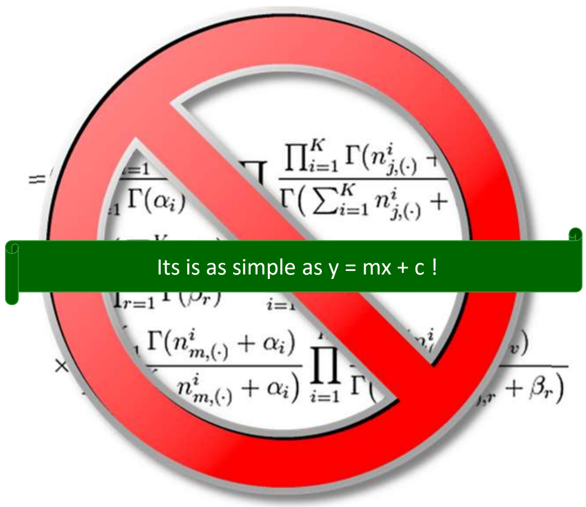
3

Solution to Equation of Perceptron



Ian Goodfellow
Yoshua Bengio
Aaron Courville

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

5

Weights

- Each of the feature has different importance
- To assign importance to each of the feature, we use weights!
- Values of each features are in different order of magnitude
  - Summation is not going to work
  - Scale the features between 0 and 1

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

- Note:
  - Variation in features have different bearing on the results
  - Team members ➔ higher the better
  - Ground cost ➔ lower the better

6

Perceptron

- In MP Neuron Model,
  - All inputs had same weights
  - Threshold 'w_0' could take limited values
  - Every feature needed to be [0,1]
- Perceptron model introduced different weights to different inputs features
- Real values are also accepted
  - Temperatures are in tens and ground rent is in hundreds.
  - Min – Max – Scaler to compensate for huge difference is values
- Threshold 'w_0' can take any value
- Outputs are still [0, 1]

7

Perceptron

- Loss Function:
  - A correction is applied on the outputs
  - To adjust values of 'w_i' to reach right results
  - It would also give us indications of what weights to be fixed to arrive at the solution

- Activation function g(x) is applied as follows:
  - If $\sum x_i \cdot w_i \geq w_0$ => ŷ = 1
  - If $\sum x_i \cdot w_i < w_0$ => ŷ = 0

pra-sâmi

8

Perceptron – Data Preprocessing

- Lets consider "Ground" and "Team Members" as features and its associated weights to arrive at the solution.

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

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

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+x2*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

pra-sâmi

11

Perceptron – Weights and Loss

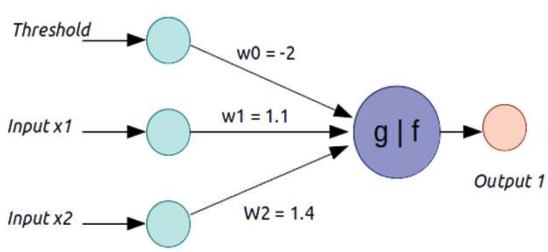
- Our best solution would be where ground truth and predicted values are same
- Loss is some function of ground truth and predicted values
- And we want it to be cumulative, Square of difference looks promising
  - $\ell(\hat{y}, y) = (y - \hat{y})^2$
  - Our overall loss was 2.
- By adjusting weights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

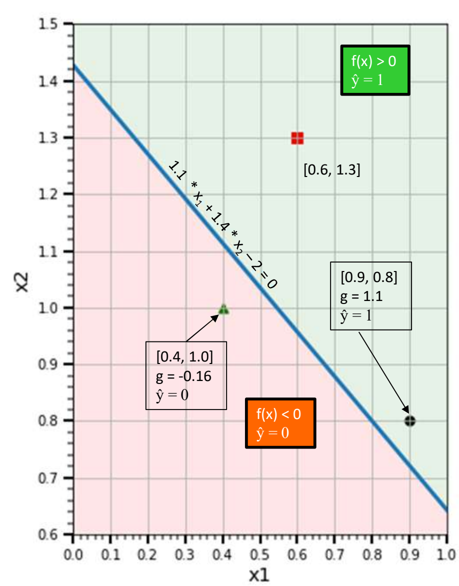
pra-sâmi

13

Perceptron



- We can represent : g = $w_0 + x_1 * w_1 + x_2 * w_2$
  - As g = $[x_1, x_2].\begin{bmatrix} W_1 \\ W_2 \end{bmatrix} + w_0$
- Given: W = $\begin{bmatrix} 1.1 \\ 1.4 \end{bmatrix}$ and w_0 = -2
  - g = $[x_1, x_2].\begin{bmatrix} 1.1 \\ 1.4 \end{bmatrix}$ - 2
  - g = $1.1 * x_1, +1.4 * x_2$ - 2

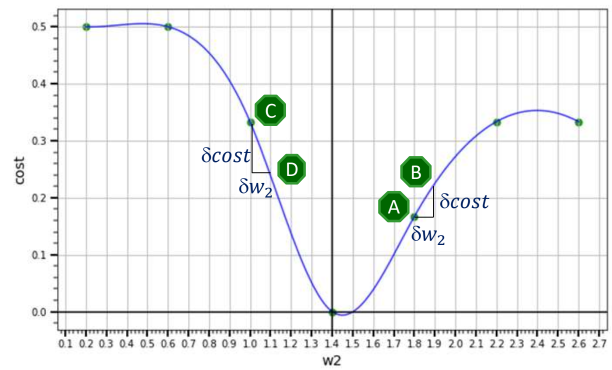


pra-sâmi

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.



11/17/2023

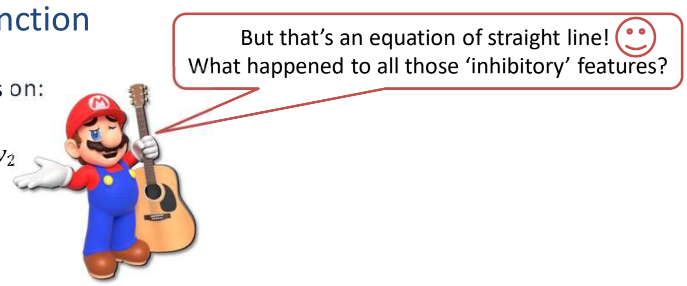
pra-sâmi

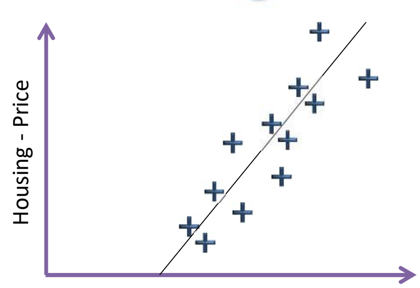
15

Perceptron – Activation Function

- So we based our entire calculations on:

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



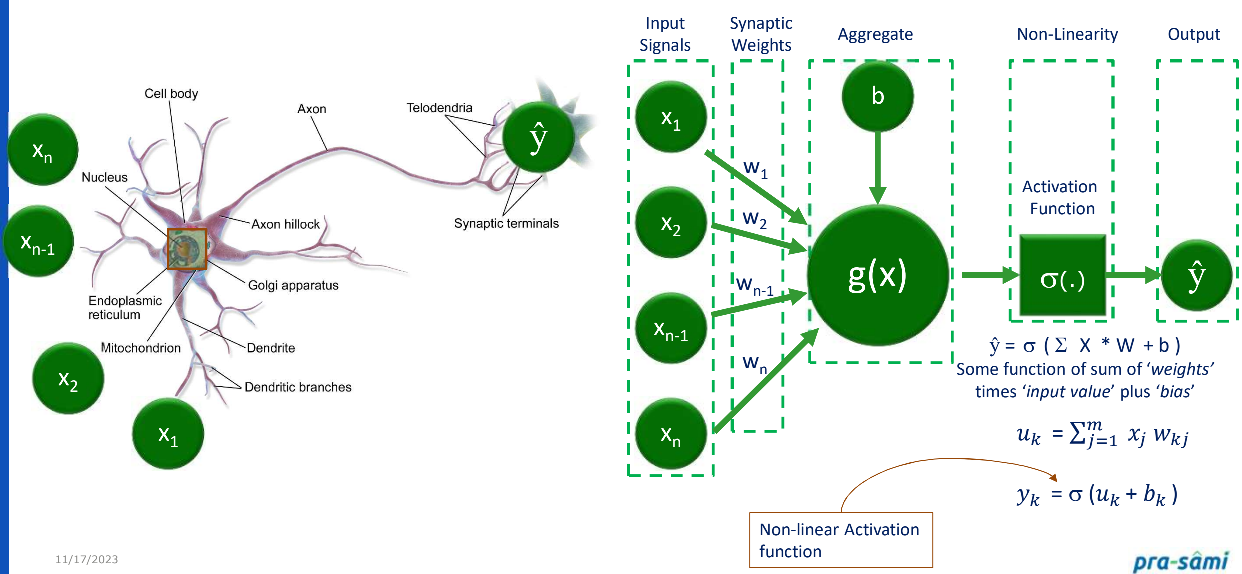


11/17/2023

pra-sâmi

16

Non Linear Activation function

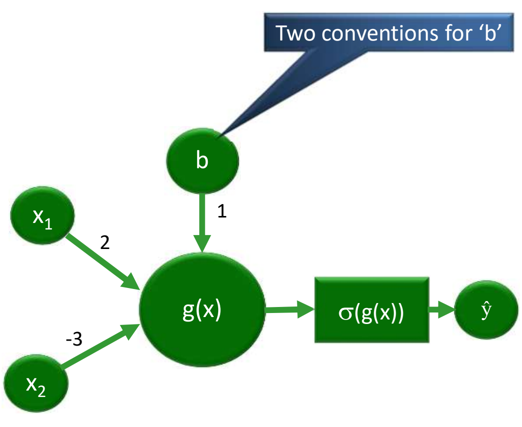


11/17/2023

pra-sâmi

17

Perceptron with non-linear activation function



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

11/17/2023

pra-sâmi

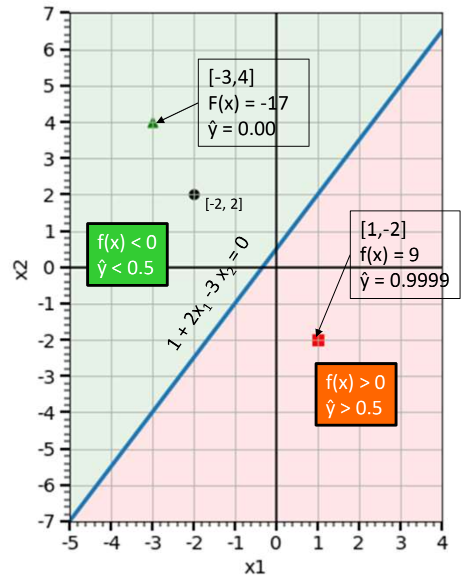
18

Perceptron with non-linear activation function

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

- For X = [-3, 4]
  - ŷ = σ (1 + 2 * (-3) -3 * 4)
  - ŷ = σ (1 - 6 -12)
  - ŷ = σ (-17)
  - ŷ = 0.0

- Similarly, for X = [1, -2]
  - ŷ = σ (1 + 2 * 1 - 3 * (-2))
  - ŷ = σ (1 + 2 - 6)
  - ŷ = σ (9)
  - ŷ = 1.0



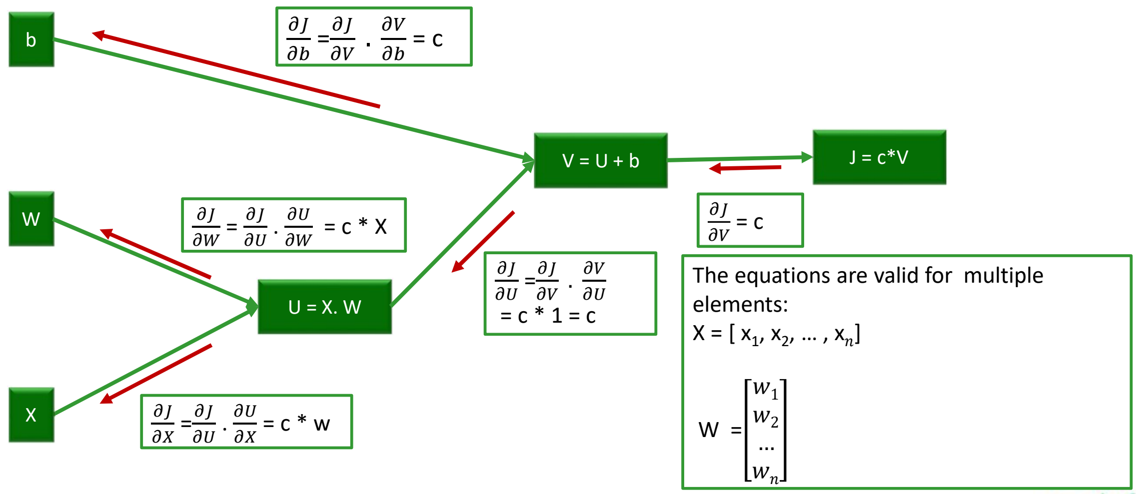
11/17/2023

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 * $\hat{y}$



11/17/2023

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

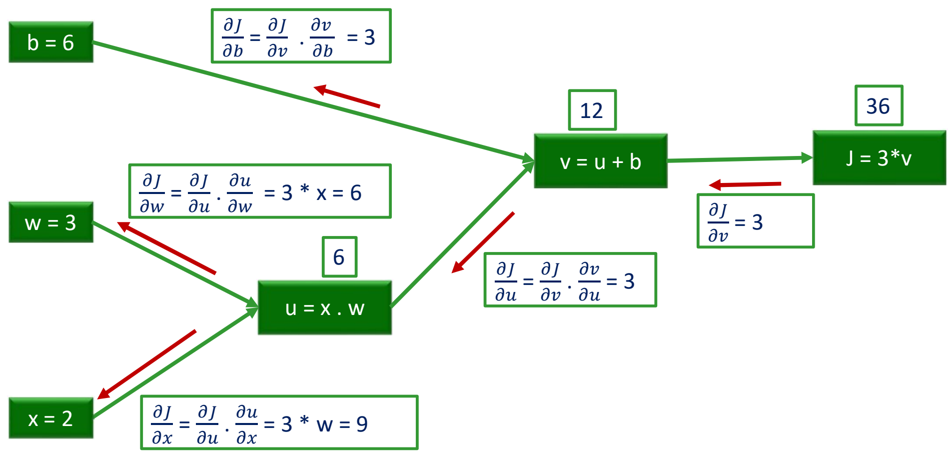
11/17/2023

pra-sâmi

22

Exercise - Solution

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

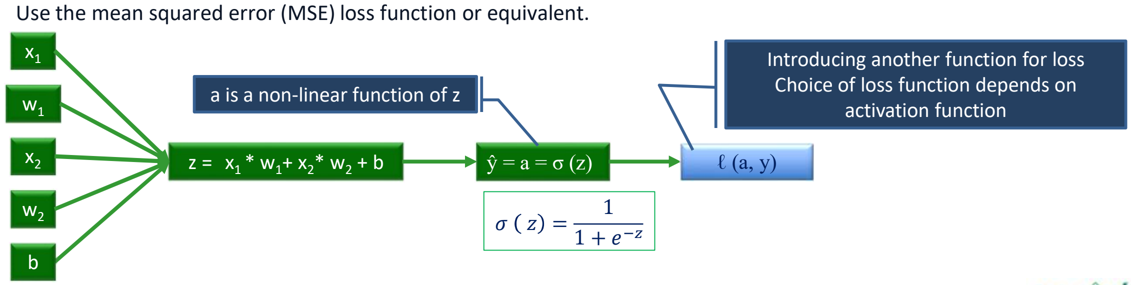


11/17/2023

23

Consider Single Path… MLE

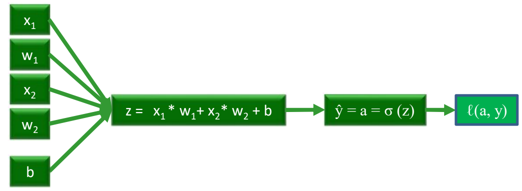
- Maximum likelihood estimation, or MLE, is a framework for inference for finding the best statistical estimates of parameters from historical training data
  - Exactly what we are trying to do with the neural network
- In Classification, output is probability of it belonging to a class
  - Maximum likelihood estimation, seeks a set of model weights that minimize the difference between the predicted probability distribution and the Ground Truth [cross-entropy]
- In Regression problems:
  - Use the mean squared error (MSE) loss function or equivalent.



11/17/2023

24

Consider Single Path… Loss Function



- A function used to evaluate a candidate solution
- Helps to maximize or minimize the objective function
- Estimates how closely the distribution of predictions made by a model matches the ground truth (maximum likelihood)
- Under maximum likelihood framework , the error between two probability distributions is measured using cross-entropy
  - Hence $\ell(\hat{y}, y) = -[y * \log(\hat{y}) + (1 - y) * \log(1 - \hat{y})]$

pra-sâmi

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
  - $Loss(\hat{y}, y) = -(y . \log \hat{y} + (1 - y) . \log(1 - \hat{y})$
  - Loss function is for an instance
  - In case of binary classification, $Loss(\hat{y}, y) = - y . \log \hat{y}$
- Cost Function: Its a sum of losses for all instances
  - $J(W, b) = \frac{1}{m}(\Sigma Loss(\hat{y}, y))$
  - $= - \frac{1}{m}(\Sigma (y . \log \hat{y} + (1 - y) . \log(1 - \hat{y})))$
- For binary classification:
  - $J(W, b) = \frac{1}{m}(\Sigma Loss(\hat{y}, y))$
  - $= - \frac{1}{m}(\Sigma (y . \log \hat{y})$

pra-sâmi

26

Forward and Back Propagation

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

$\frac{\partial \ell}{\partial z} = \frac{\partial \ell}{\partial a} \cdot \frac{\partial a}{\partial z}$

$= \{ -y/a + (1-y)/(1-a) \} * a * (1-a)$

$= a - y$

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

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

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

$z = X * W + b$

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

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

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

For binary classification:

$\ell(a, y) = -y * \log(a)$

$\Rightarrow$

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

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

$\frac{\partial \ell}{\partial b} = \frac{\partial \ell}{\partial z} = (a-y)$

$\Rightarrow$

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

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

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

Where α is learning rate. The cost function is

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

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

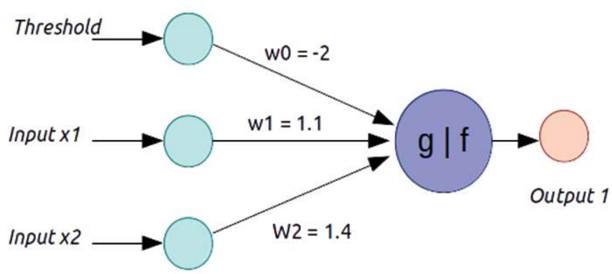
27

So where are the hidden layers!!!

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



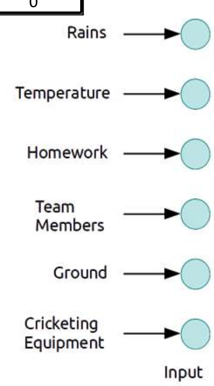
11/17/2023

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



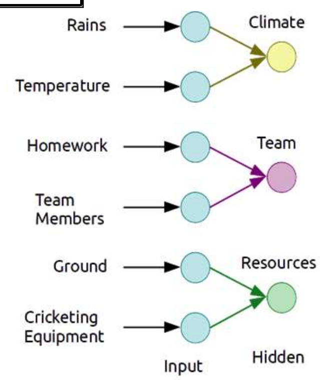
11/17/2023

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



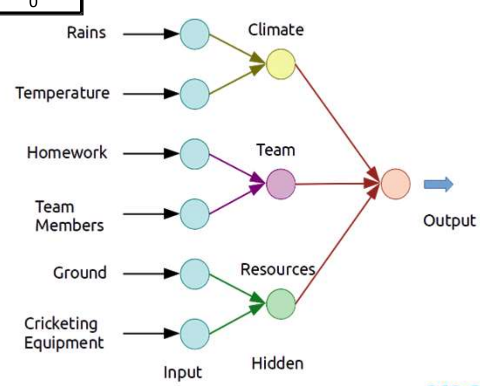
11/17/2023

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

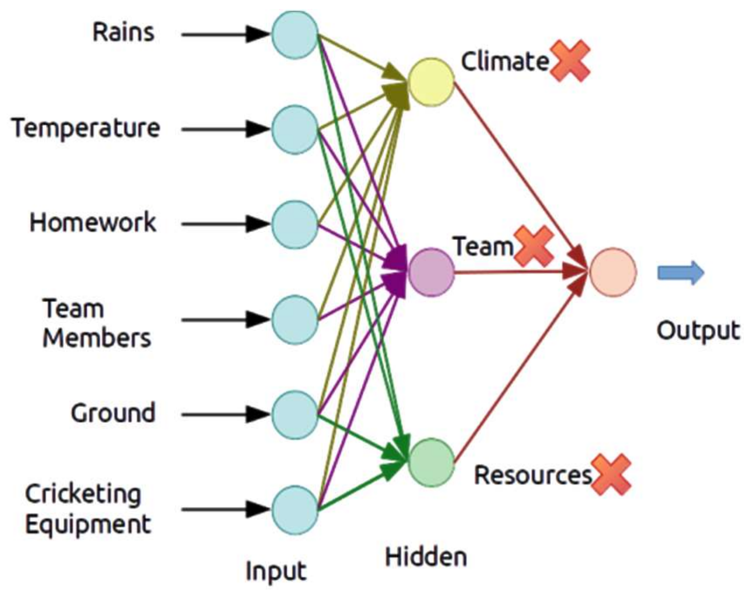


11/17/2023

pra-sâmi

32

Hidden Layers



pra-sâmi

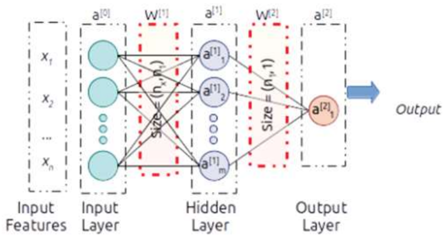
33

Hidden Layers

pra-sâmi

34

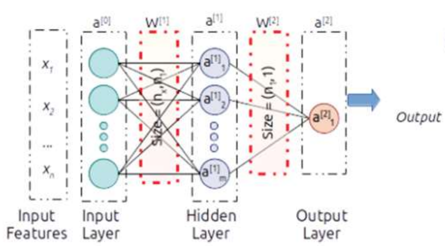
Two Major Conventions

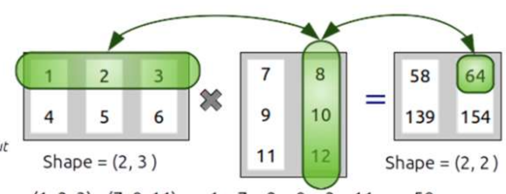


pra-sâmi

35

Two Major Conventions





(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

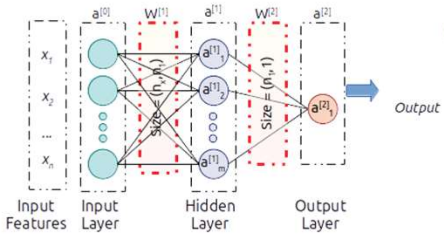
(4, 5, 6) • (7, 9, 11) = 4 × 7 + 5 × 9 + 6 × 11 = 139

(4, 5, 6) • (8, 10, 12) = 4×8 + 5×10 + 6×12 = 154

pra-sâmi

36

Two Major Conventions



$$\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix} \times \begin{bmatrix}7 & 8\\ 9 & 10\\ 11 & 12\end{bmatrix} = \begin{bmatrix}58 & 64\\ 139 & 154\end{bmatrix}$$

Shape = (2, 3)   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

$X = \begin{bmatrix} | & | & & | \\ X^{(1)} & X^{(2)} & \dots & X^{(m)} \\ | & | & & | \end{bmatrix}$ n_x features, m samples   X.shape = (n_x , m)

$y = \begin{bmatrix} y^{(1)} & y^{(2)} & \dots & y^{(m)} \end{bmatrix}$ m samples   y.shape = (1 , m)

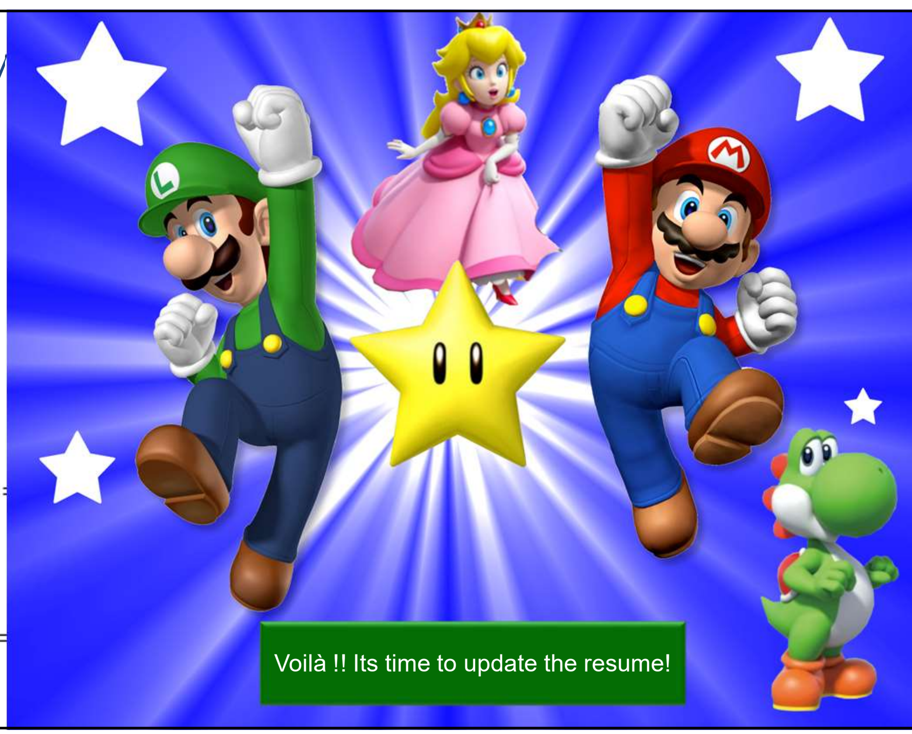
$X = \begin{bmatrix} \text{-- } X^{(1)} \text{ --} \\ \text{-- } X^{(2)} \text{ --} \\ \text{-- ... --} \\ \text{-- } X^{(m)} \text{ --} \end{bmatrix}$ m samples, n_x features   X.shape = (m, n_x)

$y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \dots \\ y^{(m)} \end{bmatrix}$ m samples   y.shape = (m, 1)

11/17/2023   pra-sâmi

37

Two M



64

154

(2, 2)

X =

y =

(m, 1)

11/17/2023   pra-sâmi

Reflect…

- How many type of layers Deep Learning Algorithms have?
  a. 2
  b. 3
  c. 4
  d. 5
- Answer : b

- The first layer is called the?
  a. Input Layer
  b. Output Layer
  c. Hidden Layer
  d. None of The Above
- Answer : a

- Which of the following is/are Limitations of deep learning?
  a. Data labeling
  b. Obtain huge training datasets
  c. Both A and B
  d. None of the above
- Answer : c

- Deep learning algorithms are _______ more accurate than machine learning algorithm in image classification.
  a. 33%
  b. 37%
  c. 40%
  d. 41%
- Answer : d

Reflect…

- In which of the following applications can we use deep learning to solve the problem
  a. Protein structure prediction
  b. Prediction of chemical reactions
  c. Detection of exotic particles
  d. All of the above
- Answer : d

- The number of nodes in the input layer is 10 and the hidden layer is 5. The maximum number of connections from the input layer to the hidden layer are:
  a. 50
  b. less than 50
  c. more than 50
  d. It is an arbitrary value
- Answer : a

- What is a perceptron?
  a. A type of neural network
  b. A reinforcement learning algorithm
  c. A clustering algorithm
  d. A regression algorithm
- Answer : a

- Who is credited with the invention of the perceptron?
  a. Geoffrey Hinton
  b. Yann LeCun
  c. Frank Rosenblatt
  d. Andrew Ng
- Answer : c

40

Reflect…

- What is the basic building block of a perceptron?
  a. Neuron
  b. Weight
  c. Activation function
  d. Bias
- Answer: a

- In a perceptron, what is the purpose of the activation function?
  a. To compute the weighted sum of inputs
  b. To introduce non-linearity
  c. To adjust the weights during training
  d. To add a bias to the output
- Answer: b

- What is the primary purpose of training a perceptron?
  a. To optimize the activation function
  b. To minimize the error in the output
  c. To increase the number of neurons
  d. To add more layers to the network
- Answer: b

- In a binary classification problem, what is the output of a perceptron?
  a. Real number
  b. Probability
  c. Binary value (0 or 1)
  d. Vector
- Answer: c

11/17/2023

pra-sâmi

41

Reflect…

- What is the perceptron learning rule used for?
  a. a. Updating weights to reduce prediction error
  b. b. Adjusting the learning rate during training
  c. c. Initializing weights in the network
  d. d. Selecting the appropriate activation function
- Answer: a

- What happens if a perceptron is unable to learn a linearly separable function?
  a. a. It converges quickly
  b. b. It converges slowly
  c. c. It never converges
  d. d. It always converges
- Answer: c

- Which of the following statements about the perceptron is true?
  a. a. It can only be used for linearly separable problems
  b. b. It is suitable for any type of problem
  c. c. It can only have one layer
  d. d. It has no weights
- Answer: a

- What is the main limitation of a single-layer perceptron?
  a. a. It cannot learn non-linearly separable functions
  b. b. It requires a large amount of training data
  c. c. It is computationally expensive
  d. d. It is not suitable for classification tasks
- Answer: a

11/17/2023

pra-sâmi

42

Next Session - Coding Perceptron Model in Python

11/17/2023

pra-sâmi

43

11/17/2023

pra-sâmi

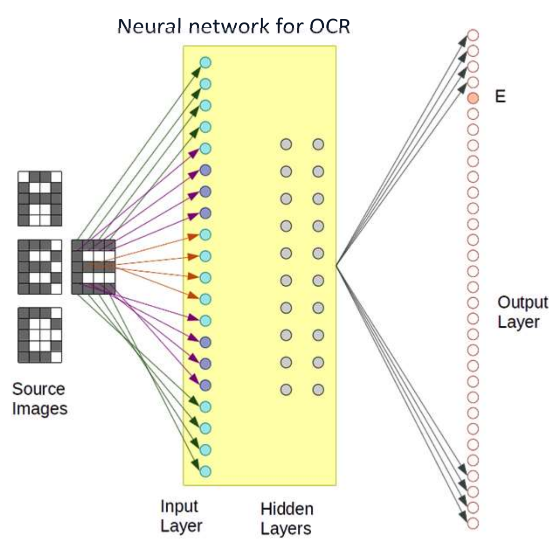
EXTRA MATERIAL

Applications

45

Applications

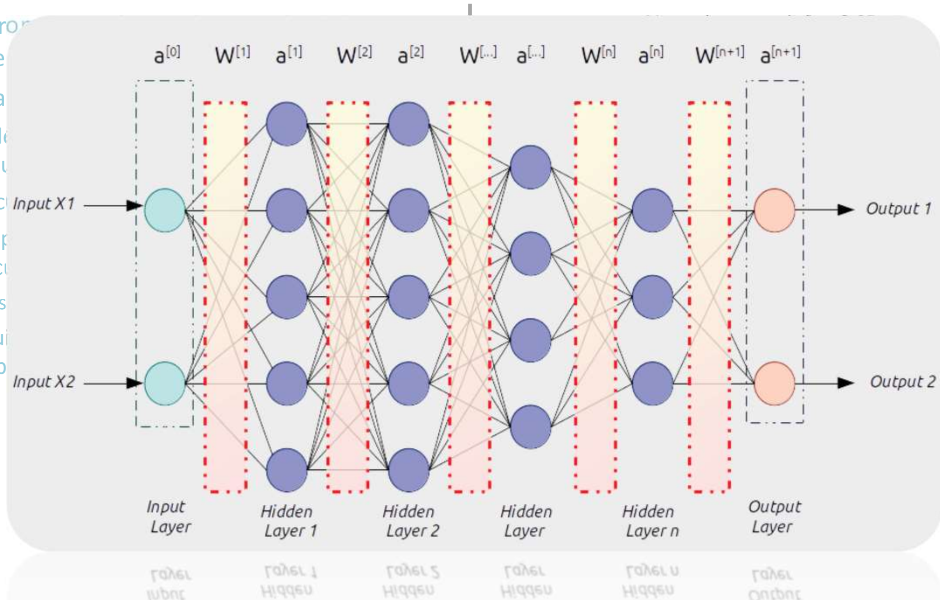
- The properties of neural networks define where they are useful
- Typical Network
  - Can learn complex mappings from inputs to outputs, based solely on samples
  - Difficult to analyse
  - Firm predictions about neural network behaviour difficult;
    - Unsuitable for safety-critical applications.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



46

Applications

- The pro
  where
- Typica
  - Can l
    outpu
  - Diffic
  - firm p
    difficu
    - Uns
  - Requi
    can b

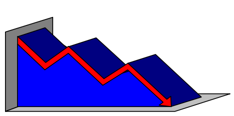


pra-sâmi

47

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

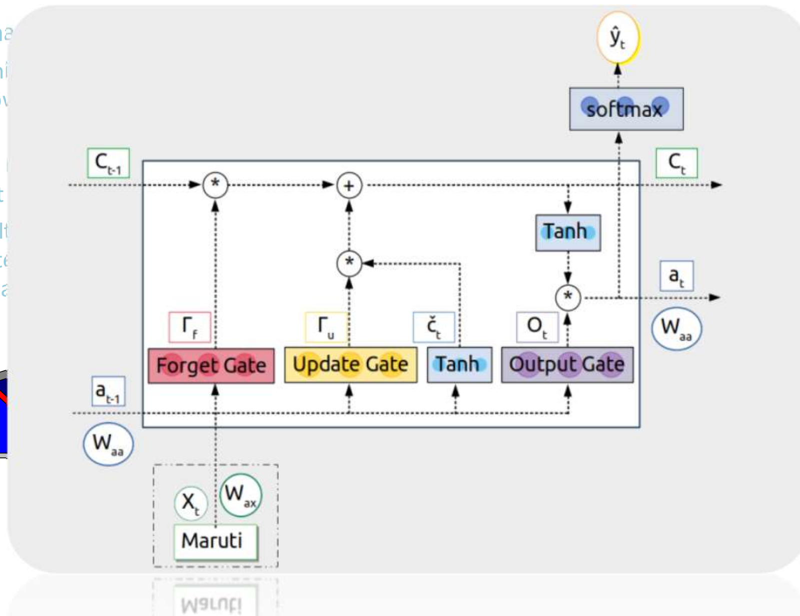
pra-sâmi

48

Applications

- Stock ma
  - "Techni
    on knov
    price
  - Neural
    predict
  - Difficult
    these te
    informa

lividual.
l cases.
ained to make
nions of expert
2% reduction in
human experts.



pra-sâmi

49

Applications

- ALVINN: Autonomous Land Vehicle In a Neural Network

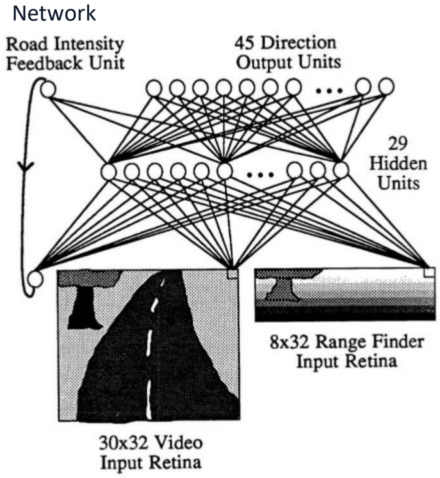
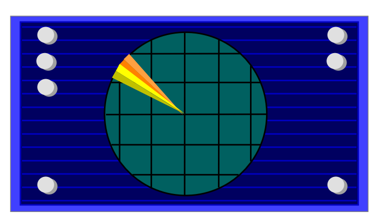


Figure 1: ALVINN Architecture

- **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.

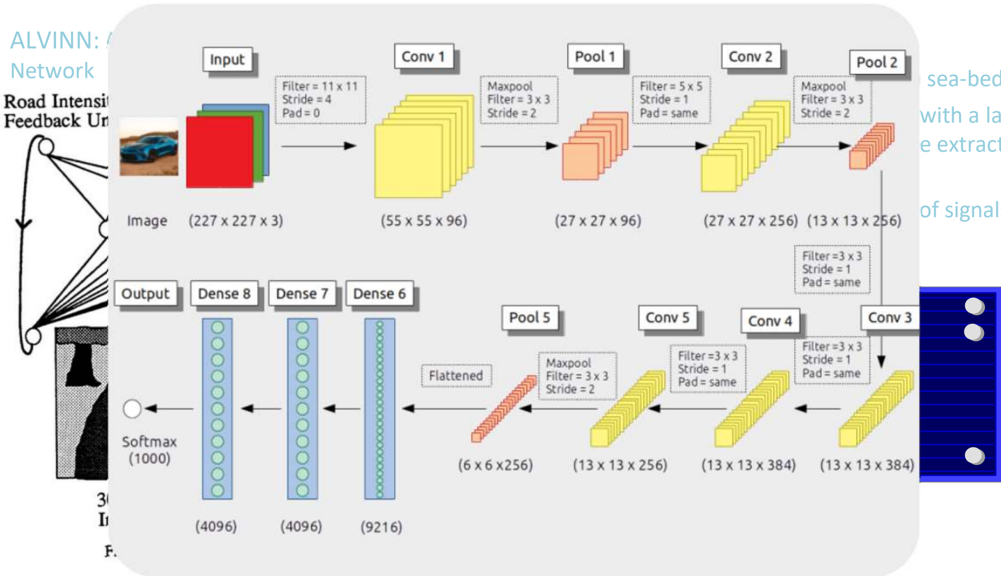


pra-sâmi

50

Applications

- ALVINN:
  Network



sea-bed
with a large
e extracted from

of signals from

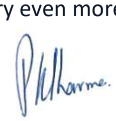
pra-sâmi

51

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.

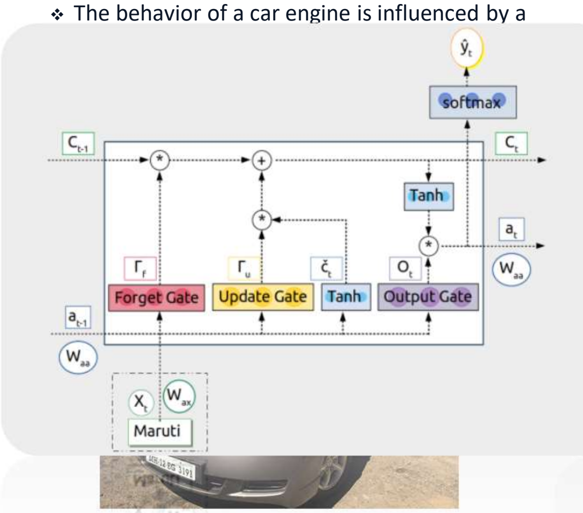


pra-sâmi

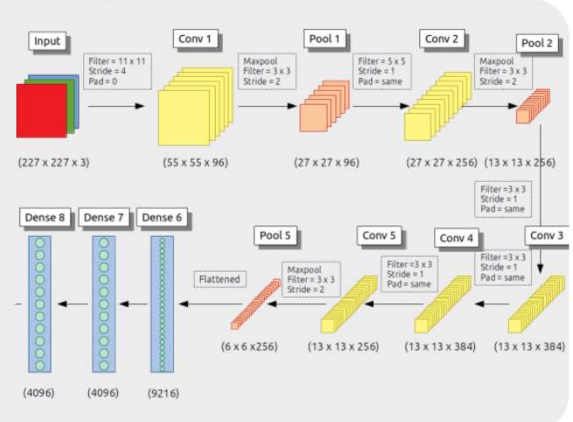
52

Applications

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



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



pra-sâmi

53

Derivation of Sigmoid

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

pra-sâmi