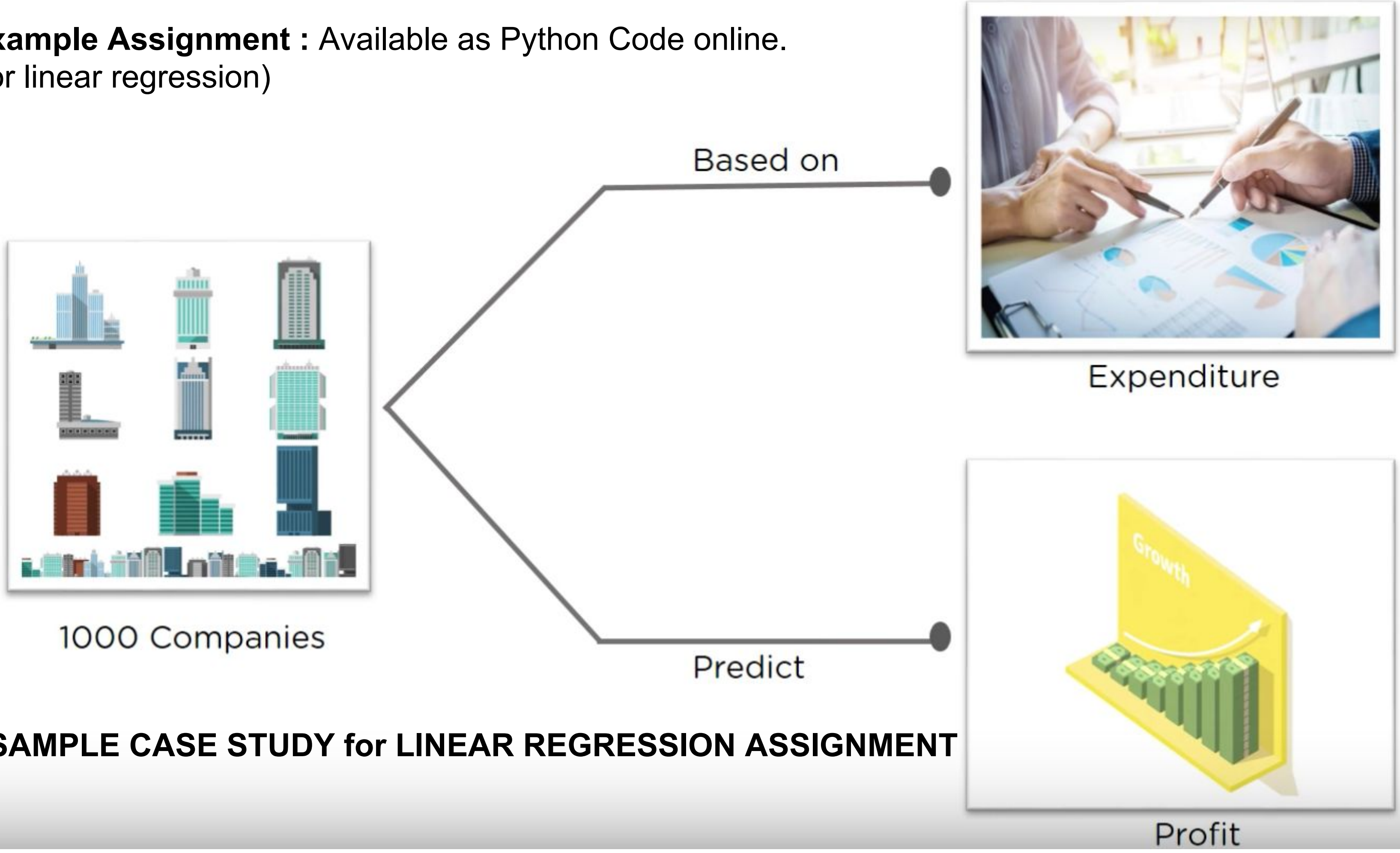


$$D = D_1^2 + D_2^2 + D_3^2 + D_4^2 + D_5^2 + D_6^2 + D_7^2 + D_8^2 + D_9^2$$

Finding the Best fit line

Minimizing the Distance: There are lots of ways to minimize the distance between the line and the data points like Sum of Squared errors, Sum of Absolute errors, Root Mean Square error etc.

Example Assignment : Available as Python Code online.
(for linear regression)

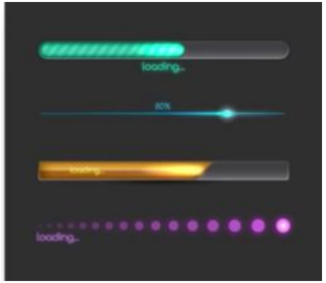


A SAMPLE CASE STUDY for LINEAR REGRESSION ASSIGNMENT

What is expected out of your Assignment ?

(Expected Steps depicted. You are to demonstrate Step #1 to Step #10, out of which Step #4 and Step #5 are optional, rest mandatory)

1. Import the libraries:



2. Load the Dataset and extract independent and dependent variables:

3. Plot the loaded data set after extracting dependant & independent variables

4. Encoding Categorical Data:

5. Avoiding Dummy Variable Trap:

Do Step #4 and Step #5 **ONLY** if required

6. *Splitting the data into Train and Test set:*

7. *Fitting Multiple Linear Regression Model to Training set:*

8. *Predicting the Test set results:*

9. Calculating the Coefficients and Intercepts:

10. Evaluating the model: