9 Introduction to Matrices and Application

Matrix

↳ Rectangular array of no's / symbols / expressions → which are arranged in rows and columns

Eg

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

→ here a_{ij} where

$i \rightarrow$ denotes row

$j \rightarrow$ denotes the column

↳ we generalize this to have $m \rightarrow$ rows $n \rightarrow$ columns

Eg (matrices in Data Science)

① Data Representation (how do we use matrices in Data Science for Data Representation)

Say I have the scores of the student

↓

Math Score	Physics Score	Biology Score
55	65	75
65	60	55
70	45	80

Suppose say I want to train a model (you don't worry about the features) → can be anything

Only thing which we worry is about the numerical values and everything inside is considered as a vector

Representing the data as a matrix

$$\begin{bmatrix} 55 & 65 & 75 \\ 65 & 60 & 55 \\ 70 & 45 & 80 \end{bmatrix}$$

3x3

Each row represents the marks obtained ① by each student

Col → represents the mark obtained ② on a specific subject

basically the feature

This is what the model understand

① It doesn't have any idea on what is a feature and

② Images in Computer Vision

0 (black) ⟷ 255 (white)

color becomes lighter and lighter when you get near 255

Say I have a (3x3 Grayscale image)

Image =

0	128	255
255	128	0
128	255	128

→ This image can be represented in the form of matrix

$$\begin{bmatrix} 0 & 128 & 255 \\ 255 & 128 & 0 \\ 128 & 255 & 128 \end{bmatrix}_{(3\times 3)}$$

③ **Confusion matrix** → Tells about the accuracy of the model

Say I have a

$f_1 \rightarrow$, $f_2 \rightarrow$ $\boxed{\text{model}} \longrightarrow$ o/p ($\hat{y}$) → This is the predicted o/p

→ y → This is the actual o/p

find out the diff

and then create a confusion matrix (which talks about the accuracy of the model) (by a 4×4 matrix)

Confusion matrix

True Positive's

false -ves

$$\begin{bmatrix} 50 & 10 \\ 5 & 35 \end{bmatrix}$$

false +ves

True -ves

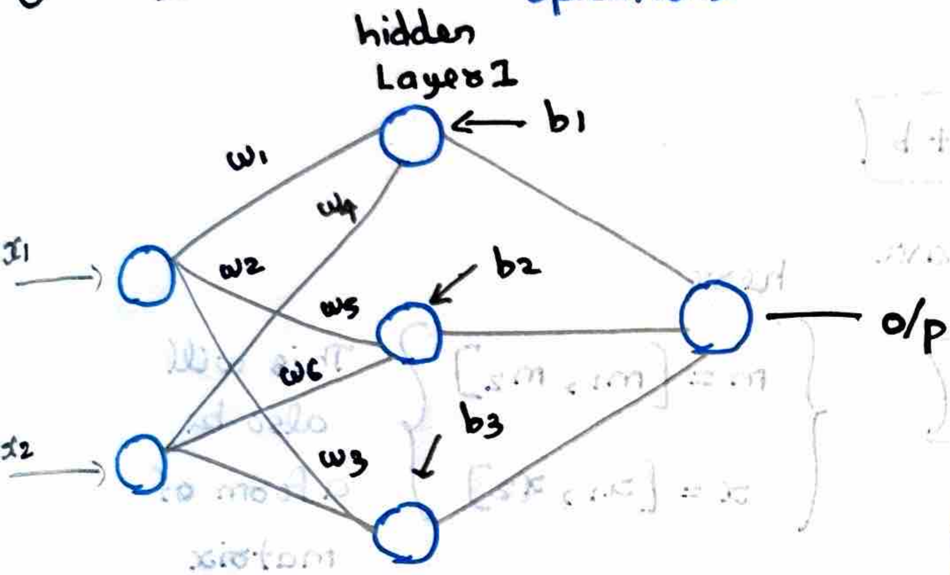
if we really want to understand how many our model correctly predicted →

$$\frac{TP + TN}{TP + FN + FP + TN}$$

we will get the accuracy

9^{th} cont

④ Neural N/w → [Matrix operation] → specifically in a concept called Linear Regression



matrix 2×3 → i/p's

↳ for each i/p we have 3 weights

During forward propagation

$$\boxed{\underset{(o/p)}{z} = w^T x + b}$$

here the i/p's

at the first stage this will get passed →

x_1	x_1
–	–
–	–
–	–
–	–

$$w = \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \end{bmatrix}$$

$$x = [x_1 \quad x_2]$$

$+$

$b_1 = [b_1 \; b_2 \; b_3]$

(This will get added at each layer)

for Linear Regression

Say you have 3 features

(x_1) No. of study hours	(x_2) IQ	Score (o/p feature) (dependent)
4	100	90
5	90	85
–	–	–

for this Regression

$y = mx + c$

$y = \underbrace{m_1 x_1 + m_2}_{\text{Slope / Coeff}} x_2 + c$

$y = \cancel{m}^1 \cancel{x}^1 + \cancel{c}$

$y = m_1 x_1 + m_2 x_2 + c$

Similarly like $\boxed{z = w^T x + b}$

we have

$\boxed{y = m^T x + c}$

here

$$\left\{ \begin{array}{l} m = [m_1, m_2] \\ x = [x_1, x_2] \end{array} \right\}$$

This will also be a form of matrix multiplication

(once we calculate this we actually get our best line based on no. of features we have)

5) **NLP**

Say I have a dataset

Review	Positive / Negative
The food is bad	0
The food is good	1

(This will be converted into vector)

when combined this turns to a matrix

mapping

$$\begin{bmatrix} [0.1 & 0.2 & 0.3] \\ [0.4 & 0.5 & 0.6] \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$