Check 20/09/2021

Review Linear vs Multiple Regression

Exp (x)	vs	Salary (y)
2		50,000
4		35,000
6		60,000
7		65,000
10		70,000

$y = b_0 + b_1 x_1$

b_0 = intercept

b_1 = Slope.

(x) Age, Size, Location — (y) Price

(iv) (dv)

Why?

R&D	Admin	Market Spend	State: New York	State: California	State: Florida	Profit
5	9	3	1	0	0	
6	2	4	0	1	0	
10	1	6	0	0	1	

$D_3 = -(D_1 + D_2) = -1$

$P + \hat{P} = 1$

$D_3 = D_1 + D_2 = ?$

$D_1 + D_2 = D_3$

$D_2 = 1 - D_1$

$= 1 - 1 = 0$

$0 = 0$

$D_2 = 1 - D_1$

$= 1 - 0 = 1$

State (numeric data)

D1 NY	D2 CA	D3 FL
1	0	0
0	1	0
0	0	1
0	1	0
1	0	0

R (1) ① ②

D1 NY	D2 CA
1	0
0	1
1	0
0	1
1	0

$D_1 = D_2$ ✗

$D_1 = -D_2$ ✗

$D_1 = 1 + D_2$ ✓

$D_1 = 1 - D_2$ ✓

$D_2 = 1 - D_1$

Fri → 24/09/2021

ANN → Neuron

Recall in Lin Reg:

$y = b_0 + b_1 x_1$

$\Rightarrow y = c + mx$

$\Rightarrow y = mx + b$

$y = b_0 + b_1 x_1 + b_2 x_2 \ldots b_n x_n$

Data (manipulate)

house price → Linear Regression, SVM, ANN

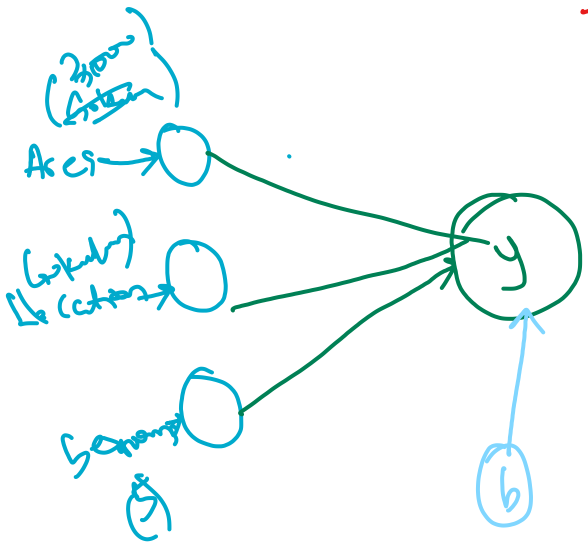
Expensive — Truth, 2crores

1) Gokulam — 3 Crores

2) Vijaynagar — 2.8 crore

3) Kuvempunagar — 1.5

y = output → less than expected value



$\Rightarrow \sum_{i=1}^{n} b_i x_i + b_0$

GIGO ⇒ Good data in, Good model out.

Quality

Label/output

3

3000

0

Note: All ML models revolves around these steps:

Step 1: Collect Data (All you need is quality data)

Step 2: Prepare the input data (cleanse, format etc)

Step 3: Analyze the input data (visualize, find features)

Step 4: Train the data

Step 5: Testing → h5

Step 6: Productionalize (Deployment)

Train (400 epoch) mnist

LC → 16hrs

0
1
2
3

Step 1:

Data: What is data? → Data

Datum Datum:

What can you say about these data?

- Textual (.csv, .arff, .xlsx,)
- Image (.png, .svg, .jpg, .Tiff, .jpeg, .bmp, .dicom)
- Audio (.mp3, .ogg,)
- Video (.mp4, mpeg)
- Signal (

Are heterogeneous and distributed sources → (on a single source) Local, Remote

As a Single Source → Connect to location (online) online

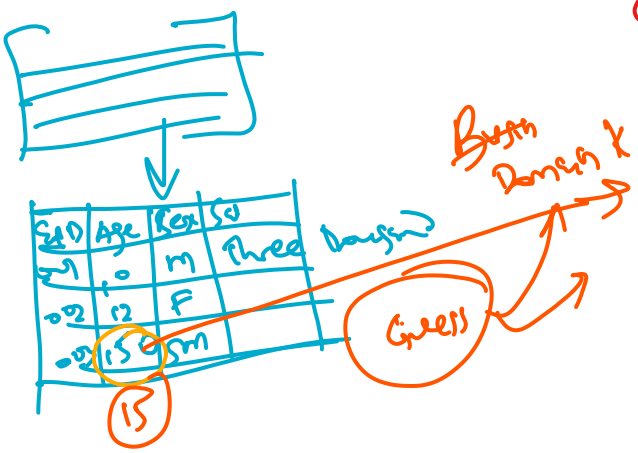
Offline vs online

Step 2: Cleansing (~~with bad water~~)

Duplicate data,

Look for outliers

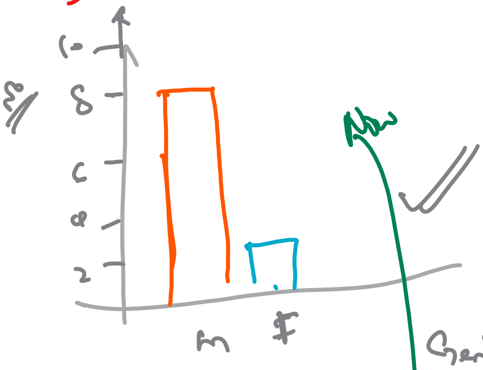
Missing values



80% of DS ~~fate~~ time revolves around what?

↳ Data cleaning?

Shown (visualization) vs Explaining



Distribute

D => Males

□ = Genders

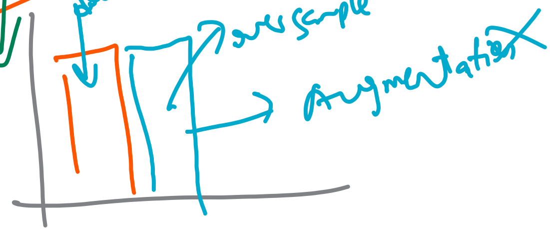
Data Imbalance ~~fitting~~ Problem

8 : 2

4 : 1

80% : 20%

Male > Female



Why steps 1–3: Because it requires "human-in-the-loop"

⇒ more time from the DE

Data Analyst vs Data Engineer vs Data Scientist

Recall: $y = b_0 + b_1 x_i$

How do we ~~recreate~~ simulate the human brain

True value: y

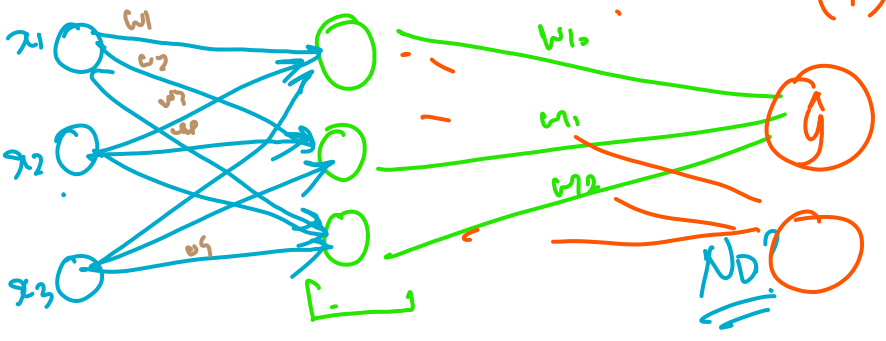
Cat vs Dog

1 X 0

Probability:

80% : 20

Levels & layers in a neural:



hidden layer (3 cells)

(A) y = actual output

(P) $\hat{y}$ = predicted output

A = 1.0

P = 0.6

(loss function)

Calculate the loss or Error

Goal: minimize the loss (~~Entropy~~)

$\equiv$ maximize the accuracy (~~Accuracy~~)

Error = A − P or P − A

1.0 − 0.6 = 0.4

0.6 − 1.0 = −0.4

ε = |A − P| or |P − A| $\Rightarrow (A-P)^2 \equiv (P-A)^2$ (MSE)

$(0.4)^2$ $(-0.4)^2$

0.16

0.16 / 2

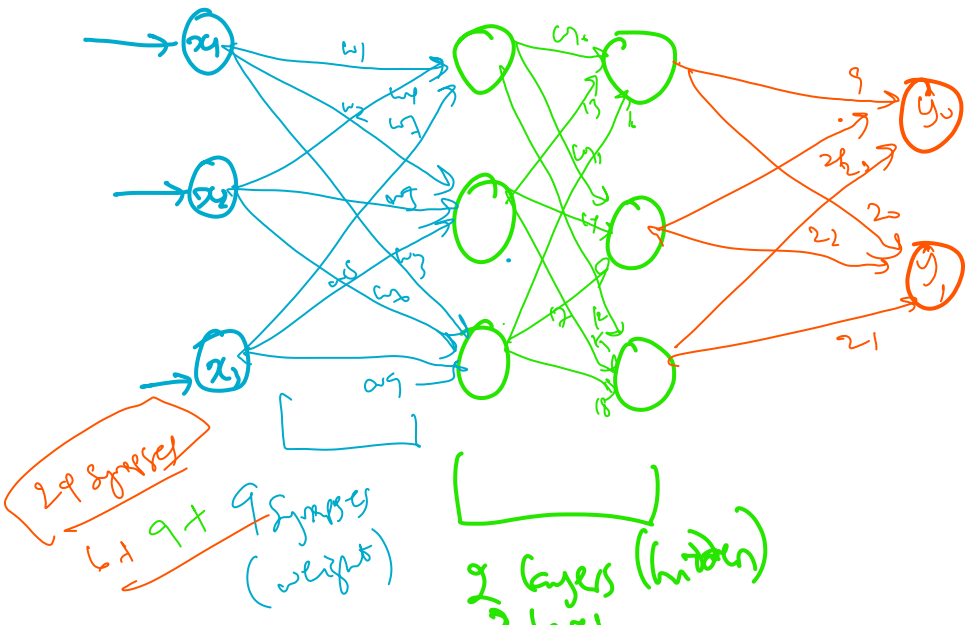
$\Rightarrow$ 0.08 ✓

$(A-P)^2$

$$\text{Error} = \frac{1}{2}\sum_{i=1}^{i}(A-P)^2$$ ✓

Convergence faster.

Explain Back propagation



24 synapses

i.e 9 + 9 synapses (weights)

2 layers (hidden)

3 level

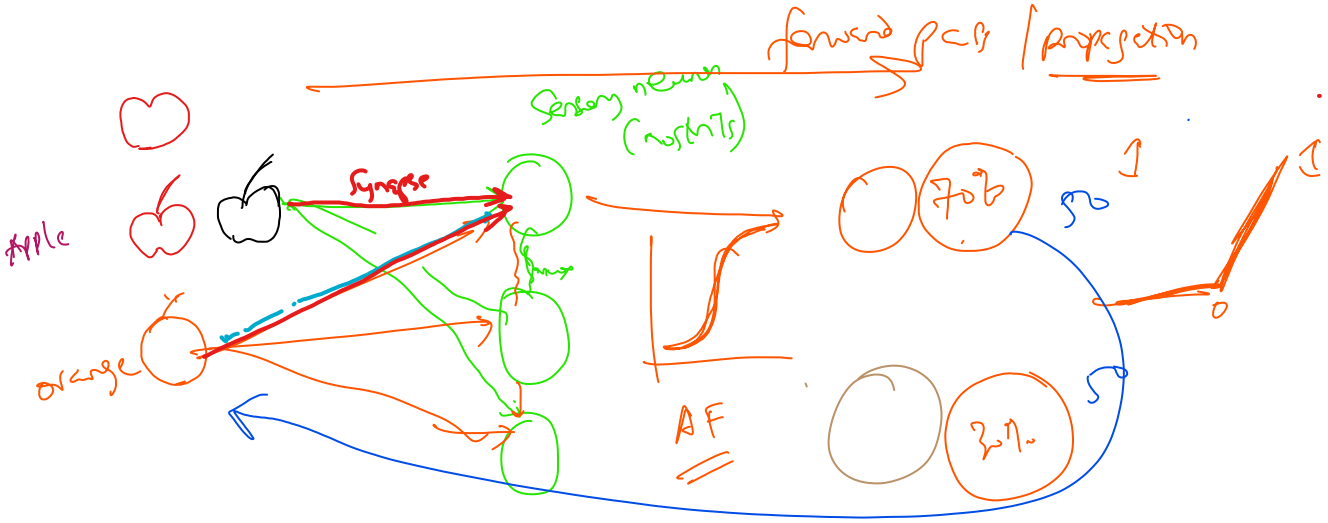
Error = $\varepsilon_{y_0} + \varepsilon_{y_1}$

$\Rightarrow \frac{(A-P)^2}{2} + \frac{(A-P)^2}{2}$

$\Rightarrow \sum(\varepsilon_{y_0} + \varepsilon_{y_1})$

$\frac{1}{2}\sum_{i=1}^{i}\varepsilon_{y_i}$

where $x = 1, 2, \ldots$



Solution

forward pass / Propagation

Input → hidden layer → output

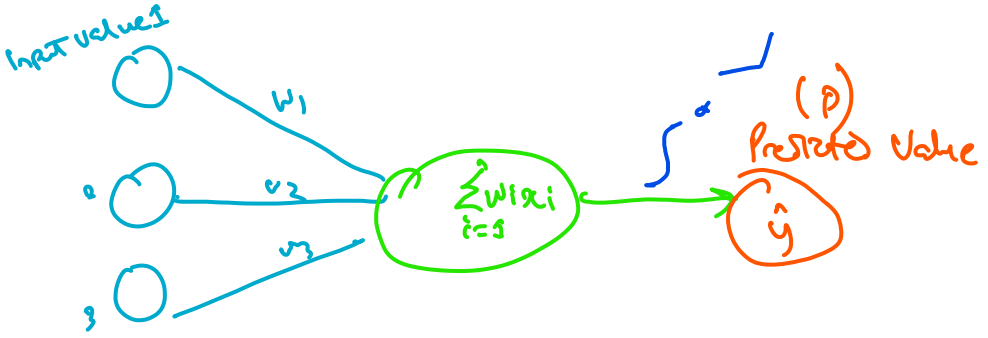
Backward pass / backpropagation

output → hidden → input

updated weight

Sun → 26/09/2021

How does a Neural Network learn?

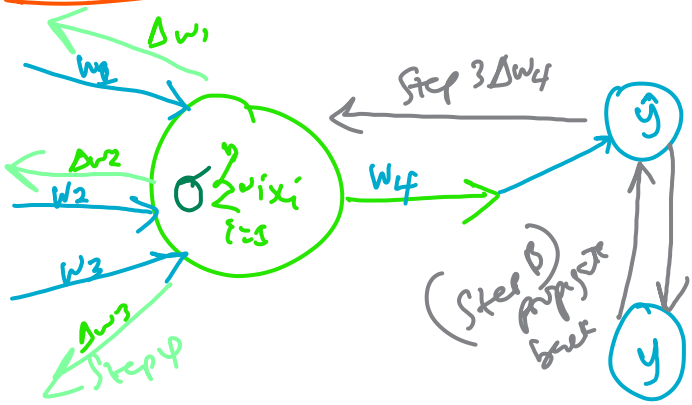


Fiji: A Single Perceptron

(Franklin Rosenblatt)

In 1958

Learning objective



σ = Mysterious ingredient

(Biryani → veg)

↔

KFC

↓

Taste X

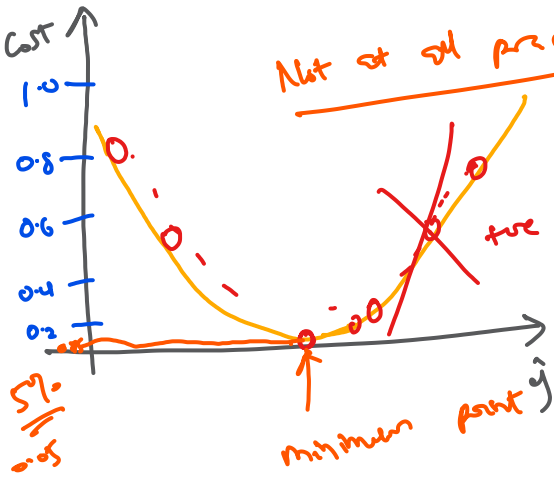
σ = Activation function

Relu
Softmax
} Activation function

Adam → Optimizers

SGD →

Gradient Descent vs Obj function: minimize



$(C = \frac{1}{2}(\hat{y}-y)^2)$

$\frac{\partial y}{\partial x}$

(+ve) less

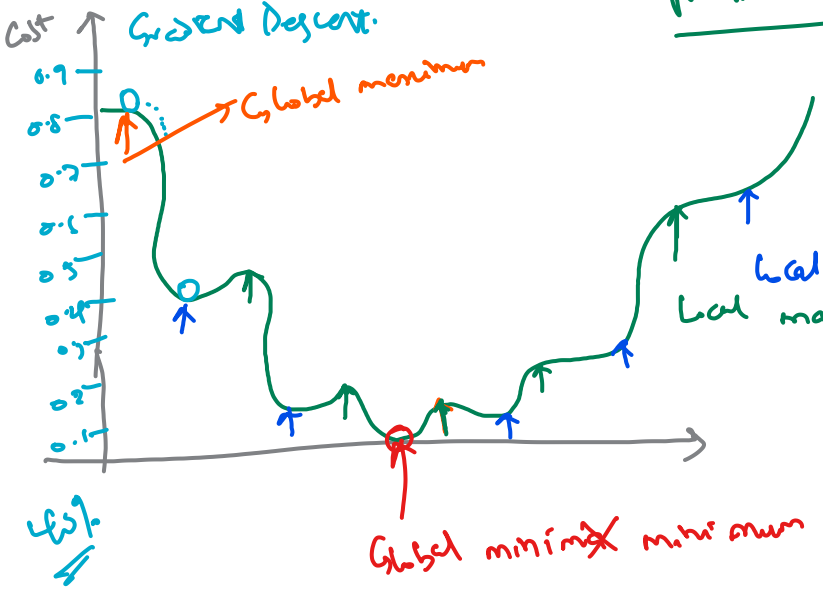
(+ve) more

Neural Network Training problem

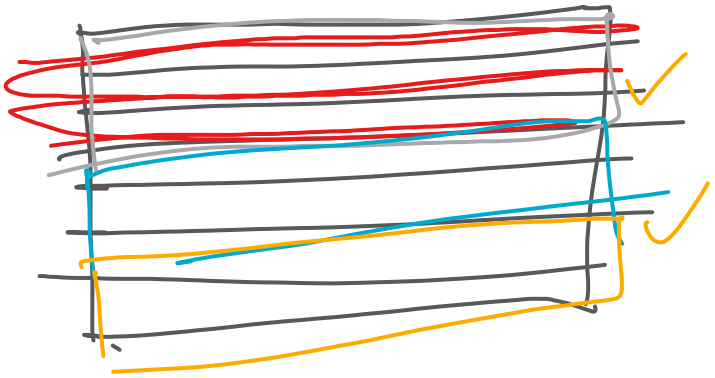
→ Vanishing Gradient problem

→ Exploding Gradient

Gradient Descent.



Why SGD: Ans It solves the problem of being stuck in the local minimas (refusing to reach global minima)



Training NN:

Interview

Time → minimum

Small (0.03) (0.04)

3 mins

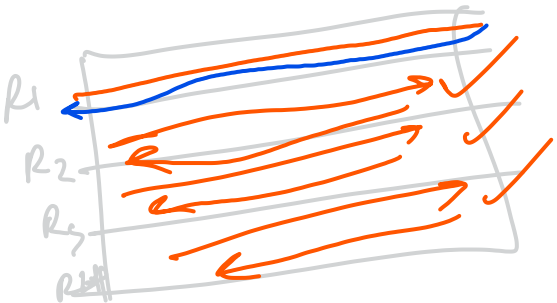
0.00 → 10^{-5} → 3 secs

Train 0.00001 minimal and negligible → 30 epochs

Vanishing gradient problem:

Exploding point

Small



weights ↓

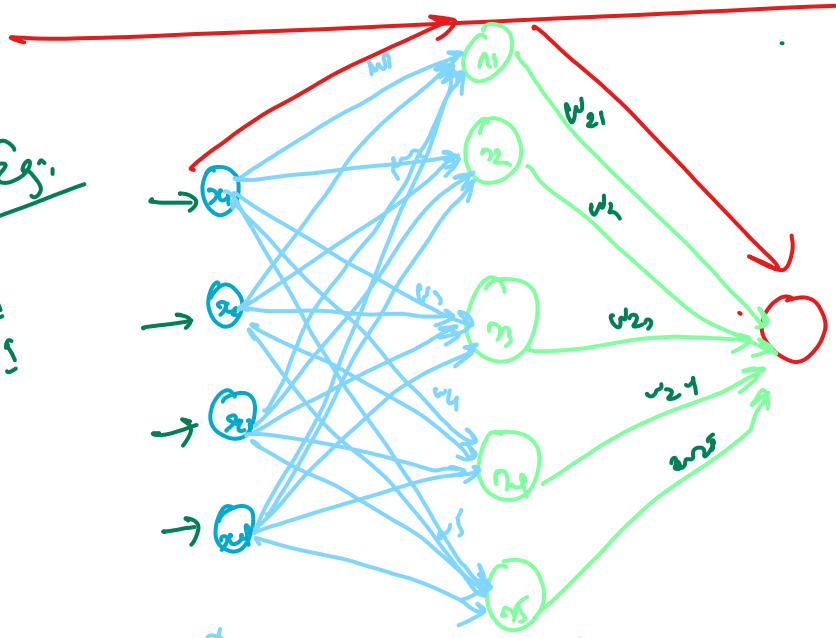
0 - 1

3, 4

b = 1

to

What happens when weights are big? → Curse of dimensionality



Eg:

$I = 1$
$w = 1$

$w_{21} \Rightarrow x_1 \times w_1 + x_2 \times w_2 + x_3 \times$

Input → hidden → output

20 synapses (weights) { = 25 weights.

Some maths:

How old is the universe?

- Around 5 billion years

almost 5b times the age of the universe.

Sunway TaihuLight → Chinese Supercomputer

↓

Guinness World Record (3rd fastest) ✓

93 PFlops

↓

Peta

1.0×10^{125} ⇒ sec ↓ minutes ↓ hours ↓ days

1.0×10^{125}

$\frac{1.0}{6} \times 10^{124}$

0.166×10^{124}

$1.66 \times 10^{-1+124}$

1.66×10^{123}

$\frac{1.66}{24} \times 10^{123}$

0.0691

$6.91 \times 10^{-2+123}$

6.91×10^{121} → yrs (365)

$\frac{6.91}{365} \times 10^{121}$

0.002

$2.0 \times 10^{-3+121}$

2.0×10^{118} yrs

$\frac{2.0 \times 10^{118-9}}{1.0 \times 10^{9}}$

2.0×10^{109} Billion years

$\frac{2.0 \times 10^{109}}{9.3 \times 10^{15}}$

$\Rightarrow \frac{2.0}{9.3} \times 10^{109-16}$

$\rightarrow \frac{2.0 \times 10^{93}}{9.3} \Rightarrow 0.02 \Rightarrow 2.0 \times 10^{-2} \times 10^{93}$

2.0×10^{91} yrs

5 Billion years

5.0×10^9 vs 2.0×10^{90} ✓

Backpropagation with an example !

Steps in Train an ANN using SGD algorithm:

Step 1: Randomly initialize weights (0–1) but not 0 & not 1 and input values

Step 2: Forward propagation ($w_i x_i$) until you get $\hat{y}$ (output)[predicted]

Step 3: Calculate the (Error) loss (by comparing Expected vs Actual).

Step 4: Backpropgate → update weights, if possible (learning rate.

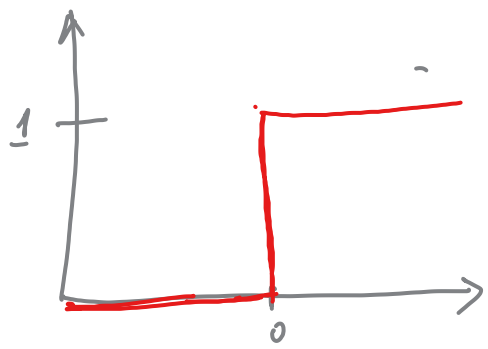
Step 5: Repeat Steps 1–4 until convergence. Then Stop.

Activation functions

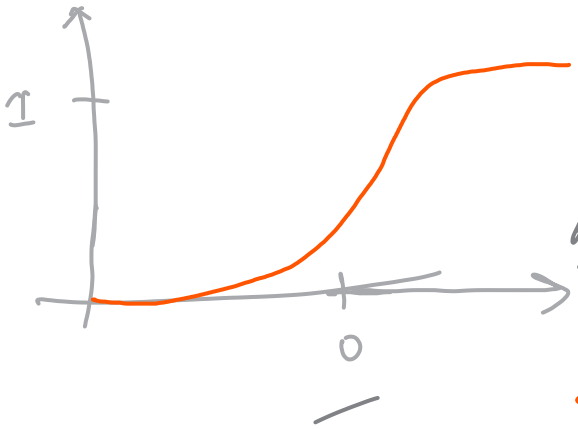
1) Threshold function
2) Sigmoid function
3) ReLU "
4) Hyperbolic Tangent (tanH)

write note on use case of all activation function

1) Threshold function



$$\phi(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0. \end{cases}$$

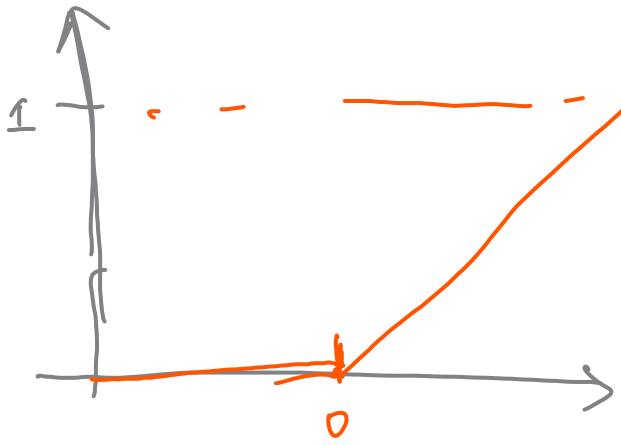


Probability prediction

$\phi = \frac{1}{1+e^{-x}} \rightarrow \frac{e^x}{1+e^x} \rightarrow$ log odds $\rightarrow \frac{p}{(1-p)}$

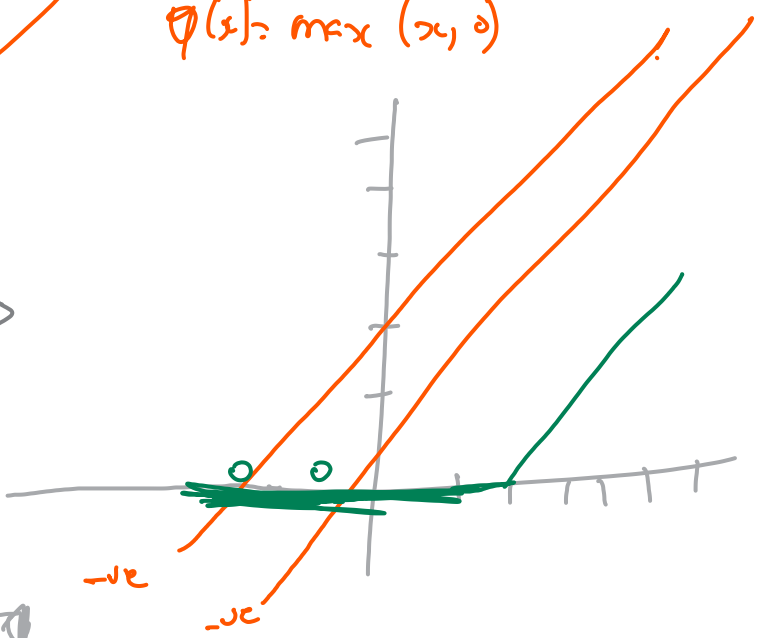
ANN

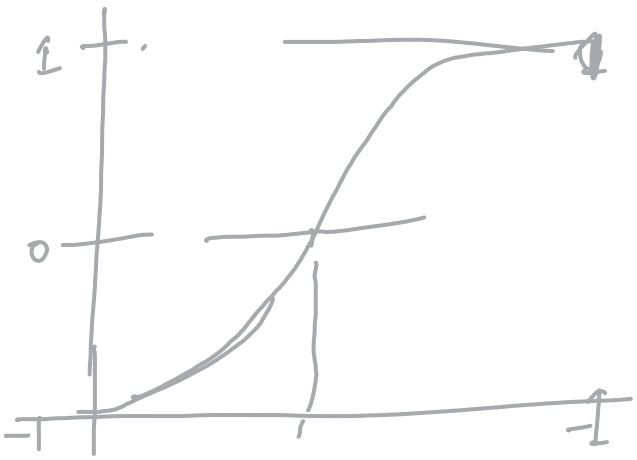
$\sum_{i=1}^{n} w_i x_i$



ReLU

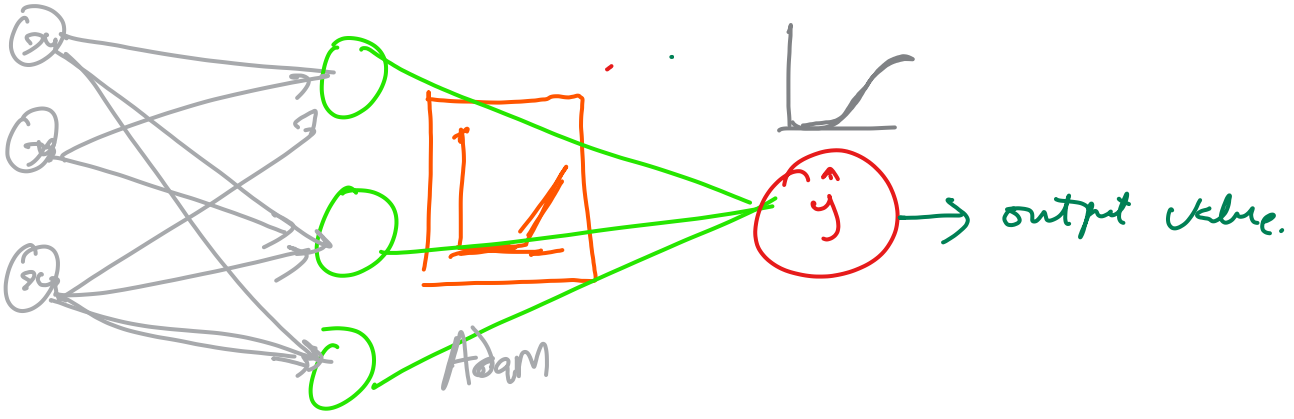
$\phi(x) = \max(x, 0)$





$$\phi = \frac{1 \pm e^{-2x}}{1 + e^{-2x}}$$

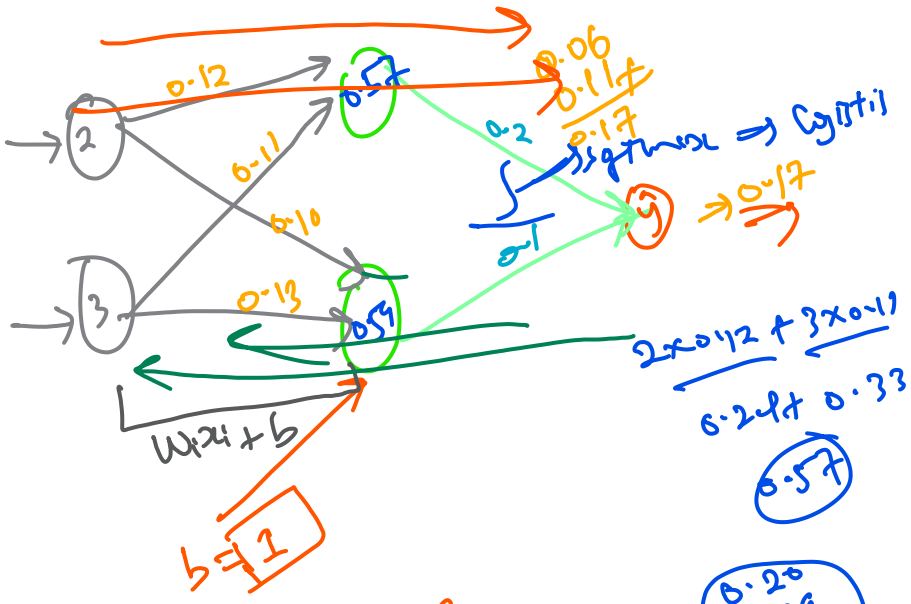
In Summary:

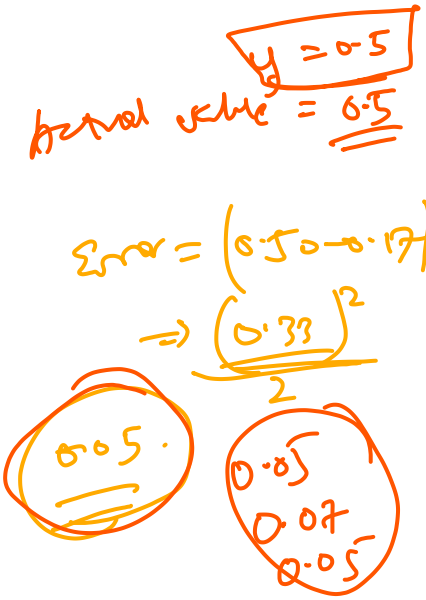


→ output value.

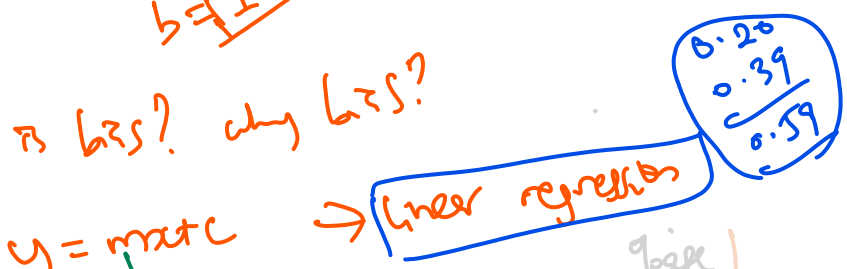
Deep Sparse Rectified (rectifier) neural networks

Author Xavier (2011)





What is LRS? why LRS?



y = mx+c

y = bx+b

y = wx+b