Perceptron, ANN, BackProp

Deep Neural Networks
Session 02

Pramod Sharma
pramod.sharma@prasami.com

2

Agenda

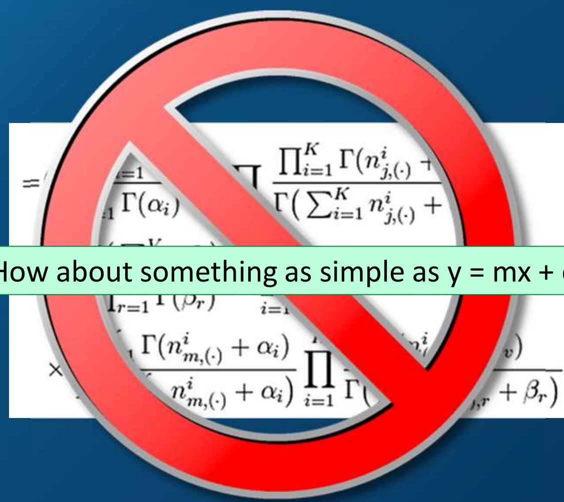
- Perceptron
- Single Layer Neural Network
- Overview of back propagation of errors

11/8/2025

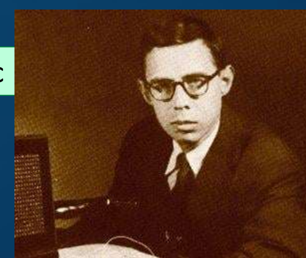
pra-sâmi

3

Solution to Equation of Perceptron



How about something as simple as y = mx + c



Frank Rosenblatt

pra-sâmi

4

To Play or Not to play…

id	Rains	Temp	Homework	Team Members	Equipment	Ground	Played
1	0	38	1	15	0	600	1
2	0	25	1	15	1	800	1
3	0	26	1	15	1	1000	1
4	5	27	1	10	1	600	0
5	20	23	0	8	1	1800	0
6	30	22	0	6	0	600	0

- Features:
  - Rains in millimeter
  - Temperature in ° C
  - Homework completed? – 0 : No; 1: Yes
  - Team members : How many team members are ready to play?
  - Is cricket equipment available?
  - Ground: per hour rent in Rupees/hour

pra-sâmi

Weights

- Feature Importance
  - Not every one is born equal
- To model, assign weights to each feature!

id	Rains	Temp	Homework	Team Members	Equipment	Ground	Played
1	0	38	1	15	0	600	1
2	0	25	1	15	1	800	1
3	0	26	1	15	1	1000	1
4	5	27	1	10	1	600	0
5	20	23	0	8	1	1800	0
6	30	22	0	6	0	600	0

- Values of each features are in different order of magnitude
  - Skewed summation highly in favor of Ground Cost
  - Scale the features between 0 and 1
- Note: Direction of influence
  - Variation in features have different bearing on the results
  - Team members ➔ higher the better
  - Ground cost ➔ lower the better

Evolution from MP Neuron to Perceptron

MP Neuron Model

- All inputs had same weights (effectively 1)
- Feature must be binary [0,1]
- Threshold 'w_0' could take limited values
- Binary Output [0, 1]
- No Preprocessing:
  - Not applicable (inputs are already binary)

Perceptron Model

- Perceptron model introduced different weights, allowing the model to learn the importance of each feature
- Accepts real-valued inputs, greatly expanding its applicability
- Threshold 'w_0' can take any value providing finer control over the decision boundary
- Still Binary Output [0, 1]
- Necessary to handle features on different scales:
  - Temperature in tens vs. Ground Rent in hundreds

7

Perceptron

- Loss Function:
  - Quantifies the model's error by comparing the prediction (ŷ) to the true label (y).
  - Loss = 0, if the prediction is correct (ŷ = y).
  - Loss = 1, if the prediction is incorrect (ŷ ≠ y)
- Optimize: The best-fitting line is the one that maximizes this total likelihood
  - Achieved by iteratively adjusting the weights (w_i) and bias (w_0)
- Aggregation Function:
  - $z = \sum x_i \,.\, w_i$
- Activation function g(z) is applied as follows:
  - If $z \geq w_0$ => ŷ = 1
  - If $z < w_0$ => ŷ = 0

11/8/2025

pra-sâmi

8

Perceptron – Data Preprocessing

- We will use 'Ground' and 'Team Members' as features, along with their associated weights, to generate a prediction.

id	Rains	Temp	Homework	Team Members	Equipment	Ground	Played
1	0	38	1	15	0	600	1
2	0	25	1	15	1	800	1
3	0	26	1	15	1	1000	1
4	5	27	1	10	1	600	0
5	20	23	0	8	1	1800	0
6	30	22	0	6	0	600	0

11/8/2025

pra-sâmi

9

Perceptron – Data Preprocessing

- Scaled Data (all columns to be between 0 and 1)

id	Rains	Temp	Homework	Team Members	Equipment	Ground	Played
1	0.00	0.00	1.00	1.00	0.00	1.00	1
2	0.00	0.81	1.00	1.00	1.00	0.83	1
3	0.00	0.75	1.00	1.00	1.00	0.67	1
4	-0.17	0.69	1.00	0.44	1.00	1.00	0
5	-0.67	0.94	0.00	0.22	1.00	0.00	0
6	-1.00	1.00	0.00	0.00	0.00	1.00	0

What about reverse correlation!

- Two option to address reverse correlation
  - Take negative of values
  - Use negative weight

11/8/2025

pra-sâmi

10

Perceptron – Weights

- Weights – consider importance of each of the feature

id	Threshold	Team Members		Ground		Calculations	Likely	Played	Loss
	w0	x1	w1	x2	w2	w0+x1*w1+x 2*w2	(y_hat)	(y)	(y-y_hat)^2
1	-1.00	1.00	1.10	1.00	1.00	1.10	1	1	0
2	-1.00	1.00	1.10	0.83	1.00	0.93	1	1	0
3	-1.00	1.00	1.10	0.67	1.00	0.77	1	1	0
4	-1.00	0.44	1.10	1.00	1.00	0.49	1	0	1
5	-1.00	0.22	1.10	0.00	1.00	-0.76	0	0	0
6	-1.00	0.00	1.10	1.00	1.00	0.00	1	0	1

11/8/2025

pra-sâmi

11

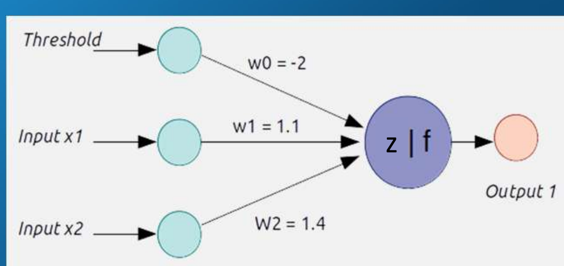
Perceptron – Weights and Loss

- Our best solution would be where ground truth and predicted values are same
- Loss is some function of ground truth and predicted values
- And we want it to be cumulative, Square of difference looks promising
  - $\ell(\hat{y}, y) = (y - \hat{y})^2$
  - Our overall loss was 2.
- By adjusting weights (w_1, w_2) and threshold (w_0) we can bring the loss to minimum (zero in this case)

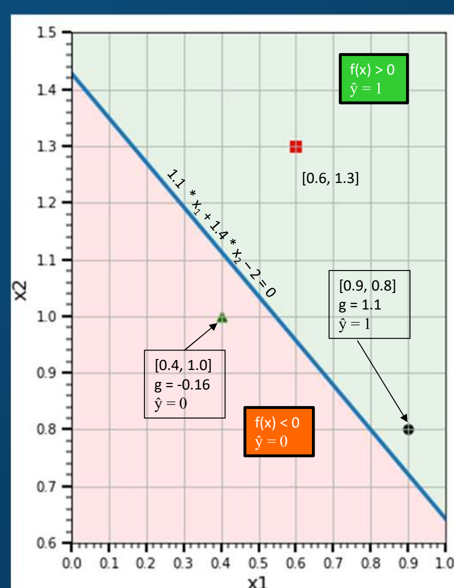
id	Threshold	Team Members		Ground		Calculations	Likely	Played	Loss
	w0	x1	w1	x2	w2	w0+x1*w1+x2*w2	(y_hat)	(y)	(y-y_hat)^2
1	-2.00	1.00	1.10	1.00	1.40	0.50	1	1	0
2	-2.00	1.00	1.10	0.83	1.40	0.27	1	1	0
3	-2.00	1.00	1.10	0.67	1.40	0.03	1	1	0
4	-2.00	0.44	1.10	1.00	1.40	-0.11	0	0	0
5	-2.00	0.22	1.10	0.00	1.40	-1.76	0	0	0
6	-2.00	0.00	1.10	1.00	1.40	-0.60	0	0	0

13

Perceptron



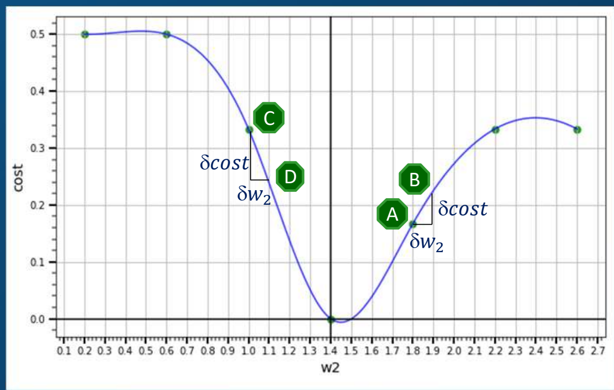
- We can represent : z = $w_0 + x_1 * w_1 + x_2 * w_2$
  - As z = $[x_1, x_2].\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + w_0$
- Given: W = $\begin{bmatrix} 1.1 \\ 1.4 \end{bmatrix}$ and w_0 = -2
  - z = $[x_1, x_2].\begin{bmatrix} 1.1 \\ 1.4 \end{bmatrix}$ - 2
  - z = 1.1 * x_1, +1.4 $* x_2$ - 2



14

Perceptron – Gradient Descent

- w_0, w_1, w_2 need to be adjusted to arrive at most optimal solution i.e. lowest point on the graph.
- Assume that w_0 is fixed at -2, and w_1 at 1.1 and w_2 varies from 0 to 3 (only one variable considered to make plotting simple)
- From point A to B, slope is positive hence w_2 value needs to be decreased
- From point C to D slope is negative hence w_2 needs to be increased.



11/8/2025

pra-sâmi

15

Perceptron – Activation Function

- So we based our entire calculations on:

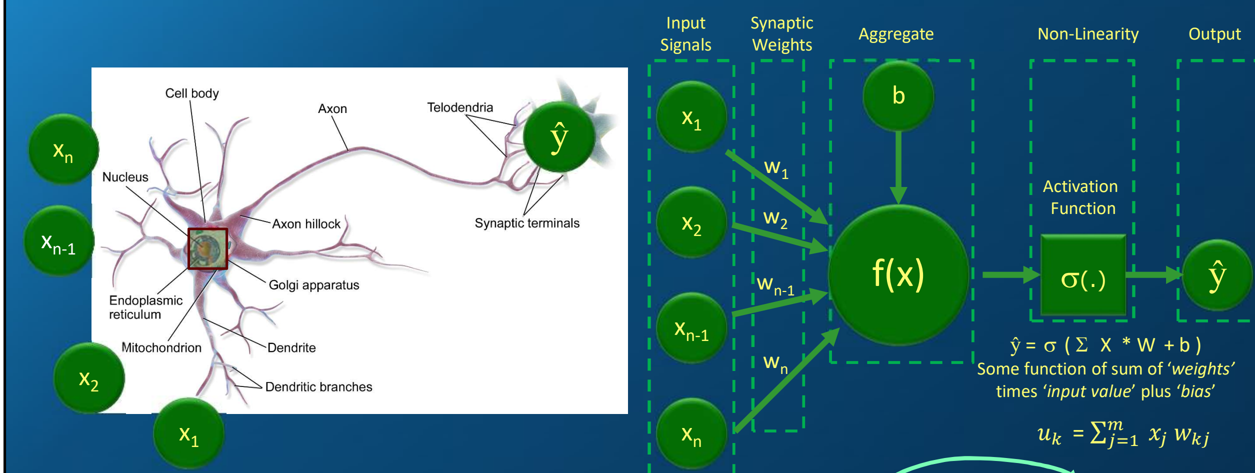
$$z = w_0 + x_1 * w_1 + x_2 * w_2$$

11/8/2025

pra-sâmi

16

Non Linear Activation function



$\hat{y} = \sigma\,(\,\Sigma\;\; X\;\; * W + b\,)$

Some function of sum of '*weights*' times '*input value*' plus '*bias*'

$$u_k = \sum_{j=1}^{m} x_j\, w_{kj}$$

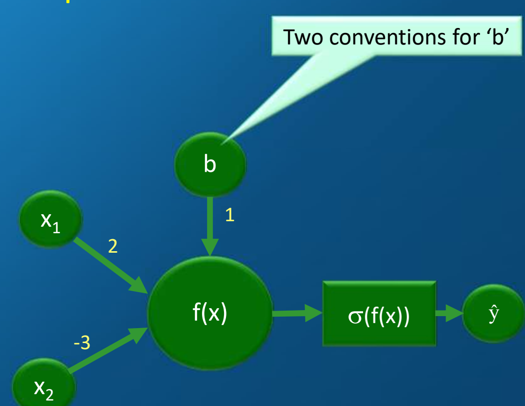
$$y_k = \sigma\,(u_k + b_k\,)$$

11/8/2025

pra-sâmi

17

Perceptron with non-linear activation function



- Given:
  - $W = \begin{bmatrix} 2 \\ -3 \end{bmatrix}$ and b = 1
  - $\hat{y} = \sigma\,(\,[x1, x2].\begin{bmatrix} 2 \\ -3 \end{bmatrix} + 1\,)$
  - $\hat{y} = \sigma\,(\underbrace{1 + 2 * x1 - 3 * x2}_{z})$
- $\hat{y} = \sigma\,(\,z\,)$;
- Lets use sigmoid function for σ.
  - $\hat{y} = \frac{1}{(1+e^{-z})}$

11/8/2025

pra-sâmi

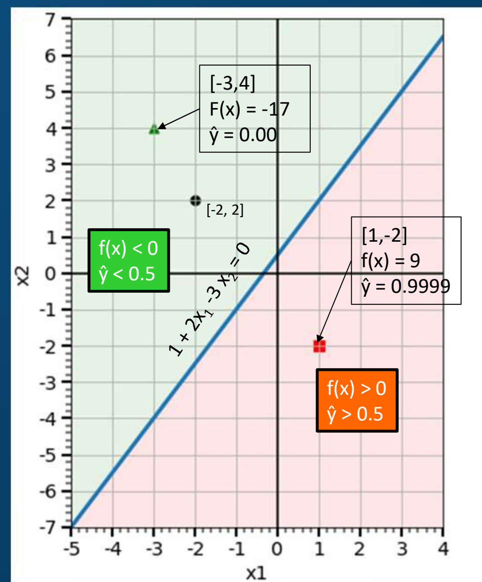
18

Perceptron with non-linear activation function

- $\hat{y} = \sigma\ (1 + 2 * x_1 - 3 * x_2)$

- For X = [-3, 4]
  - $\hat{y} = \sigma\ (1 + 2 * (-3)\ -3 * 4)$
  - $\hat{y} = \sigma\ (1 - 6\ -12)$
  - $\hat{y} = \sigma\ (-17)$
  - $\hat{y} = 0.0$

- Similarly, for X = [1, -2]
  - $\hat{y} = \sigma\ (1 + 2 * 1 - 3 * (-2))$
  - $\hat{y} = \sigma\ (1 + 2\ - 6)$
  - $\hat{y} = \sigma\ (9)$
  - $\hat{y} = 1.0$



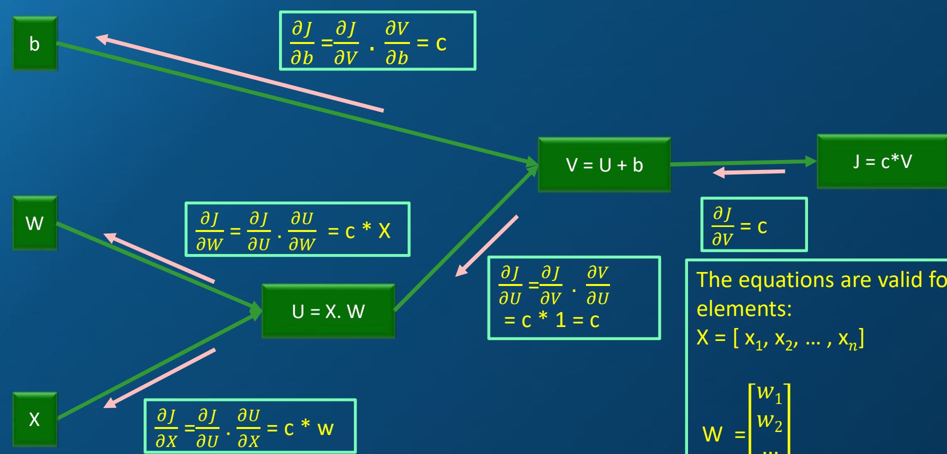
11/8/2025

pra-sâmi

20

Computational Graph

- Consider following hypothetical case, basic equation for single neuron :
  - $\hat{y}$ = X. W + b and Cost is some constant times $\hat{y}$; J = c * $\hat{y}$



The equations are valid for multiple elements:

X = [x_1, x_2, … , x_n]

$$W = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_n \end{bmatrix}$$

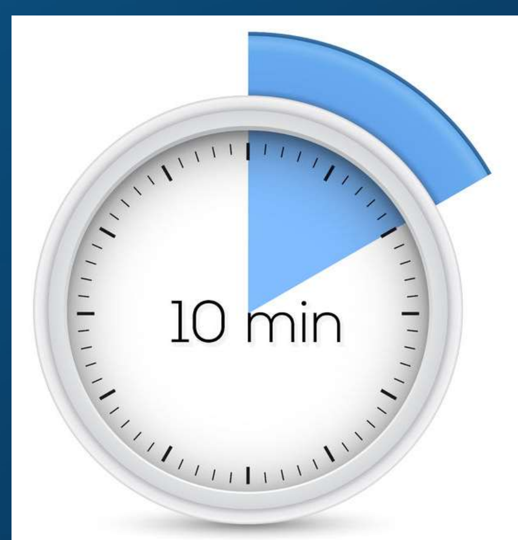
11/8/2025

pra-sâmi

21

Exercise 2 : Computational Graph

- Given a Cost Function J
  - J (w, x, b) = 3 *(b + x * w)
- Calculate $\frac{\partial J}{\partial W}$, $\frac{\partial J}{\partial x}$ and $\frac{\partial J}{\partial b}$
- Calculate slope at point :
  - b = 6
  - w = 3
  - x = 2



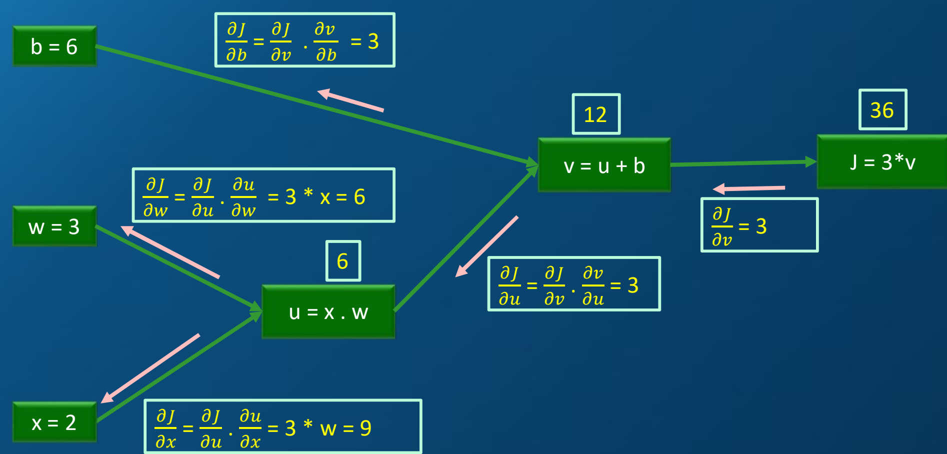
11/8/2025

pra-sâmi

22

Exercise - Solution

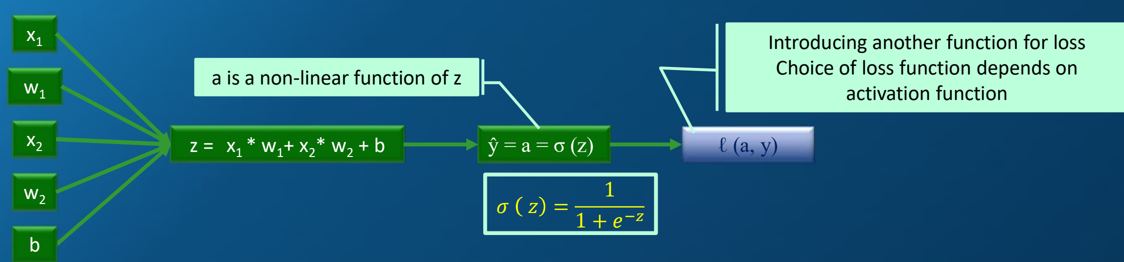
- Given a Cost Function J (w, x, b) = 3 *(b + w*x)



pra-sâmi

23

Common Loss Functions in Deep Learning



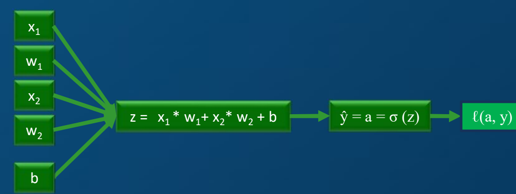
- Classification Tasks
  - **Goal:** Predict the probability that an instance belongs to a class
  - **Common Loss Function:** Cross-Entropy Loss
- Regression Tasks
  - **Goal:** Predict a continuous target value
  - **Common Loss Function:** Mean Squared Error (MSE)

pra-sâmi

24

Loss Functions & Maximum Likelihood

- What is a Loss Function?
  - A loss function, ℓ(ŷ, y), is a function used to evaluate a candidate solution (a set of model parameters).
  - Its primary purpose is to quantify the error between the model's prediction (ŷ) and the true value (y).
  - Helps to maximize or minimize the objective function



- The Role in Optimization
  - Find the parameters that minimize this loss function, thereby optimizing the model's performance.
- Connection to Maximum Likelihood
  - For probabilistic models (like classification), a fundamental approach is Maximum Likelihood Estimation (MLE).
  - We seek the model that makes the observed data (ground truth) most probable.
  - Under the MLE framework, minimizing the cross-entropy loss is equivalent to maximizing the likelihood.
  - Cross-entropy measures the divergence between the predicted probability distribution and the true data distribution.
- Binary Cross-entropy : $\ell(\hat{y}, y) = -[y * \log(\hat{y}) + (1 - y) * \log(1 - \hat{y})]$

25

Cost Function

- $\hat{y} = \sigma(\Sigma W * X + b)$
- Where $\sigma(z) = \frac{1}{1+e^{-z}}$
- Loss function:
  - A parameter which defines how good our outputs are i.e.
  - How far our predicted values 'ŷ' (y hat) were from ground truth 'y'
- For logistic regression
  - Loss(ŷ, y) = - (y . log ŷ + (1 – y) . log (1 - ŷ)
  - Loss function is for an instance
- Cost Function: Its a sum of losses for all instances
  - $J(W, b) = \frac{1}{m}(\Sigma \text{Loss}(\hat{y}, y))$
  - $= -\frac{1}{m}(\Sigma(y . \log \hat{y} + (1 - y) . \log(1 - \hat{y})))$

26

Forward and Back Propagation

x1, w1, x2, w2, b → $z = x_1 * w_1 + x_2 * w_2 + b$ → $\hat{y} = a = \sigma(z)$ → $\ell(a, y)$

$$\frac{\partial \ell}{\partial z} = \frac{\partial \ell}{\partial a} \cdot \frac{\partial a}{\partial z}$$
$$= \{ -\frac{y}{a} + \frac{1-y}{1-a} \} * a * (1-a)$$
$$= a - y$$

$$\frac{\partial \sigma(z)}{\partial z} = \sigma(z) . (1 - \sigma(z))$$
$$\frac{\partial a}{\partial z} = a . (1-a)$$

$$\frac{\partial \ell}{\partial a} = -\frac{y}{a} + \frac{1-y}{1-a}$$

$z = X * W + b$

$\hat{y} = a = \sigma(z)$

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

$\ell(a, y) = -[\, y * \log(a) + (1-y) * \log(1-a) \,]$

$\Longrightarrow$

$$\frac{\partial \ell}{\partial w1} = x_1 \cdot \frac{\partial \ell}{\partial z} = x_1 . (a-y)$$
$$\frac{\partial \ell}{\partial w2} = x_2 \cdot \frac{\partial \ell}{\partial z} = x_2 . (a-y)$$
$$\frac{\partial \ell}{\partial b} = \frac{\partial \ell}{\partial z} = (a-y)$$

$\Longrightarrow$

$$w_1 = w_1 - \alpha * \frac{\partial \ell}{\partial w_1} = w_1 - \alpha * x_1 * (a-y)$$
$$w_2 = w_2 - \alpha * \frac{\partial \ell}{\partial w_2} = w_2 - \alpha * x_2 * (a-y)$$
$$b = b - \alpha * \frac{\partial \ell}{\partial b} = b - \alpha * (a-y)$$

Where α is learning rate. The cost function is

$$J(W, b) = \frac{1}{m} * (\Sigma\, \ell(a, y))$$

Hence $\frac{\partial J}{\partial w_1} = \frac{1}{m} * (\Sigma \frac{\partial \ell(a, y)}{\partial w_1})$

11/8/2025

pra-sâmi

27

So where are the hidden layers!!!

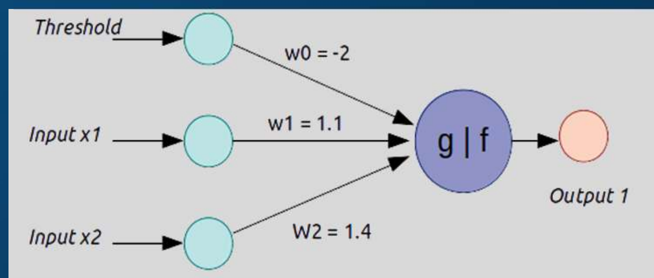
11/8/2025

pra-sâmi

28

Hidden Layers

id	Threshold	Team Members		Ground	
	x0	x1	w1	x2	w2
1	-2.00	1.00	1.10	1.00	1.40
2	-2.00	1.00	1.10	0.83	1.40
3	-2.00	1.00	1.10	0.67	1.40
4	-2.00	0.44	1.10	1.00	1.40
5	-2.00	0.22	1.10	0.00	1.40
6	-2.00	0.00	1.10	1.00	1.40



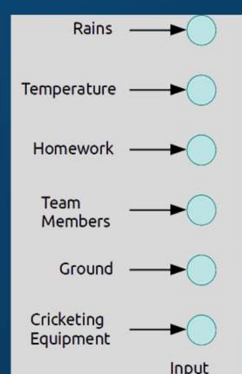
11/8/2025

pra-sâmi

29

Hidden Layers

id	Rains	Temp	Homework	Team Members	Equipment	Ground	Played
1	0.00	0.00	1.00	1.00	0.00	1.00	1
2	0.00	0.81	1.00	1.00	1.00	0.83	1
3	0.00	0.75	1.00	1.00	1.00	0.67	1
4	-0.17	0.69	1.00	0.44	1.00	1.00	0
5	-0.67	0.94	0.00	0.22	1.00	0.00	0
6	-1.00	1.00	0.00	0.00	0.00	1.00	0



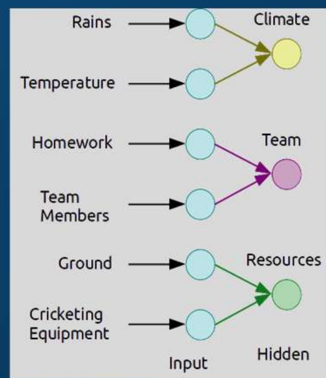
11/8/2025

pra-sâmi

30

Hidden Layers

id	Rains	Temp	Homework	Team Members	Equipment	Ground	Played
1	0.00	0.00	1.00	1.00	0.00	1.00	1
2	0.00	0.81	1.00	1.00	1.00	0.83	1
3	0.00	0.75	1.00	1.00	1.00	0.67	1
4	-0.17	0.69	1.00	0.44	1.00	1.00	0
5	-0.67	0.94	0.00	0.22	1.00	0.00	0
6	-1.00	1.00	0.00	0.00	0.00	1.00	0



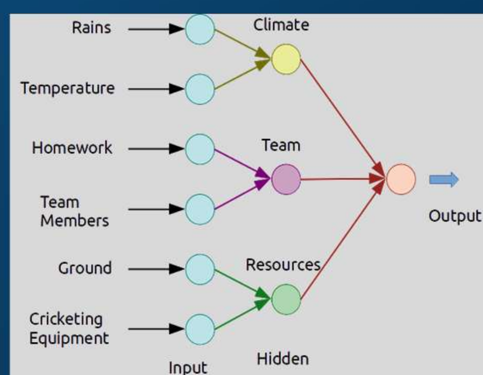
11/8/2025

pra-sâmi

31

Hidden Layers

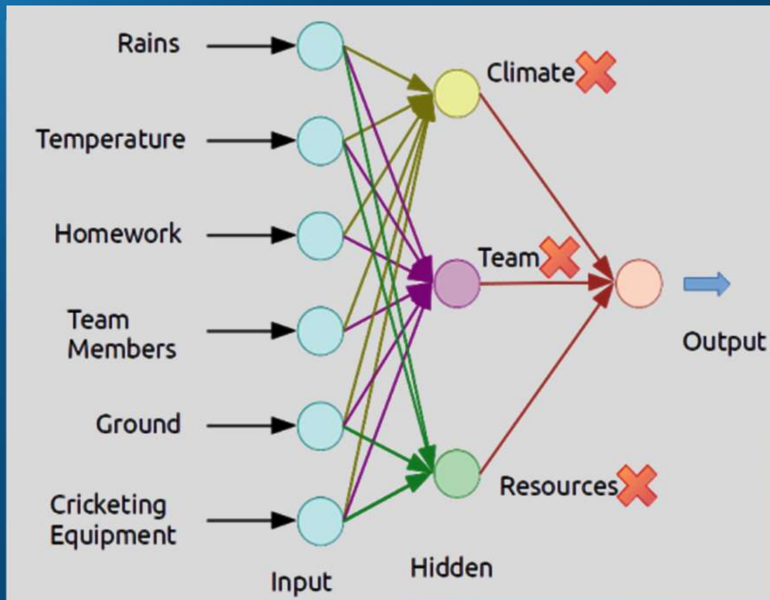
id	Rains	Temp	Homework	Team Members	Equipment	Ground	Played
1	0.00	0.00	1.00	1.00	0.00	1.00	1
2	0.00	0.81	1.00	1.00	1.00	0.83	1
3	0.00	0.75	1.00	1.00	1.00	0.67	1
4	-0.17	0.69	1.00	0.44	1.00	1.00	0
5	-0.67	0.94	0.00	0.22	1.00	0.00	0
6	-1.00	1.00	0.00	0.00	0.00	1.00	0



11/8/2025

pra-sâmi

32 Hidden Layers



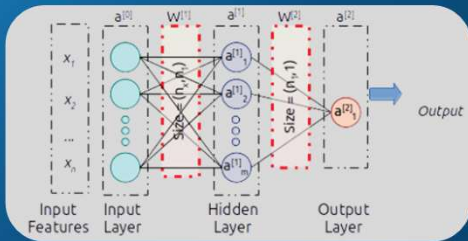
11/8/2025 pra-sâmi

33 Hidden Layers

11/8/2025 pra-sâmi

Two Major Conventions

34

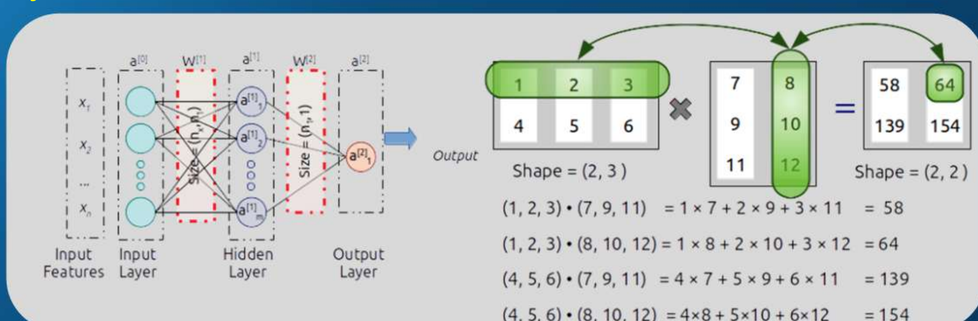


11/8/2025

pra-sâmi

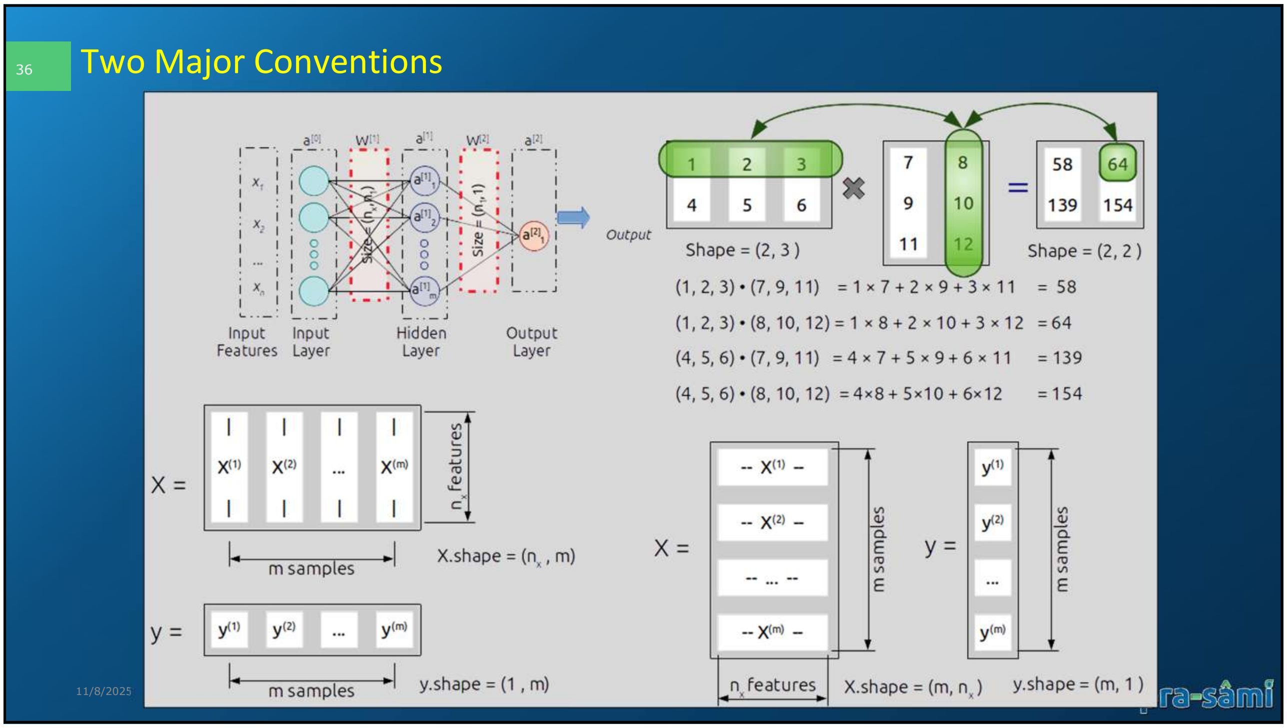
Two Major Conventions

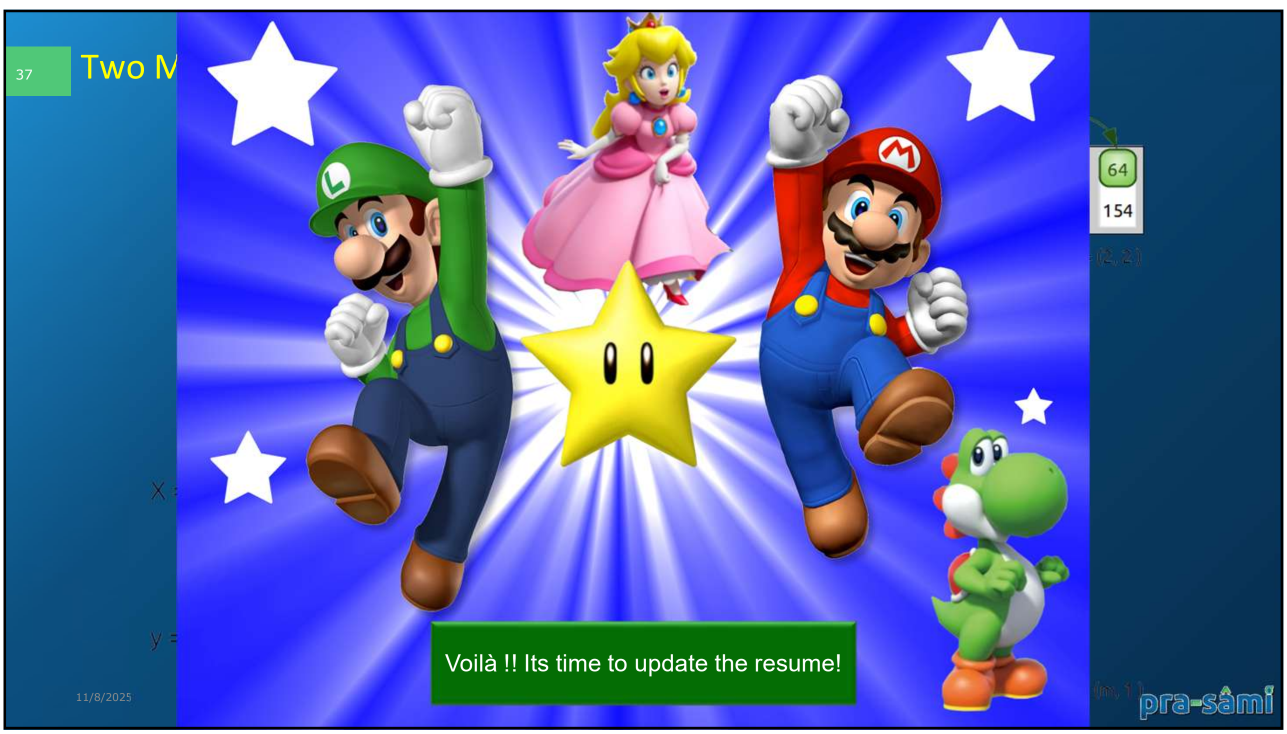
35



11/8/2025

pra-sâmi


36
Two Major Conventions
Input Features
Input Layer
Hidden Layer
Output Layer
Output
Size = (n_x, n_1)
Size = (n_1, 1)
1 2 3
4 5 6
Shape = (2, 3)
7 8
9 10
11 12
58 64
139 154
Shape = (2, 2)
(1, 2, 3) • (7, 9, 11) = 1 × 7 + 2 × 9 + 3 × 11 = 58
(1, 2, 3) • (8, 10, 12) = 1 × 8 + 2 × 10 + 3 × 12 = 64
(4, 5, 6) • (7, 9, 11) = 4 × 7 + 5 × 9 + 6 × 11 = 139
(4, 5, 6) • (8, 10, 12) = 4×8 + 5×10 + 6×12 = 154
X =
n_x features
m samples
X.shape = (n_x , m)
y =
m samples
y.shape = (1 , m)
X =
m samples
n_x features
X.shape = (m, n_x)
y =
m samples
y.shape = (m, 1)
11/8/2025
pra-sâmi


37
Two M
Voilà !! Its time to update the resume!
11/8/2025
pra-sâmi

38

Next Session - Coding Perceptron Model in Python

11/8/2025

39

11/8/2025

ADDITIONAL MATERIAL

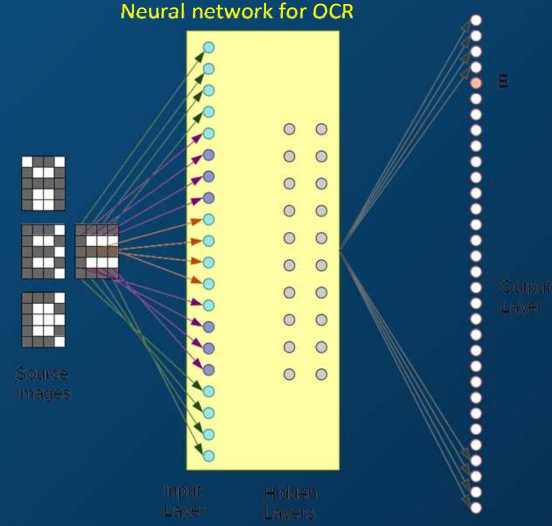
Applications

41

Applications

- The properties of neural networks define where they are useful
- Typical Network
  - Can learn complex mappings from inputs to outputs, based solely on samples
  - Difficult to analyse
  - Firm predictions about neural network behaviour difficult;
    - Unsuitable for safety-critical applications.
  - Require limited understanding from trainer, who can be guided by heuristics

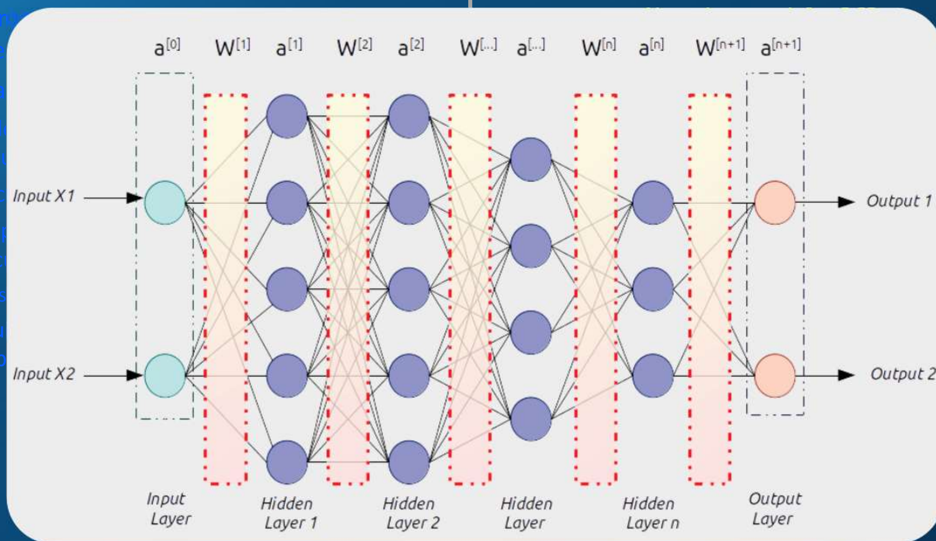
Neural network for OCR



11/8/2025

42

Applications



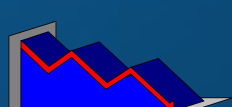
11/8/2025

pra-sâmi

43

Applications

- Stock market prediction
  - "Technical trading" refers to trading based solely on known statistical parameters; e.g. previous price
  - Neural networks have been used to attempt to predict changes in prices.
  - Difficult to assess success or otherwise
    - Since companies using these techniques are reluctant to disclose information.



- Mortgage assessment
  - Assess risk of lending to an individual
  - Difficult to decide on marginal cases
  - Neural networks have been trained to make decisions, based upon the opinions of expert underwriters
  - Neural network produced a 12% reduction in delinquencies compared with human experts

11/8/2025

pra-sâmi

44

Applications

Stock m

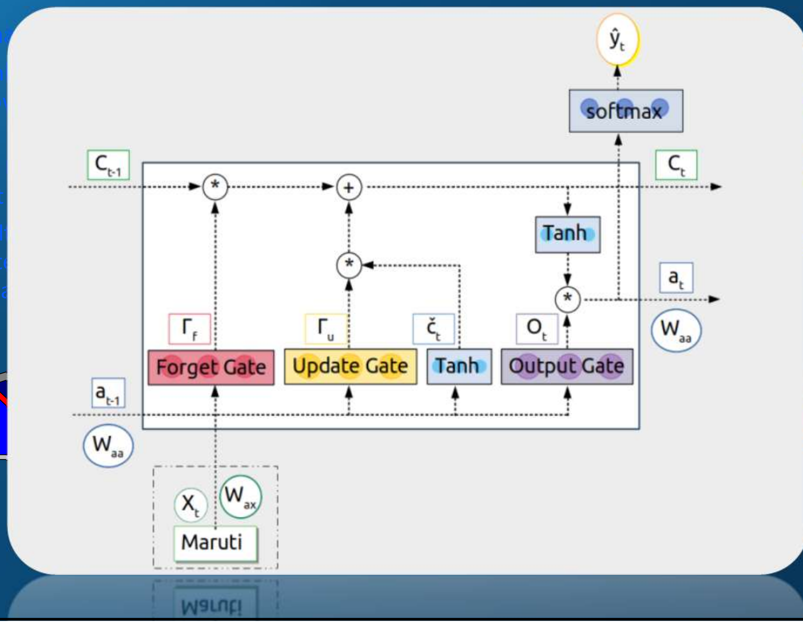
"Techn
on kno
price

Neural
predict

Difficul
these t
informa

dividual.
l cases.
ained to make
nions of expert

2% reduction in
human experts.



pra-sâmi

45

Applications

- ALVINN: Autonomous Land Vehicle In a Neural Network

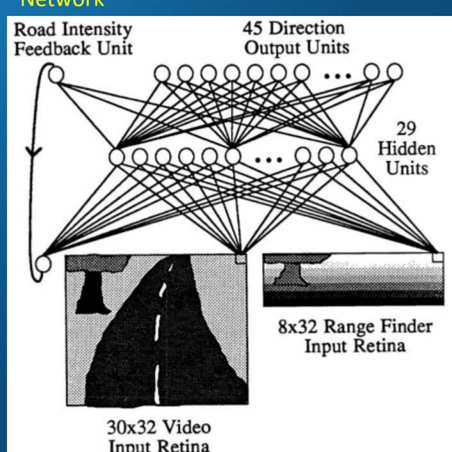
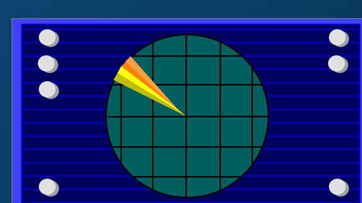


Figure 1: ALVINN Architecture

- Sonar target recognition
  - Distinguish mines from rocks on sea-bed
  - The neural network is provided with a large number of parameters which are extracted from the sonar signal.
  - The training set consists of sets of signals from rocks and mines.



pra-sâmi

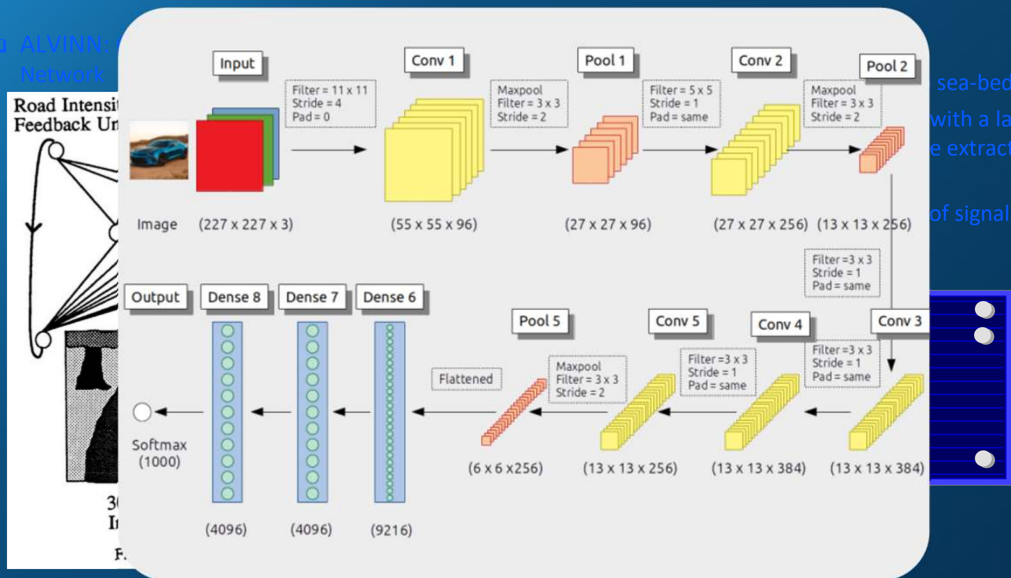
46

Applications

- ALVINN:
  Network

sea-bed
with a large
e extracted from

of signals from



11/8/2025

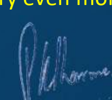
pra-sâmi

47

Applications

- Engine management
  - The behavior of a car engine is influenced by a large number of parameters
    - temperature at various points
    - fuel/air mixture
    - lubricant viscosity.
  - Major companies have used neural networks to dynamically tune an engine depending on current settings.

- Signature recognition
  - Each person's signature is different.
  - There are structural similarities which are difficult to quantify.
  - Recognizes signatures to a high level of accuracy.
  - Considers speed in addition to gross shape
  - Makes forgery even more difficult.



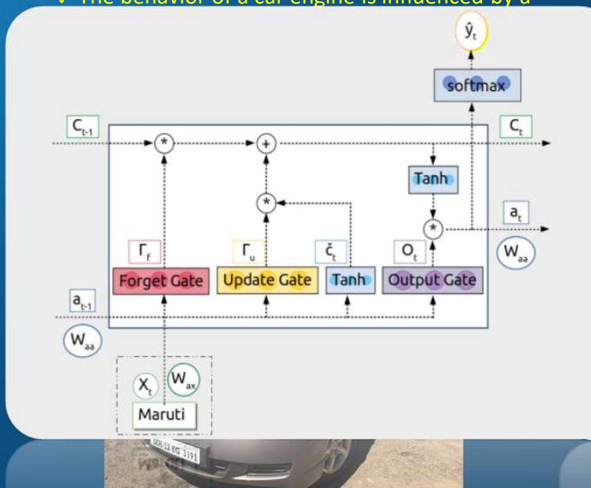
11/8/2025

pra-sâmi

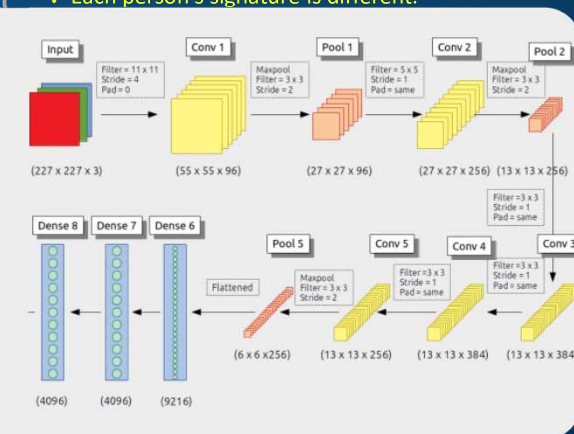
48

Applications

- Engine management
  - The behavior of a car engine is influenced by a



- Signature recognition
  - Each person's signature is different.



11/8/2025

pra-sâmi

49

Derivation of Sigmoid

$$
\begin{aligned}
\partial a &= \partial\sigma(z) \\
&= \frac{\partial}{\partial z}\left[\frac{1}{1+e^{-z}}\right] \\
&= \frac{\partial}{\partial z}(1+e^{-z})^{-1} \\
&= -(1+e^{-z})^{-2}(-e^{-z}) \\
&= \frac{e^{-z}}{(1+e^{-z})^{2}} \\
&= \frac{1}{1+e^{-z}} \circ \frac{e^{-z}}{1+e^{-z}} \\
&= \frac{1}{1+e^{-z}} \circ \frac{(1+e^{-z})-1}{1+e^{-z}} \\
&= \frac{1}{1+e^{-z}} \circ \left[\frac{1+e^{-z}}{1+e^{-z}} - \frac{1}{1+e^{-z}}\right] \\
&= \frac{1}{1+e^{-z}} \circ \left[1 - \frac{1}{1+e^{-z}}\right] \\
&= \sigma(z) \circ (1-\sigma(z)) \\
&= a \circ (1-a)
\end{aligned}
$$

11/8/2025

pra-sâmi