LLM

ChatGPT → Generative Pre-Trained Transformer

Architecture

⇓

LSTM

⇓

RNN

⇓

ANN

⇓

Artificial Neural Networks

GPT → OpenAI

Claude → Anthropic

Gemini → Google

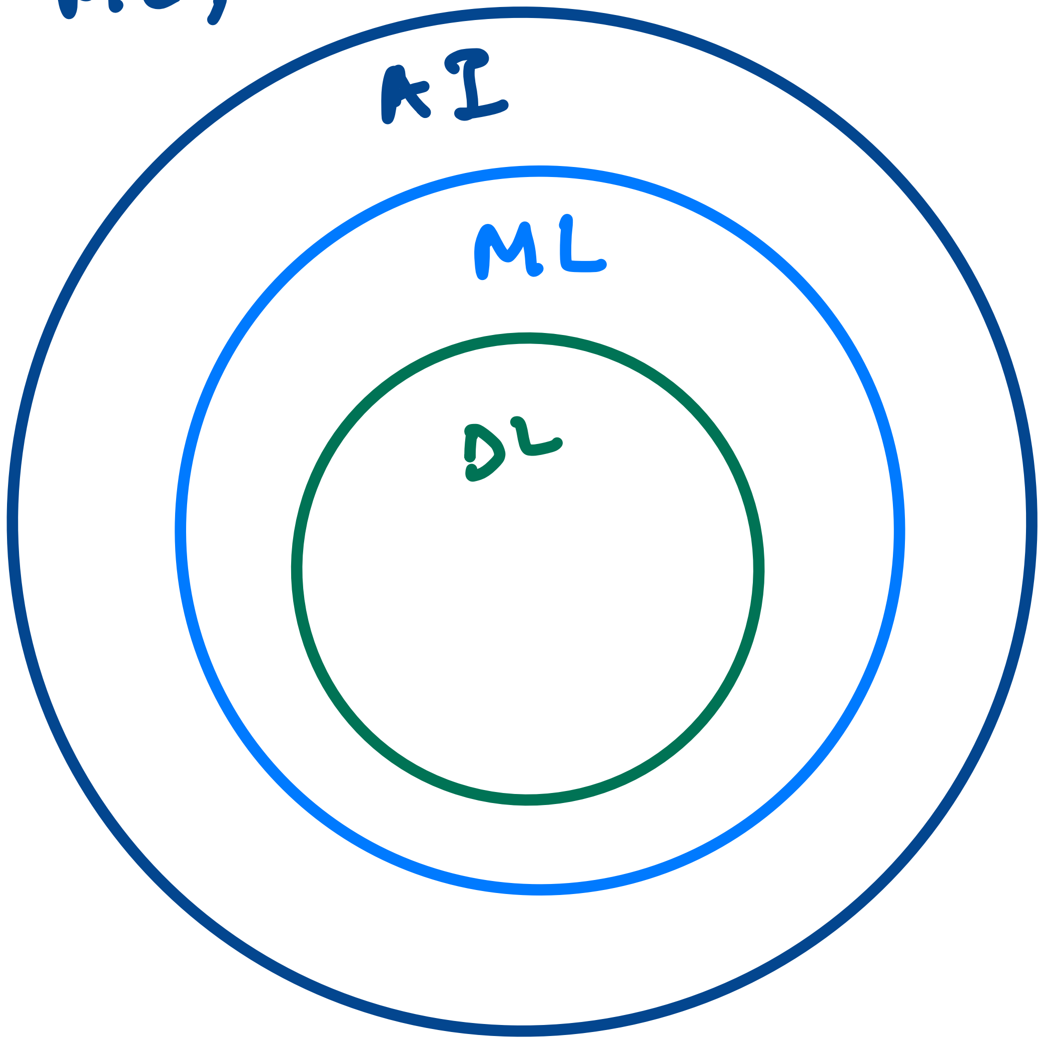
Deepseek → Deepseek AI

Llama → Meta

Mistral → Mistral AI

AI Agents, RAGs

ML, AI



Deep Learning

→ Large Data Sets
→ Complex Patterns
→ High Computational Power

Universal Approximation Theorem

→ Neural networks can approximate any function if it has enough hidden layers and neurons

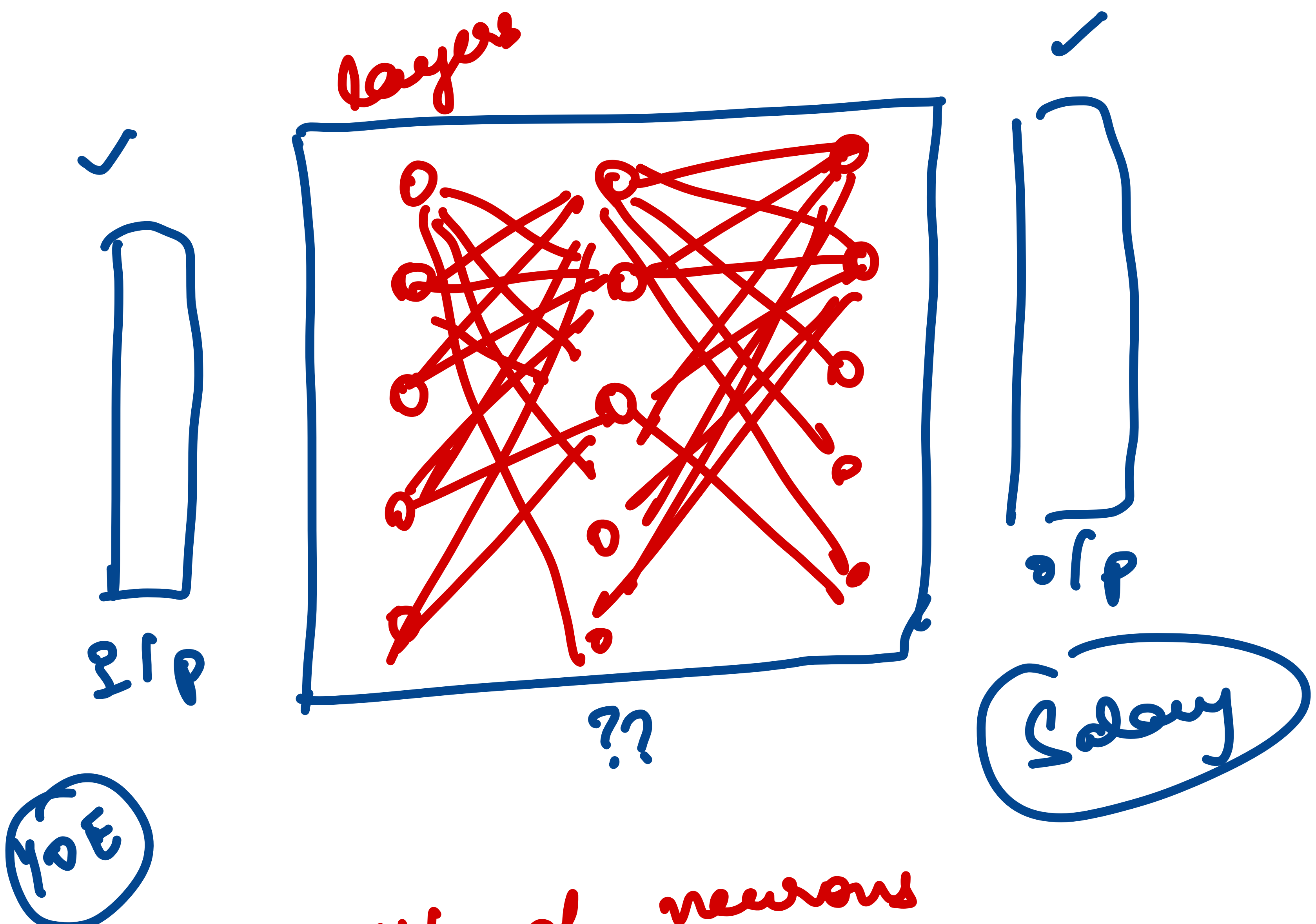
DL → Universal Function Approximator

→ Mimic Human Brain

→ Neurons

→ DL → Artificial Neurons

Salary
⇑
YOE of employee



N/w of neurons

Categories of Machine Learning

Supervised

data → label

YOE → Salary

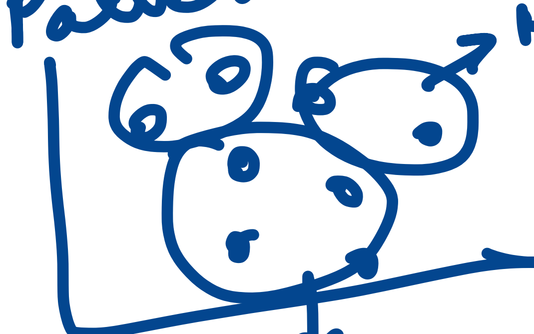
locat → price

→ employee

100K employees salary

Unsupervised

Patterns



→ Managers

Ser

customers

→ High paying customers

unlabeled

Reinforcement

↓

feedback

Deepseek

4 4
4 4

5 5

9 9 9

8
9

Image of Cat → Tail, Fur, Ears, Eyes

Speech → Audio → Sounds
⇓
Syllables
⇓
Word
⇓
Phrase

General Steps

① Define Problem Statement

② Collect and prepare data

③ Split Data

100K Data points

YOE → Salary

locatn, size → Price

→ Spam / Spam

— —

70%, 15%, 15%

⇓ 70K Train

⇓ 15K Validate

⇒ Test

10

5 → 20Lpa

2 → 10 Lpa

Train

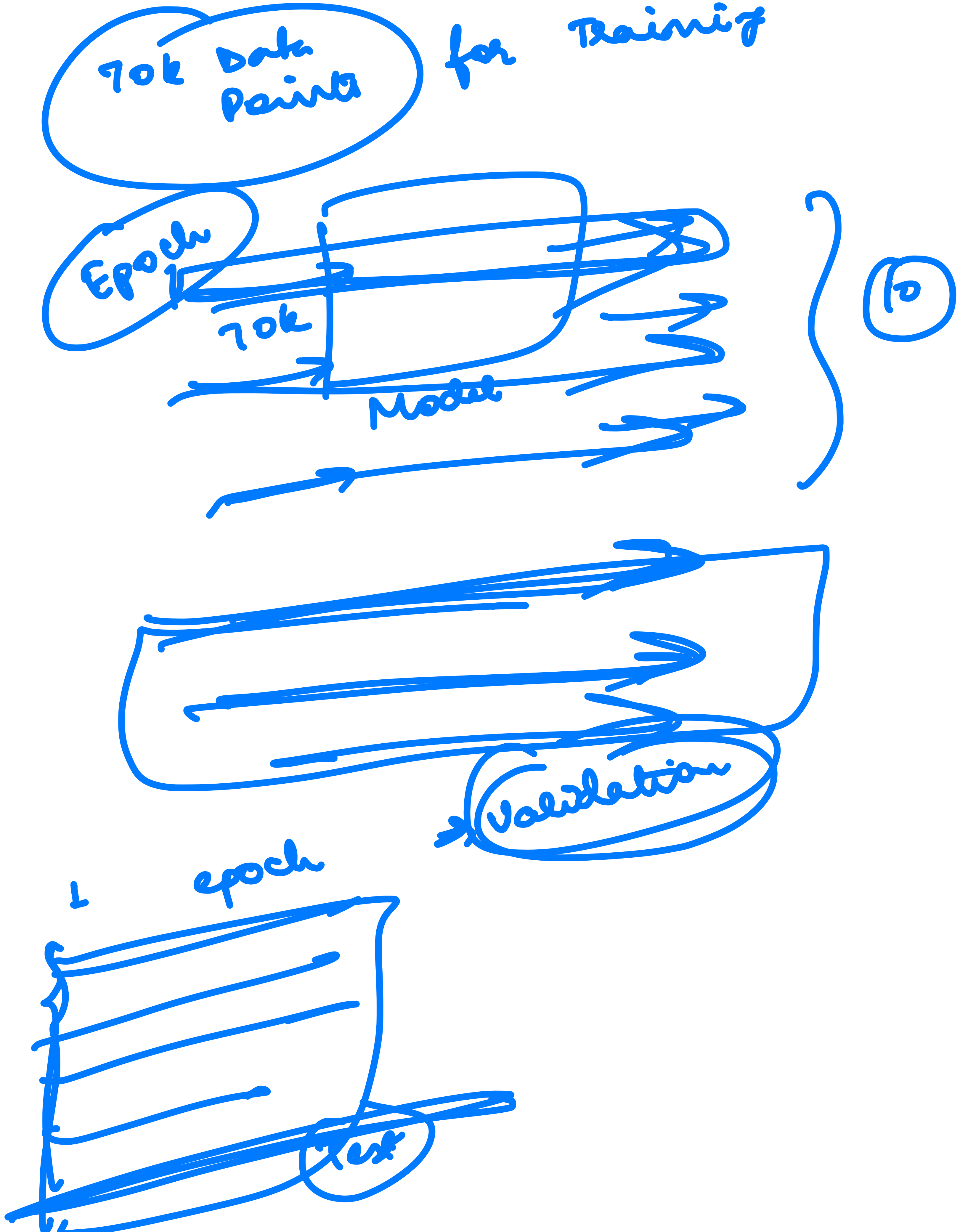
⇓

Smart

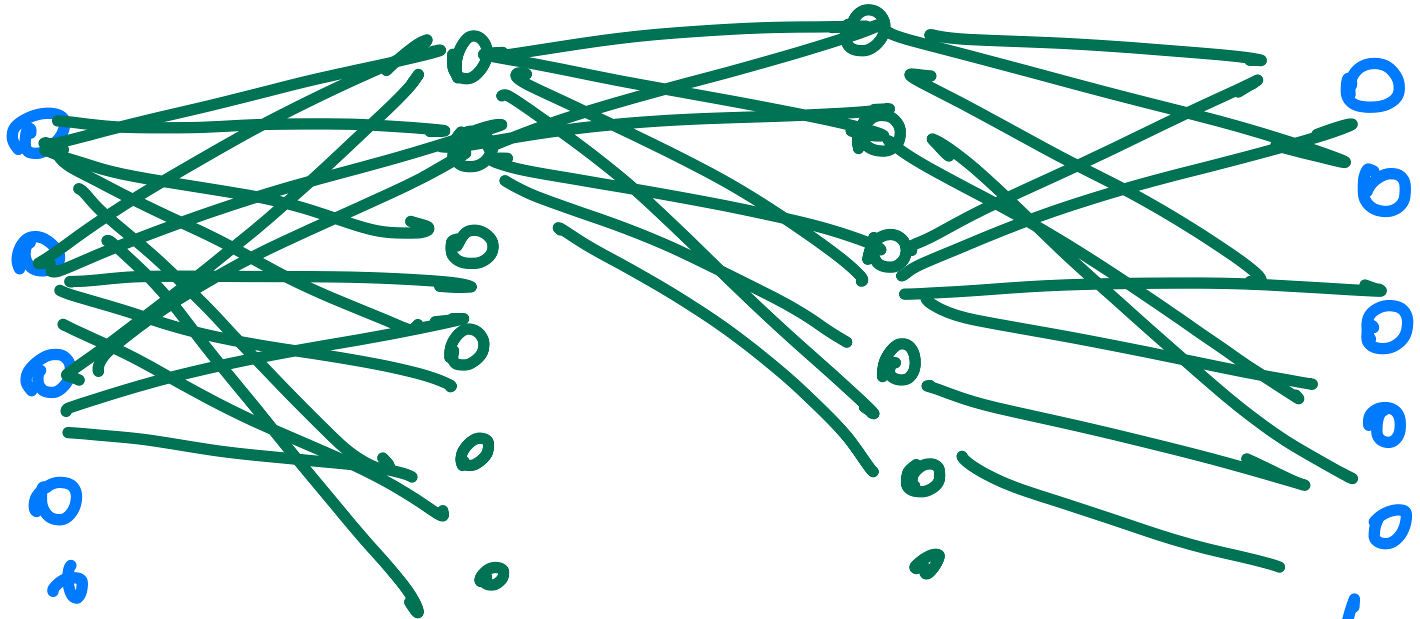
⇓

Test

% accuracy


70k Data Points
for Training
Epoch
70k
Model
10
Validation
1 epoch
Test

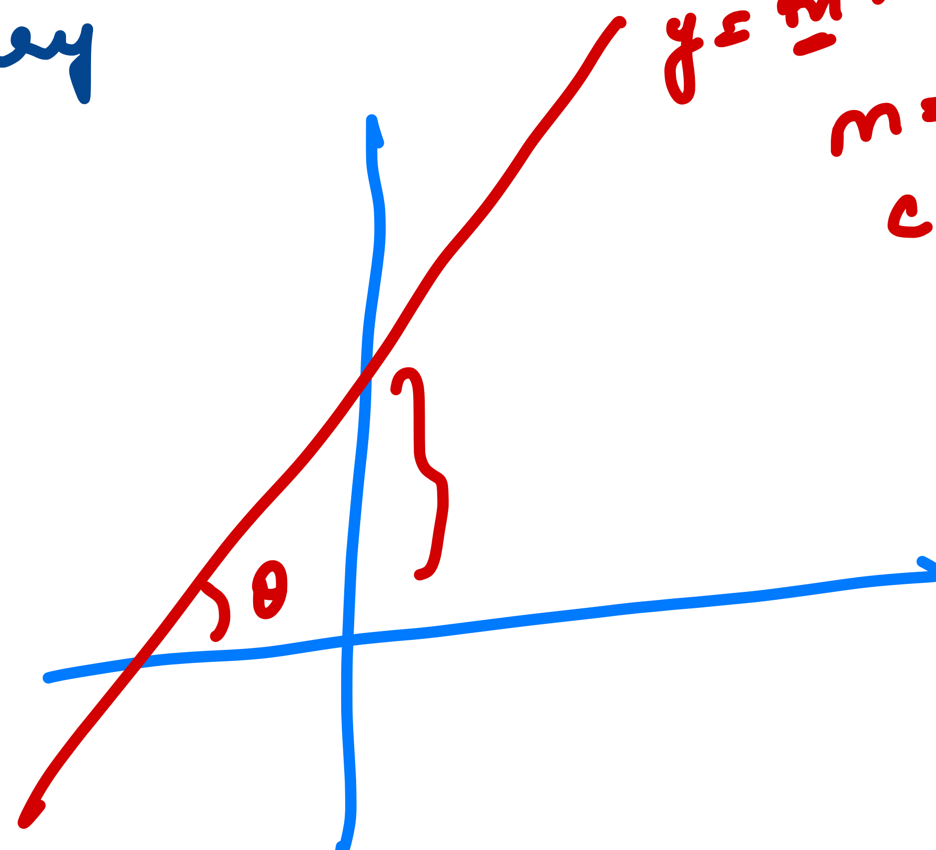
4. Define NN Architecture
   →
5. Train NN
6. Validate NN
7. Test NN
8. Deploy NN
9. Monitor



YOE vs Salary

Linear Regression

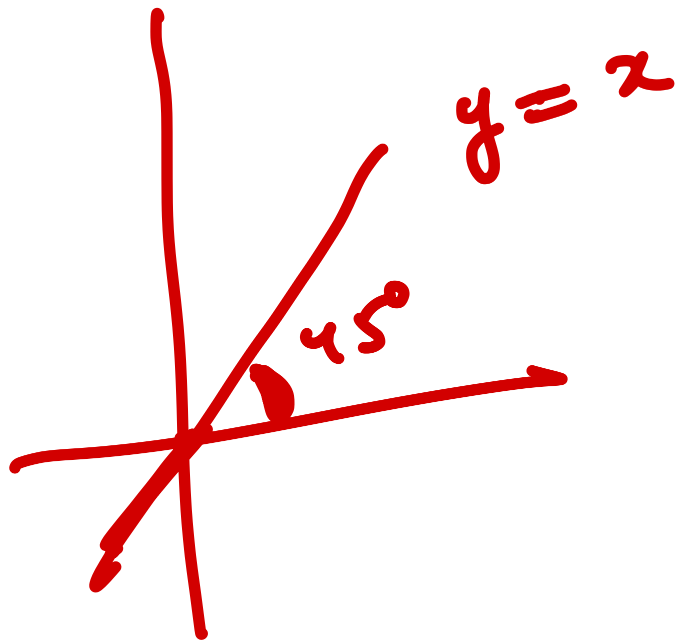
$y = mx + c$



$y = \underline{m}x + c$

m = slope

c = intercept



$m = 1$

$c = 0$

$y = 1.x + 0$

$\downarrow$

1

i/p $\longrightarrow$ o/p $\rightarrow$ Salary

$y = mx + c$

$y = wx + b$

x = YOE

$y_{true} \rightarrow$ Salary

$\hat{y} \rightarrow$ Predicting

$w\underline{x} + b$

$\downarrow$

yoe

Train

YOE = 5

Salary = 20 Lpa

y_{true}

$\hat{y} = wx + b$

$= w\,5 + b$

$w = 2$, $b = 5$

random w and b

$= 10 + 5$

$= 15$

$y_{true} = 20$ $\hat{y} = 15$

$(y_{true} - \hat{y}) = 5$

minimum

Loss | Cost $= (y - \hat{y})^2$

$y \rightarrow$ true

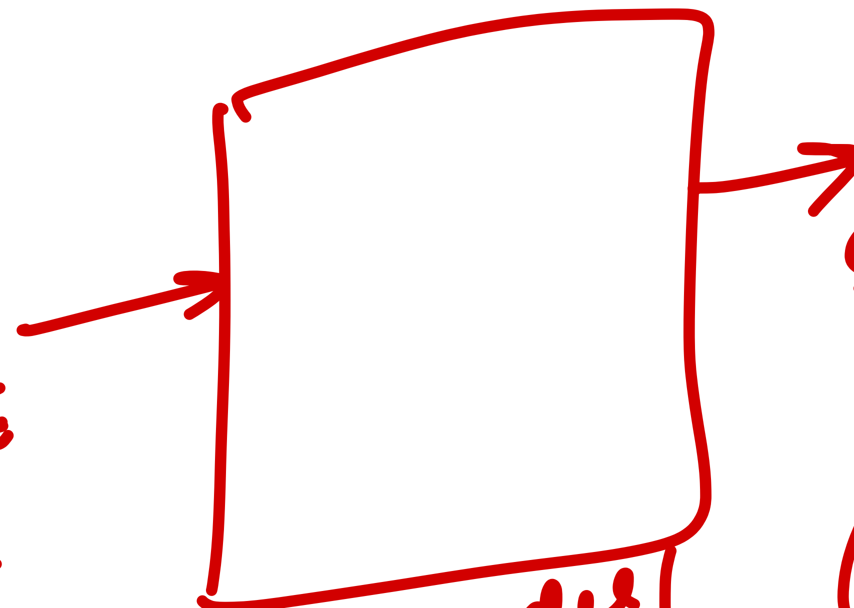
from training data set

$\hat{y} \rightarrow$ Predicted

i/P

YOE

x



o/P

Salary

$y \rightarrow$ true

$\hat{y}$

Model

NN

assume $\Rightarrow$ linear relationship

$y = mx + c$

$\quad wx + b$

Loss | Cost $\quad y - \hat{y}$

Goal ⇒ Minimise loss / cost

$y - \hat{y}$

Making true and the predicted value as close as possible

$\frac{\partial c}{\partial w}$

$\frac{\partial c}{\partial b}$

$b = b - \alpha \frac{\partial c}{\partial b}$

$w = w - \alpha \frac{\partial c}{\partial w}$

learning rate

$$C = (y - \hat{y})^2 \qquad \hat{y} = wx + b$$

$$\frac{\partial C}{\partial w} = \frac{\partial C}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w} \qquad \frac{\partial C}{\partial b} = \frac{\partial C}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial b}$$

$$= -2(y - \hat{y}) \cdot 1$$

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

$$= -2(y - \hat{y})$$

$$\frac{\partial \hat{y}}{\partial b} = \frac{\partial}{\partial b}(wx + b) \quad = 0 + 1$$

$$\frac{\partial \hat{y}}{\partial w} = \frac{\partial}{\partial w}(wx + b) \quad = x$$

$$b = b - \alpha\left(-2(y - \hat{y})\right)$$

$$\frac{\partial C}{\partial w} = -2(y - \hat{y})\, x$$

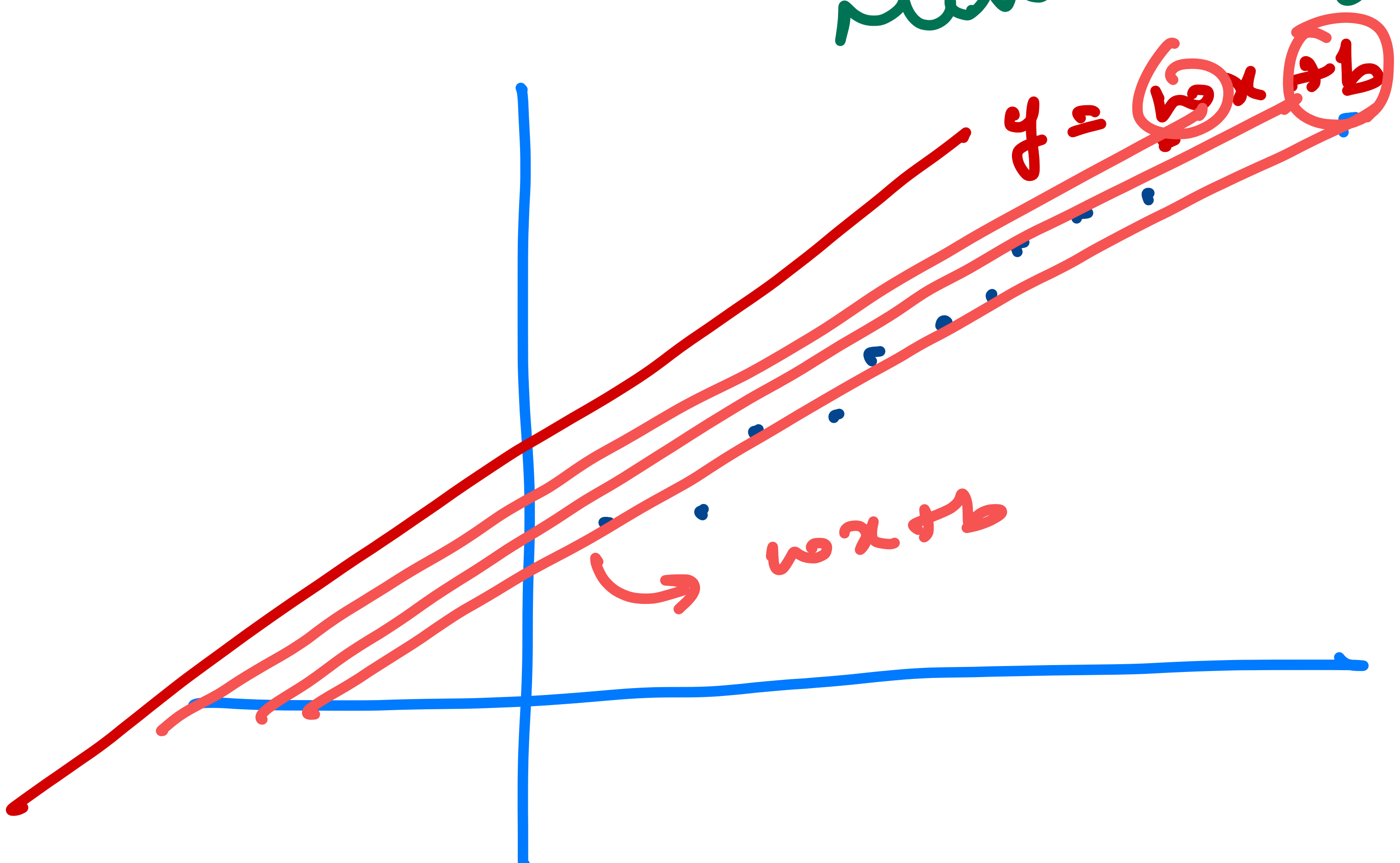
$$= -2x(y - \hat{y})$$

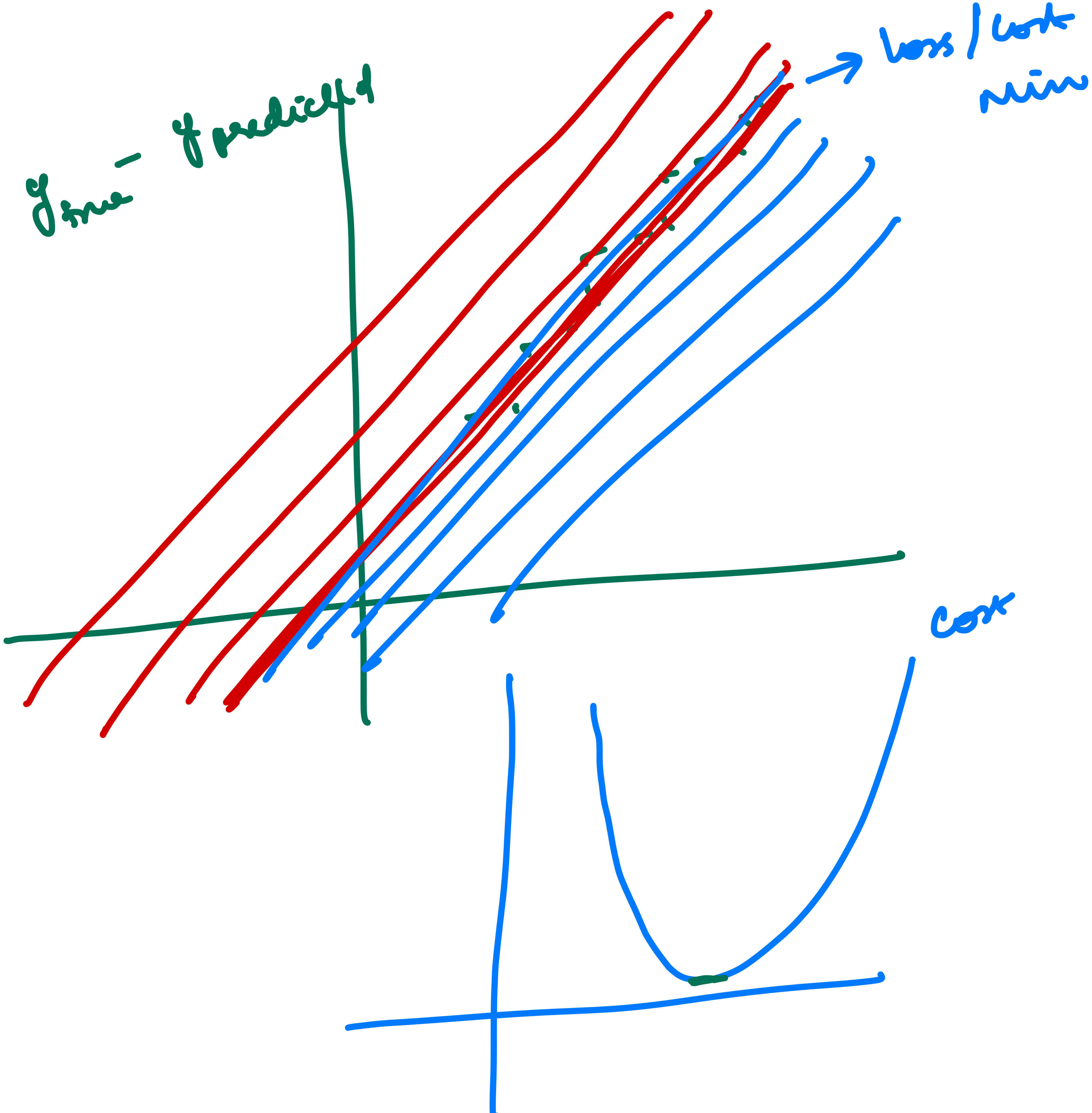
$$w = w - \alpha\left(-2x(y - \hat{y})\right)$$

$$y = wx + b$$

predict $\swarrow$ $\hat{y}$ = (random w) x + (random b)

$$\underbrace{y_{true/actual} - \hat{y}_{predicted}}_{\text{Minimise cost/loss}}$$




y true - y predicted
loss / cost
min
cost

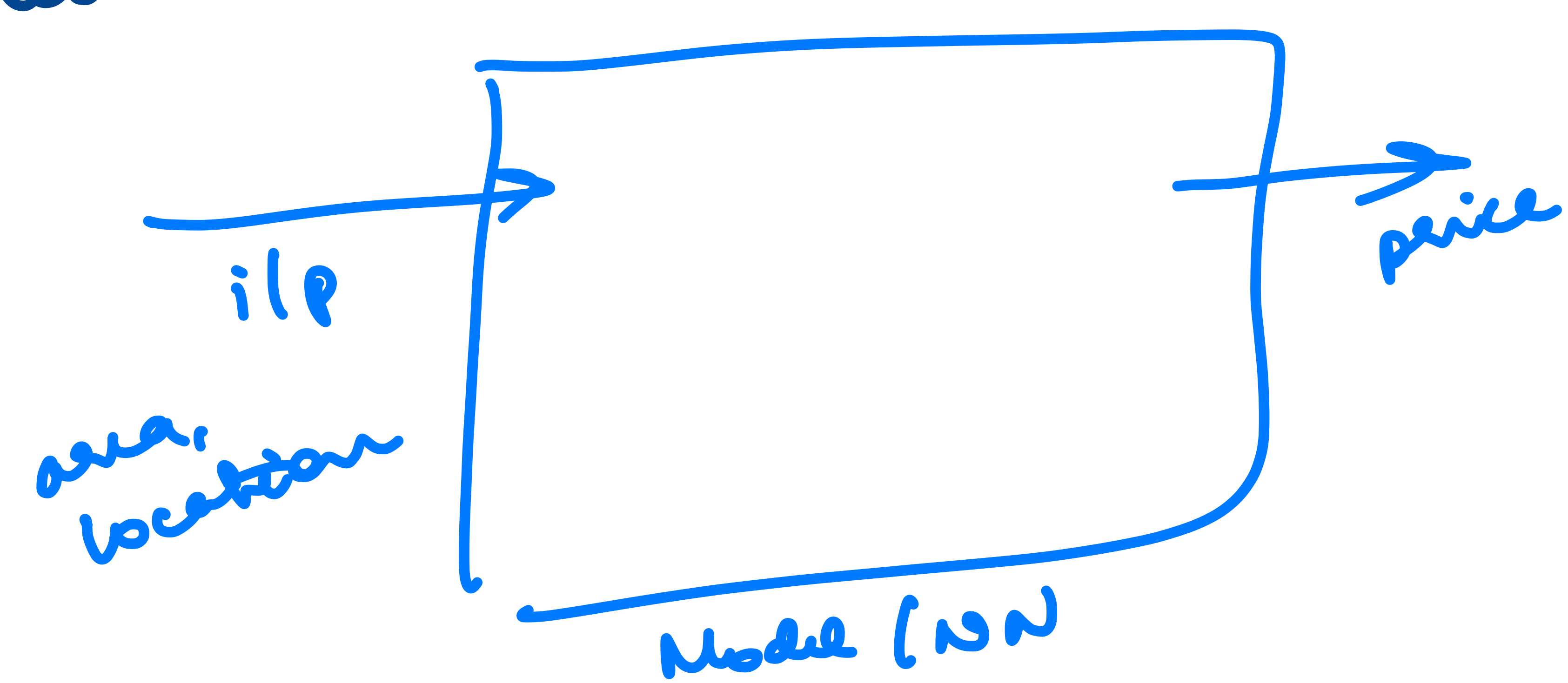
Data Points

location, area of house → House Price

100k Data →
— — → — } f_{true}
— —

⇓

new locatn, area → price ??



$\hat{y}$ predicted price $= w_1 x_1 + w_2 x_2 + b$

$w_1, w_2, b \rightarrow$ random values

$x_1 \rightarrow$ area

$x_2 \rightarrow$ location

$y_{true} - \hat{y}_{predicted}$

$\frac{\partial c}{\partial w}$

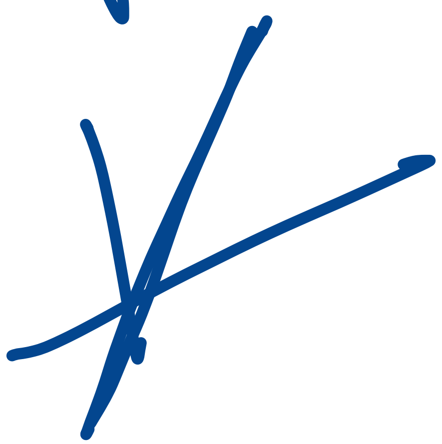
$\frac{\partial c}{\partial b}$

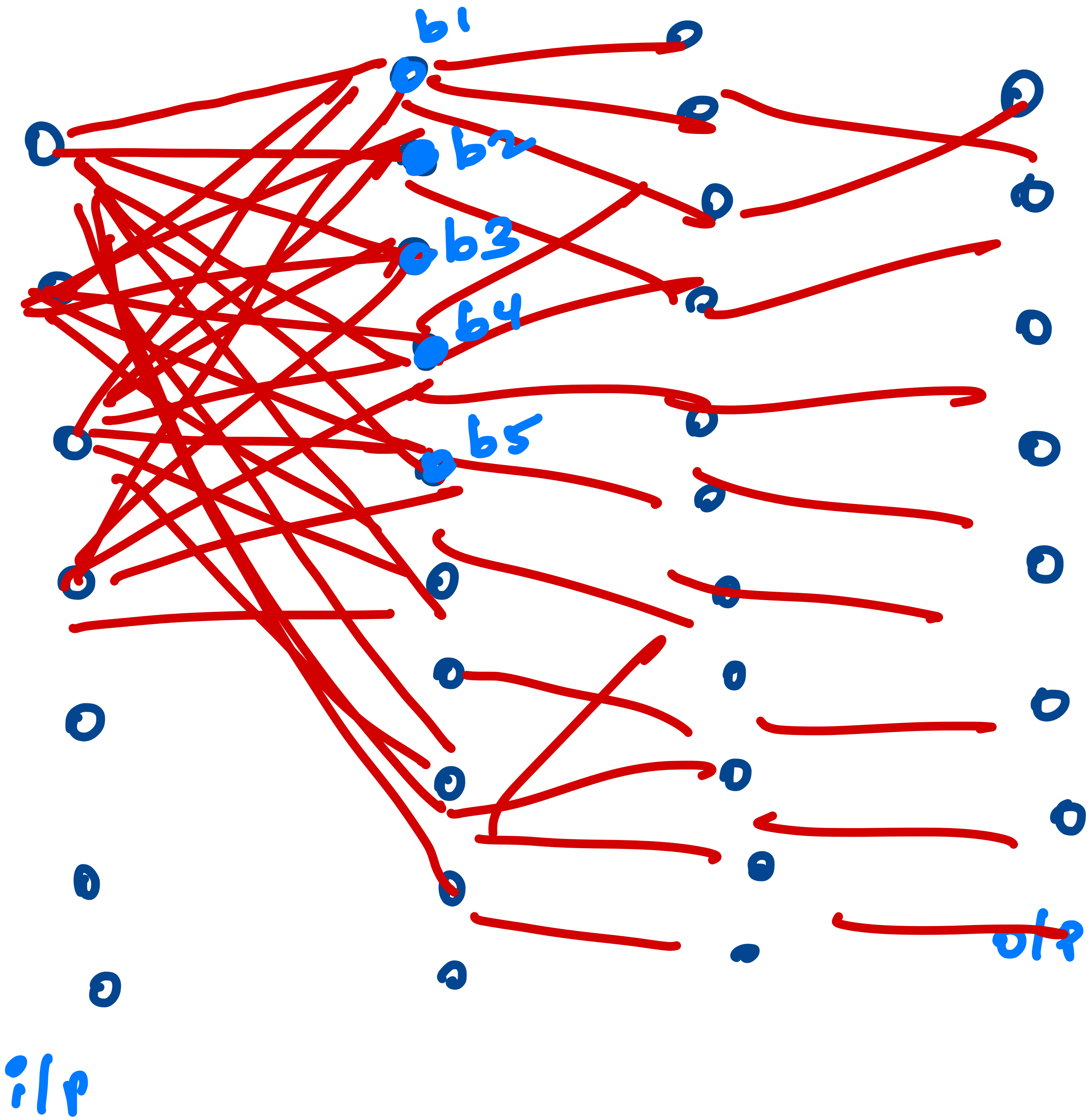
minimize cost / loss $= (y - \hat{y})^2$

weight = w

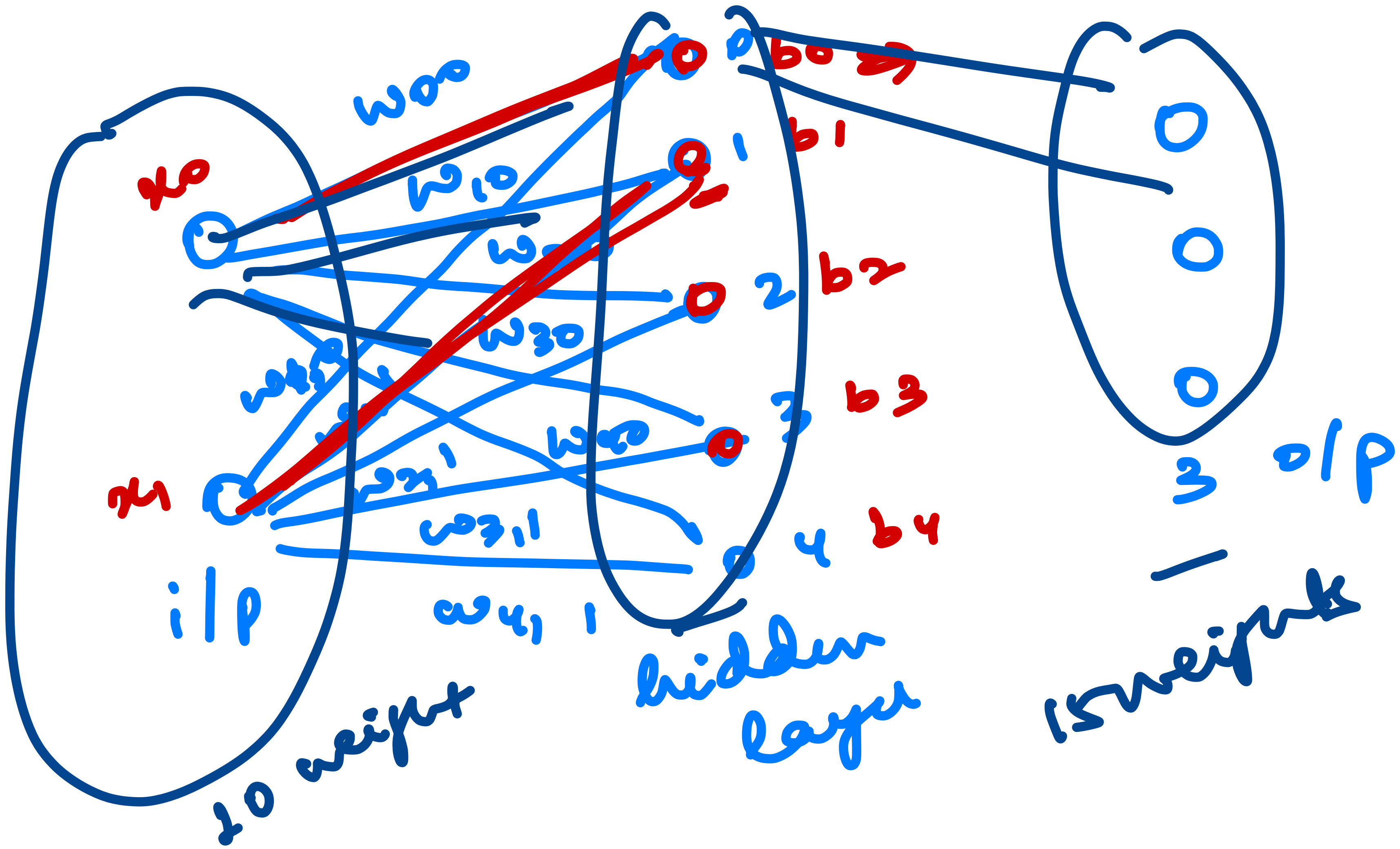
$y = wx + b$

bias



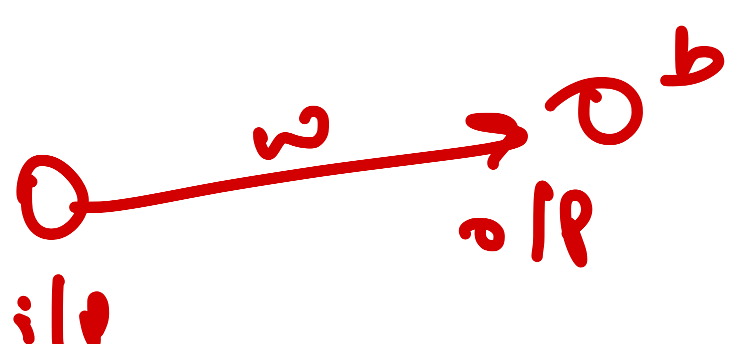


i/p $\xrightarrow{w}$ o/p



$w_{00} x_0 + b_0$

$w_{10} x_0 + b_1 + w_{11} x_0$



10 + 15

= 25 weights

& biases

Training Process

① Initialise weights & bias → random values

② I/p → o/p → Calculate predicted o/p $\hat{y}$ → fwd pass

③ Calculate loss/cost → Algos | loss function

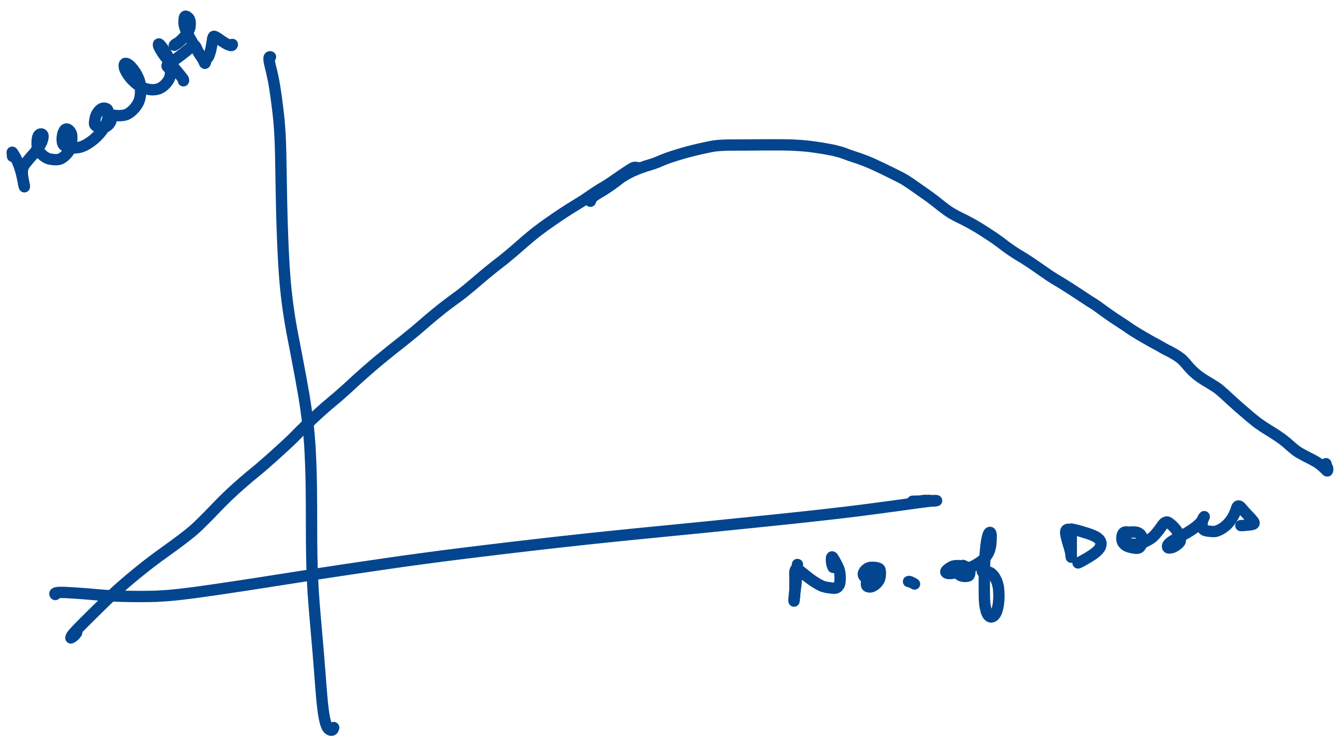
④ Compute Gradient $\frac{\partial C}{\partial w}$, $\frac{\partial C}{\partial b}$ using chain rule

→ Backward Propagation

⑤ Update weights, biases using gradient

⑥ Repeat until loss/cost is minimised

No. of doses vs Health



$$y = wx + b$$

$$y = \underbrace{\sigma(x)}_{\text{Activation Function}}(wx + b)$$

Sigmoid function

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

$$z = (wx+b)$$

$$\hat{y} = \sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(wx+b)}}$$

$$y = \underline{wx+b}$$

linear func.

$$\underline{\sigma(wx+b)}$$

non-linearity

tanh, ReLU, leaky ReLU,
Softmax

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

$$\frac{\partial}{\partial x}\left(\sigma(x)\right) = \frac{d}{dx}\left(\frac{1}{1+e^{-x}}\right)$$

$$\frac{\partial}{\partial z} z^{-1} = -z^{-2}$$

$$x^{n} \Rightarrow n x^{n-1}$$

$$z^{-1} \Rightarrow -1 z^{-2}$$

$$\frac{1}{(1+e^{-x})^{2}} \; - \frac{d}{dx}\left(1 + e^{-x}\right)$$

$$1 \Downarrow 0 \qquad e^{-x} \Downarrow$$

$$\frac{-e^{-x}}{(1+e^{-x})^{2}}$$

$$e^{x} \Rightarrow e^{x}$$

$$e^{-x} \Rightarrow e^{-x} \frac{d}{dx}(-x)$$

$$= -1(e^{-x})$$

$$\frac{-e^{-x}}{(1+e^{-x})^2}$$

$$\frac{1}{1+e^{-x}}\left(\frac{1-(e^{-x}+1)}{1+e^{-x}}\right)$$

$\sigma(x)$

$\sigma(x)(1-\sigma(x))$

$$\frac{d}{dx}(\sigma(x)) = \sigma(x)(1-\sigma(x))$$

$$\boxed{\frac{\partial C}{\partial w} \qquad \frac{\partial C}{\partial b} \qquad \hat{y} = \sigma(wx+b)}$$

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$z = wx+b$$

$$\sigma(wx+b)$$

$$\sigma(z) = \frac{1}{1+e^{-(wx+b)}}$$

Image

28 × 28 pixels

70k Data points

of handwritten digits

9 1 2 3 4 5

⇓

what digit

Read Image ⟹

$\underbrace{0, 1, 2, 3 \ldots 9}_{10}$

70k Data Points

60k

⇓

Training

10k

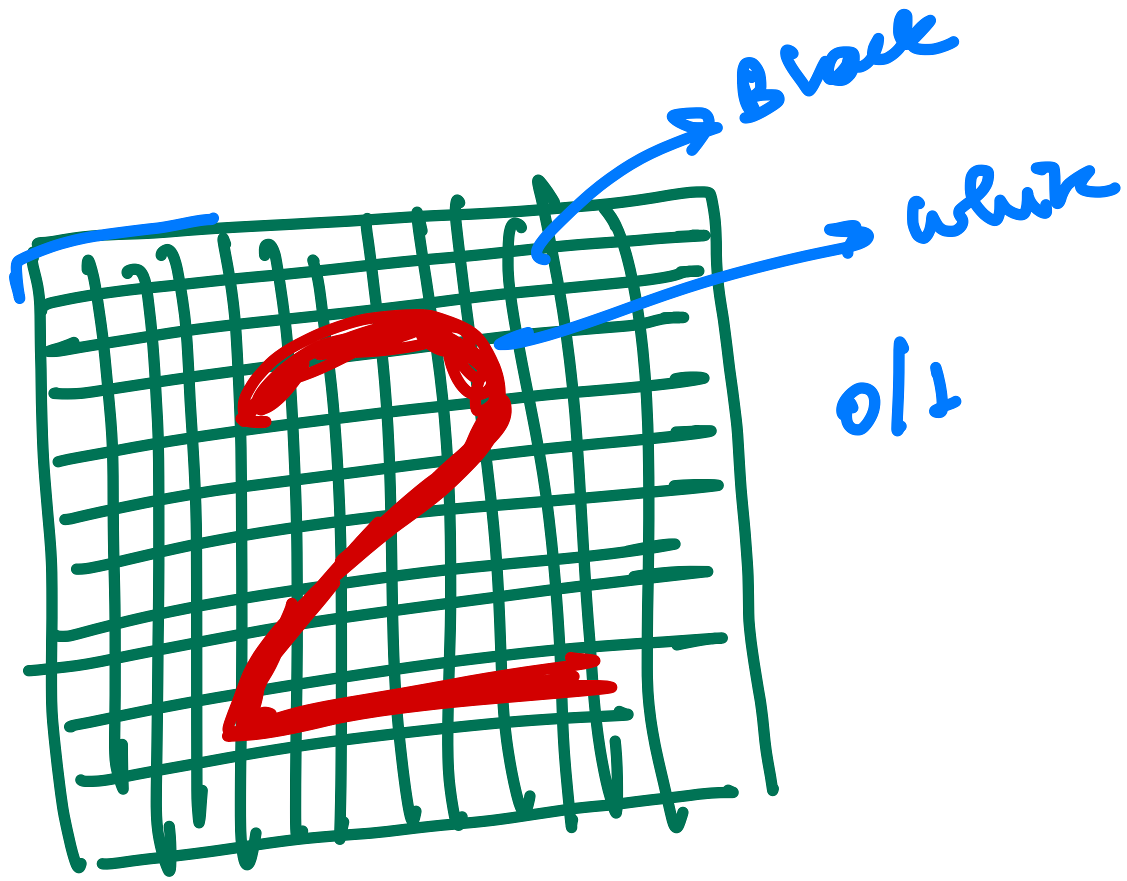
⇓

Testing

Join @hmaracollege for more free courses🙃

0 0
1 0
2 0
3 0
⋮
781 0
782 0
783 0
784

0 0
0 1
0 2
0 3
⋮
0 9
o/p layer


Black
white
0/1