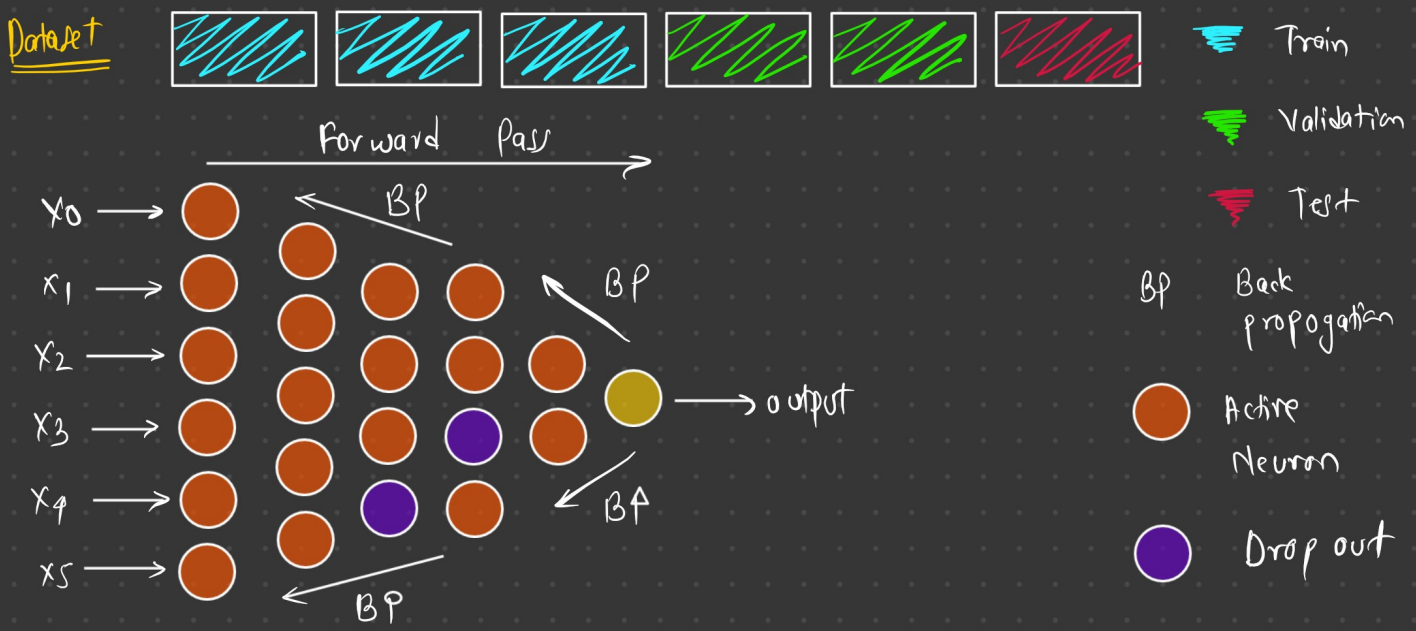
01 - 12 - 2024

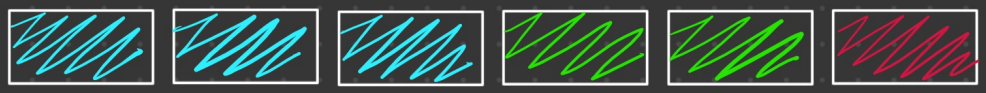
Agenda

- Flow of data in Neural Network
- Training intuition in Neural Network
- Activation function intuition
- Components of Neural Network

Components of Neural Network

- Input layer
- Hidden layer
- Output layer
- Back propogation
- Batch Normalization
- Forward Pass
- weights
- Neurons
- loss Function
- Learning rate
- Epoch & Iteration
- Hyper parameter
- Bias
- Activation Function
- Optimizer
- Dropout
- Batches

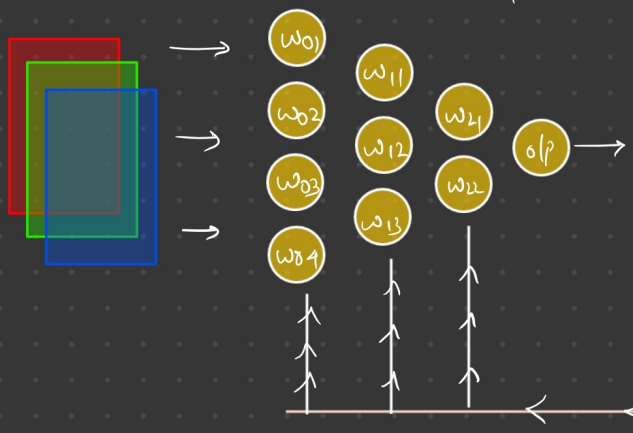




X train
(Image)

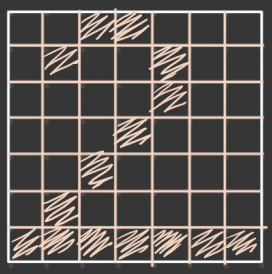
Y train
(Ground Truth)
(Actual value)

Train
Validation
Test



Forward pass → Prediction → Loss function
(Actual − Prediction)
↓
Optimize weights
↓
Optimizer [GD, ADAM]
↓
Back propogation

Decoding Neural Network



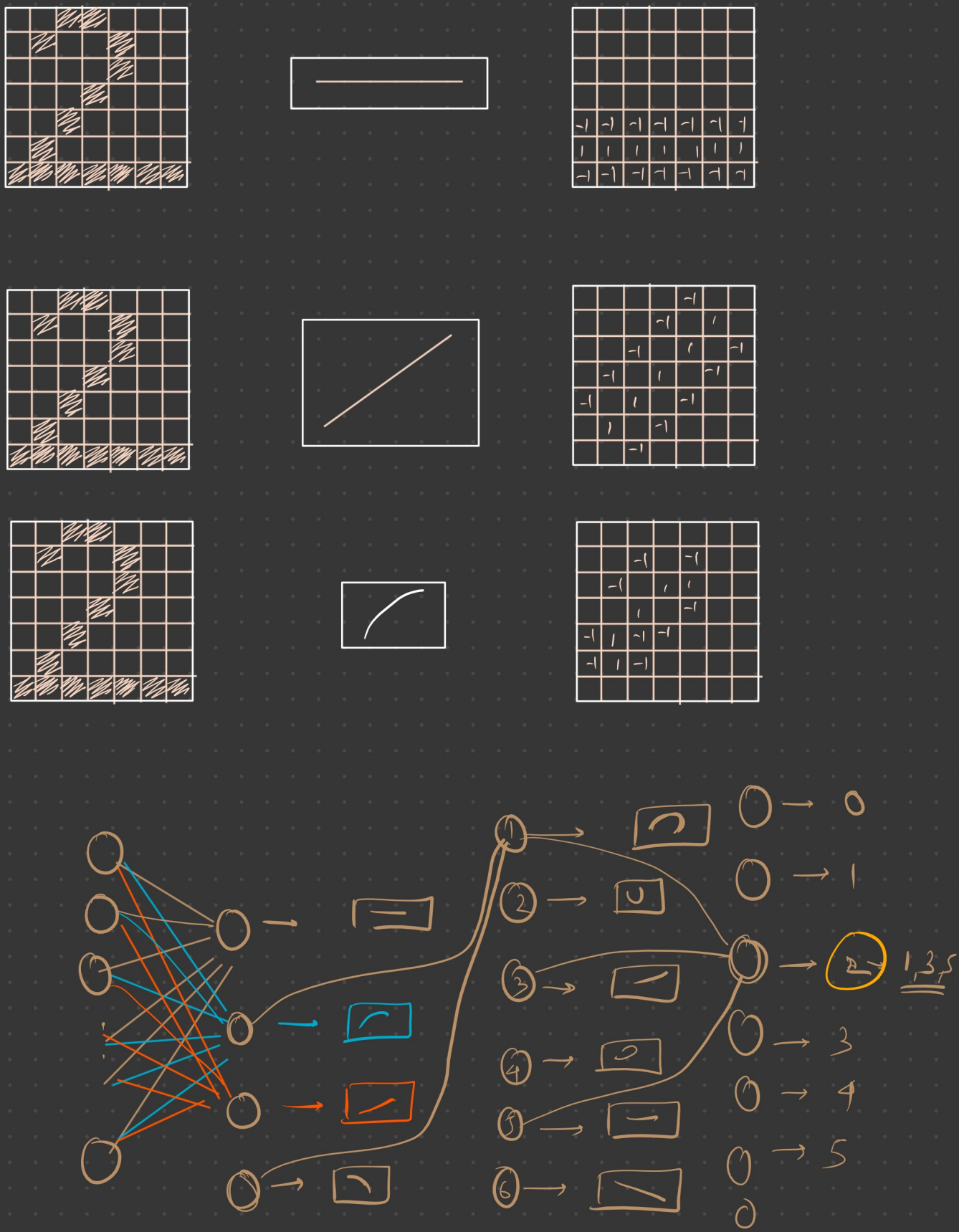
7 x 7

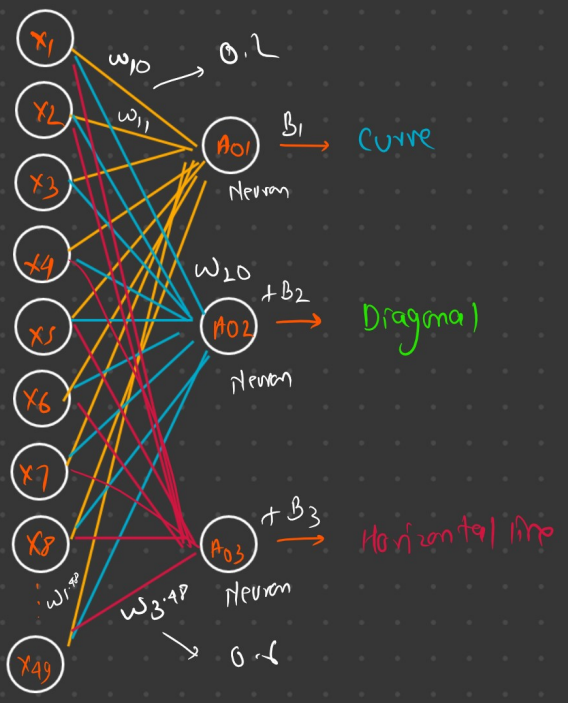
Pattern 1

Patter 2

Pattern 3

Pattern 4





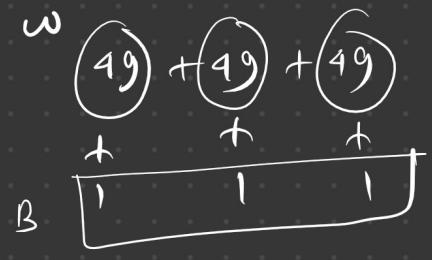
model → 0

→ 1

→ random → 0.2

→ 0.3

→ 0.6



150 total parameter

Curve = $X_1 \cdot w_{10} + X_2 \cdot w_{11} + X_3 \cdot w_{12} + X_4 \cdot w_{13} \cdots X_{49} \cdot w_{1\cdot 48} + B_1$ (0.3, −0.1)

= sum tve or -ve

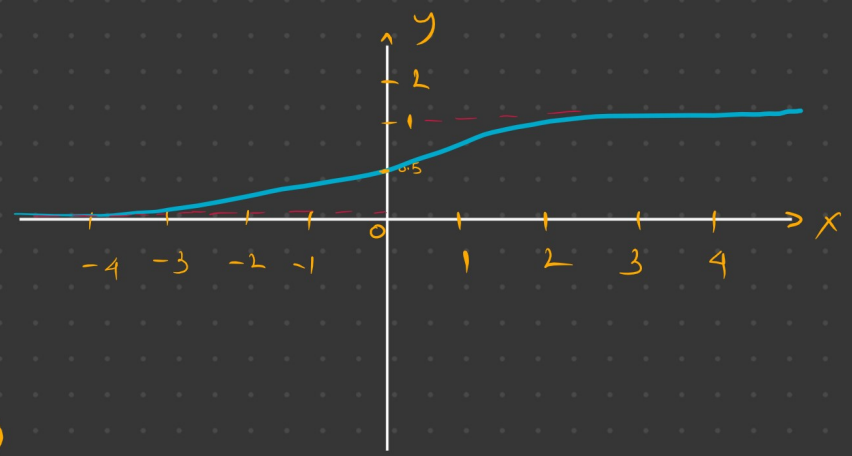
= 2

Diagonal = $x_1 \cdot w_{20} + x_2 \cdot w_{21} + x_3 \cdot w_{22} + \cdots x_{49} \cdot w_{2\cdot 48} + B_2$

Horizontal line = $x_1 \cdot w_{30} + x_2 \cdot w_{31} + x_3 \cdot w_{32} + \cdots x_{49} \cdot w_{3\cdot 48} + B_3$

$f(x)$ ← sum → Sigmoid.

$$\rightarrow \frac{1}{1+e^{-x}}$$



Range → 0 to 1

→ 0.1, 0.2, 0.9

→ 0.5 ≥ 1 → Active

0.5 < 0 → Inactive

So activation basically is how tve the weighted sum is.

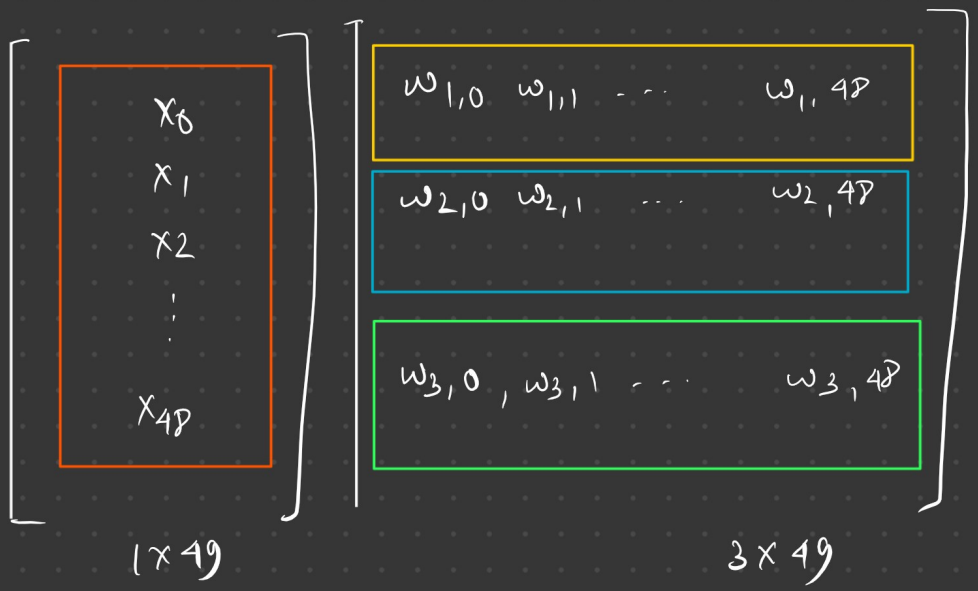
But its not always that want the neuron to be active when weighted sum $\geq 0 \rightarrow 0.5 \geq$ → Active

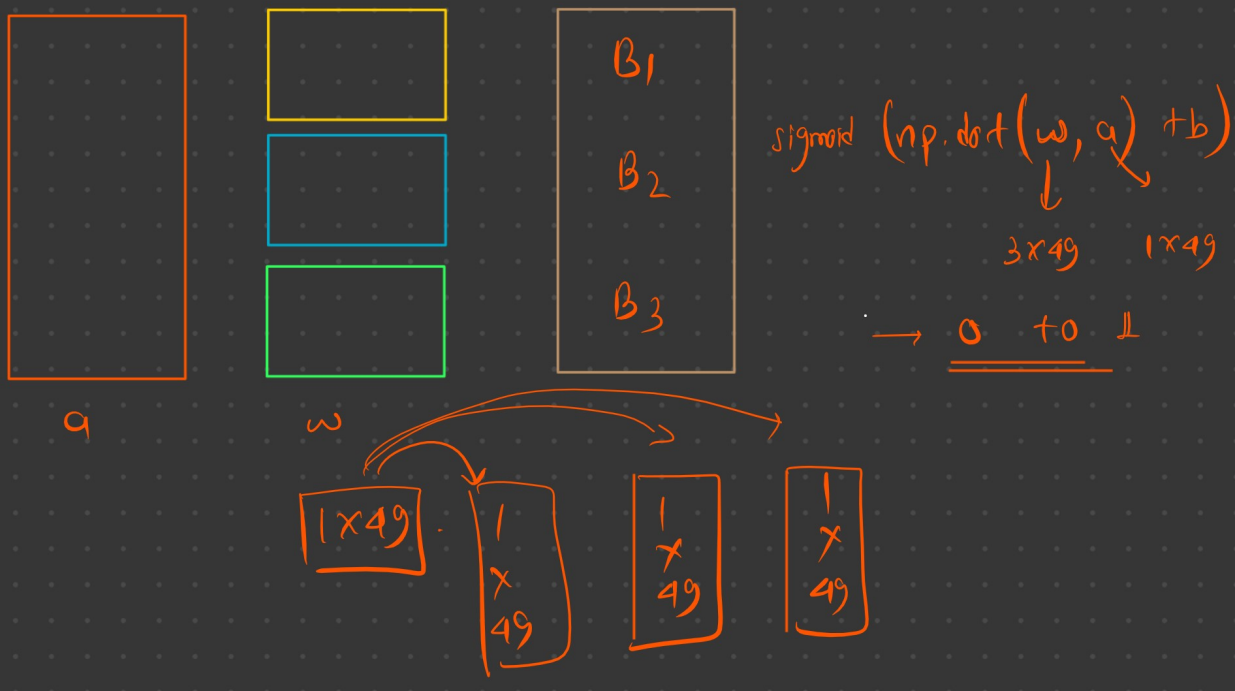
(sum → sigmoid)

Q. I want my neuron to be only active where sum > 15

Then we need to add bias : -15

So weight is responsible for pixel pattern a particular neuron is picking, & bias tell you the sum of weights need to be before the neuron start to getting meanfully active.





dataset → 0 0 0 - 9

1 \

Cost Function

	Actual
0 → 0.1	0
1 → 0.7	1
2 → 0.3	0
3 → 0.1	0
4 → 0.4	0
5 → 0.1	0
6 → 0.01	0
7 → 0.5	0
8 → 0.1	0
9 → 0.4	0

→ Input

1

max (all value)

↓

0.7

↓

index

↓

1 →

softmax → 0, 1, 3, 9

sigmoid → binary classification → 0
1

loss function — MSE

cost → Total loss across my data

cost → $(0.1-0)^2 +$
$(0.7-1)^2 +$
$(0.3-0)^2 +$
$(0.1-0)^2 +$
$(0.4-0)^2 +$
$(0.1-0)^2 +$
$(0.01-0)^2 +$
$(0.5-0)^2 +$
$(0.1-0)^2 +$
$(0.4-0)^2$

→ 1.6

↓

opt