DEEP LEARNING

Introduction to ML, DL, AI and OpenVino

Session 06

Pramod Sharma
pramod.sharma@prasami.com

2

Agenda

- Introduction
- AI vs ML vs Deep Learning
- MP Neuron
- Perceptron
- Single Layer Neural Network
- Overview of back propagation of errors

3

References

- Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courville
- Neural Networks and Learning Machines, Simon Haykin
- Pattern Recognition and Machine Learning, Christopher M. Bishop
- Deep Learning with Python - François Chollet
- Hands-On Machine Learning with Scikit-Learn and TensorFlow
- TensorFlow Deep Learning Cookbook
- Reinforcement Learning with TensorFlow: A Beginner's Guide to Designing Self-learning Systems with TensorFlow and OpenAI Gym Sayon Dutta
- Hands-On Reinforcement Learning with Python: Master Reinforcement and Deep Reinforcement Learning Using OpenAI Gym and TensorFlow Sudharsan Ravichandiran
- Deep Reinforcement Learning Hands-On: Apply Modern RL Methods, with Deep Q-networks, Value Iteration, Policy Gradients, TRPO, AlphaGo Zero and More Maxim Lapan

pra-sâmi

4

Agent In Uncertain Environment

- Agents don’t have complete knowledge about the world.
- Agents need to make (informed) decisions given their uncertainty.
- It isn’t enough to assume what the world is like.
  - Example: wearing a seat belt.
- An agent needs to reason about its uncertainty.
- When an agent takes an action under uncertainty, it is gambling ⇒ probability

pra-sâmi

5

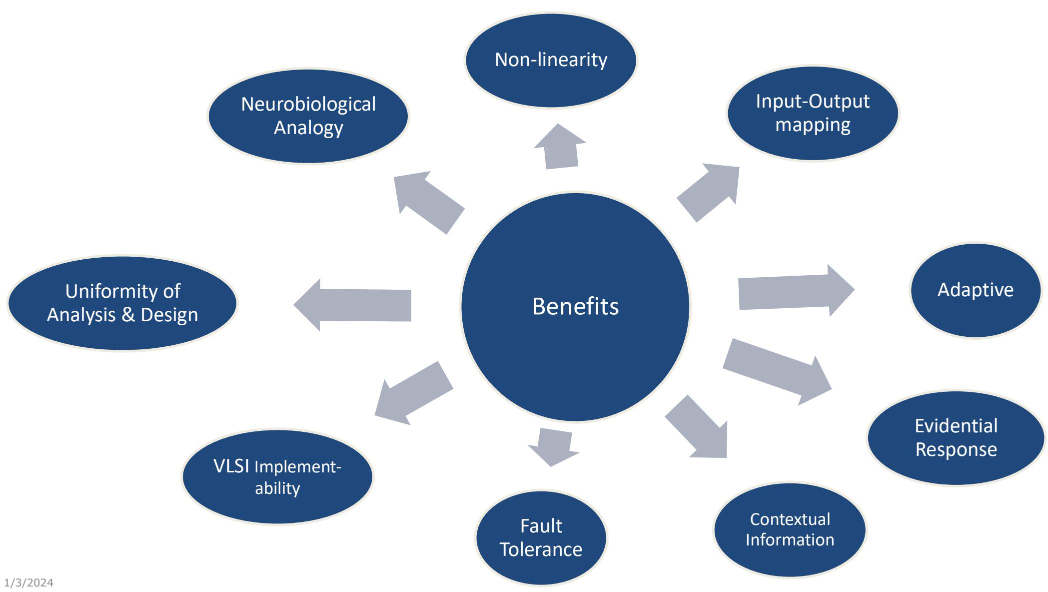
Overview

- Nature is a continuum where as math is discreet values
  - Old film based images were continuous painting of colors where as digital images are pixels
- Brain works differently than our mathematical computations
- Brain is highly complex, nonlinear and parallel computer
- Neural networks are supposed to be inspired fro
- Highly generalized form, a Neural N ... mulate manner in which brain performs a task

pra-sâmi

6

Benefits of Neural Networks



pra-sâmi

7

What has been achieved so far

- Learn to see and hear… so natural to humans but elusive to machines earlier
- Image classification
- Speech recognition
- Handwriting recog
- Writing style recognition (who was the author)
- Improved machine translation
- Text-to-speech conversion

- Digital assistants such as Google Now and Amazon Alexa
- Little autonomous driving
- Improved ad targeting, as used by Google, Baidu, and Bing
- esults on the web
- Ability to answer natural-language questions
- Superhuman games playing: chess, go…

Still long way to go…
Human-level general intelligence too far away…

1/3/2024 pra-sâmi

8

Deep learning is difficult!

1/3/2024 pra-sâmi

9

Neurons

pra-sâmi

10

Neurons

- Take an example whether to go and play Cricket or not.

- Features:
  - Is it raining?
  - Is it too hot?
  - Have I completed my homework?
  - Are sufficient players ready?
  - Is cricket equipment ready?
  - Is ground available?

- Depending on the feature values, you may get to play or not

- Features like homework and availability of ground can be considered as 'inhibitory'.

id	Dry Weather	Low Temp	Homework Done	Team Members	Equipment	Ground	Played
1	1	1	1	1	0	1	1
2	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1
4	0	1	0	1	1	1	0
5	0	0	1	1	1	0	0
6	0	0	0	0	0	1	0

- Notes :
  - Aggregator function is sum and threshold can be 3.
  - Assign 0 or 1 if a parameter is in favor or not

Given sufficient data point, we can train an algorithm to make such simple decisions for us.

pra-sâmi

11

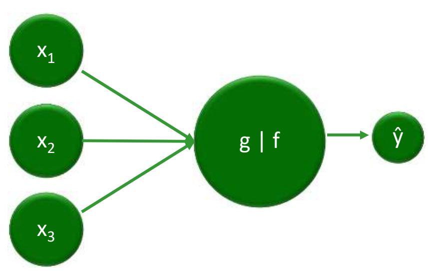
MP Neuron

- In 1943 Warren S. McCulloch, a neuroscientist, and Walter Pitts, a logician, published "A logical calculus of the ideas immanent in nervous activity" in the Bulletin of Mathematical Biophysics
- In this paper McCulloch and Pitts tried to understand how the brain could produce highly complex patterns by using many basic cells that are connected together
- These basic brain cells are called neurons, and McCulloch and Pitts gave a highly simplified model of a neuron in their paper
- The McCulloch and Pitts model of a neuron, which we will call an MCP neuron for short, has made an important contribution to the development of artificial neural networks -- which model key features of biological neurons

12

MP Neuron

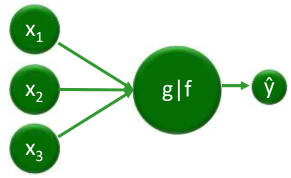
- Neurons receive signals and produce a response
- In this model:
  - All inputs are binary i.e. [0,1]
  - Inputs are “inhibitory” or “excitatory”.
  - Inhibitory have maximum influence on the model
  - It has an aggregator ‘g’ and a function ‘f’
  - There is a threshold
  - If g is more than threshold, ŷ = 1 else 0



- ŷ = 0 if any x_i is inhibitory, else g(x) = $\sum x_i$
- ŷ = 1 if g(x) ≥ threshold else ŷ = 0

13

MP Neuron

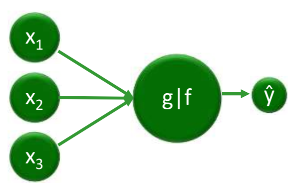


id	Dry Weather	Low Temp	Homework Done	Team Members	Equipment	Ground	Sum	Played
1	1	1	1	1	0	1	5	1
2	1	1	1	1	1	1	6	1
3	1	1	1	1	1	1	6	1
4	0	1	0	1	1	1	4	0
5	0	0	1	1	1	0	3	0
6	0	0	0	0	0	1	1	0

pra-sâmi

14

MP Neuron



id	Dry Weather	Low Temp	Homework Done	Team Members	Equipment	Ground	Sum	Played
1	1	1	1	1	0	1	5	1
2	1	1	1	1	1	1	6	1
3	1	1	1	1	1	1	6	1
4	0	1	0	1	1	1	4	0
5	0	0	1	1	1	0	3	0
6	0	0	0	0	0	1	1	0

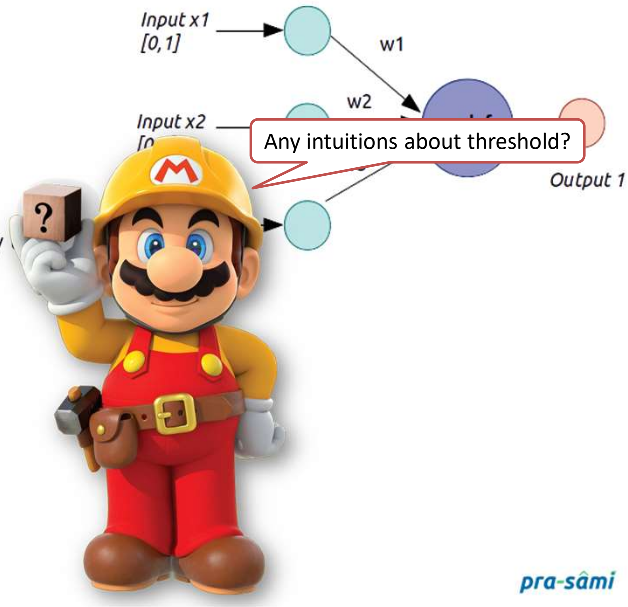
The logic is straight forward. Let's implement this model on a dataset.

pra-sâmi

15

Code Example1 – MP Neurons

- Need a dataset with plenty of features and binary output
- Use Breast Cancer dataset from scikit-learn
  - data = sklearn.datasets.load_breast_cancer()
- Its features are a continuous and we need binary
  - Use pandas pd.cut to bin the columns
  - X_bin = X . apply (pd.cut, bins=2, labels=[1,0])
  - For b in range [0, num_features+1]
    - Sum it by row and compare with b



pra-sâmi

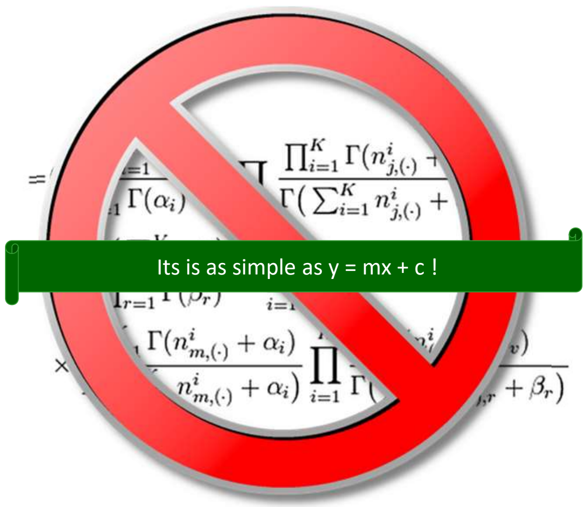
16

Perceptrons

pra-sâmi

17

Solution to Equation of Perceptron



Ian Goodfellow
Yoshua Bengio
Aaron Courville

18

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

19

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

pra-sâmi

20

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]

pra-sâmi

21

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 $\sum x_i . w_i \geq w_0$ => ŷ = 1
  - If $\sum x_i . w_i < w_0$ => ŷ = 0

22

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

23

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

24

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

25

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

1/3/2024

pra-sâmi

26

Perceptron – Weights and Los

- Our best solution would be w ... h and ... re same.
- Loss is some function of gro ... dicted
- And we want it to be cumul
  - Square of difference looks pr
  - $\ell(\hat{y}, y) = (y - \hat{y})^2$
- Our overall loss was 2.
- By adjusting weights (w1, ... 0) ... loss to minimum (zero in this case)

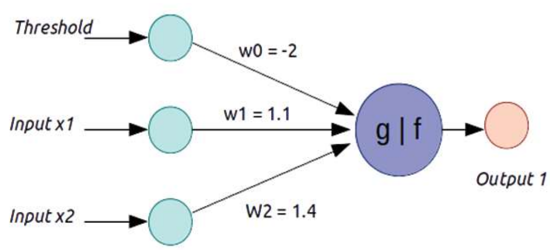
id	Threshold	[illegible]	[illegible]	Grour	[illegible]	[illegible]	[illegible]	Played	Loss
	w0	[illegible]	w1	[illegible]	[illegible]	w0+x1 ... 2*v	[illegible]	(y)	(y-y_hat)^2
1	-2.00	[illegible]	1.1	.00	[illegible]	0.5	[illegible]	1	0
2	-2.00	1.0	1	0.83	1	0.27	1	1	0
3	-2.00	1.00	[illegible]	0.67	1.4	.03	1	1	0
4	-2.00	0.44	[illegible]	1.00	1.40	11	0	0	0
5	-2.00	0.22	[illegible]	0.00	1.40	6	0	0	0
6	-2.00	0.00	[illegible]	1.00	1.40	[illegible]	0	0	0

1/3/2024

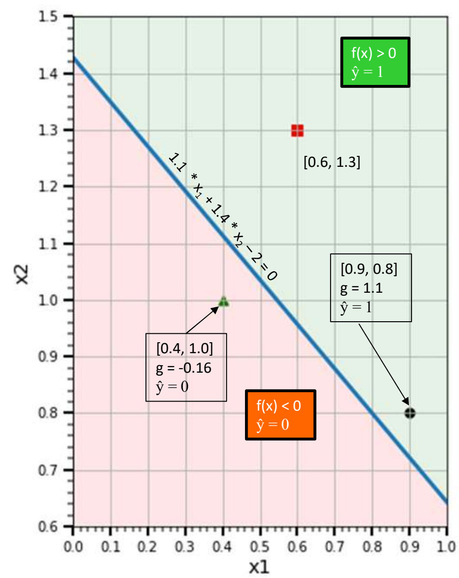
pra-sâmi

27

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

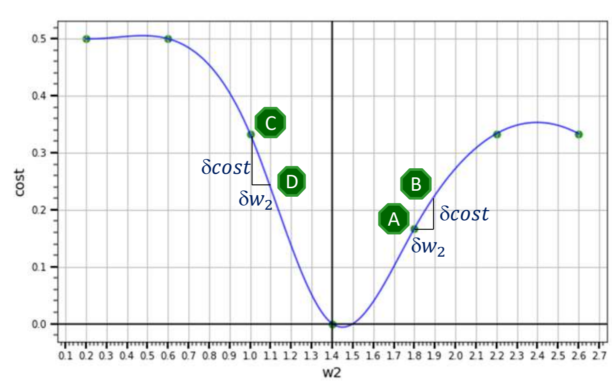


pra-sâmi

28

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.



pra-sâmi

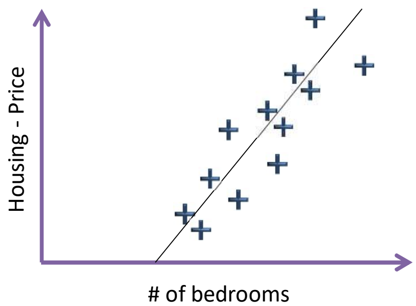
29

Perceptron – Activation Function

But that's an equation of straight line! ☺
What happened to all those 'inhibitory' features?

- So we based our entire calculations on:

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

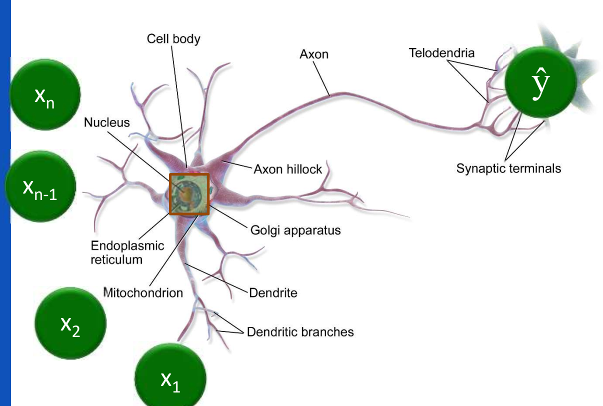


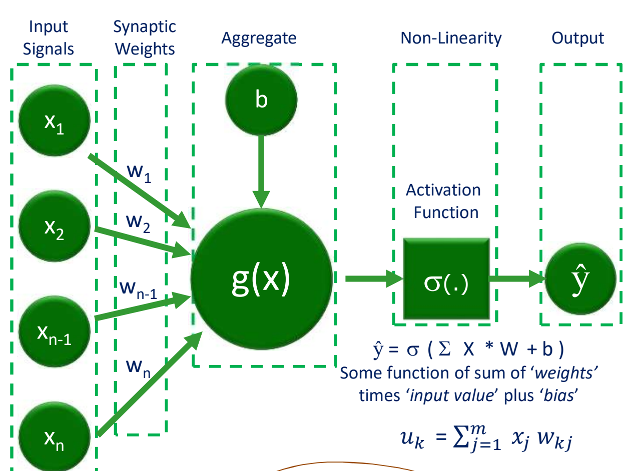
1/3/2024

pra-sâmi

30

Non Linear Activation function





ŷ = σ (Σ X * W + b)
Some function of sum of '*weights*' times '*input value*' plus '*bias*'

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

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

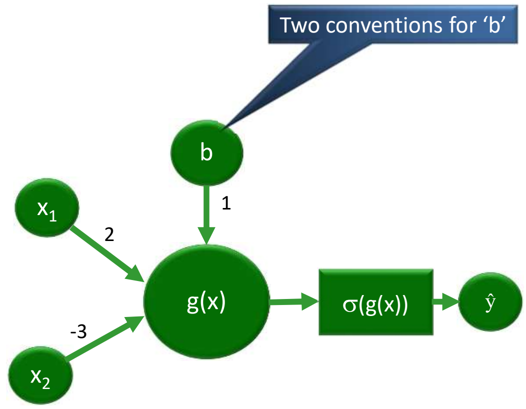
Non-linear Activation function

1/3/2024

pra-sâmi

31

Perceptron with non-linear activation function



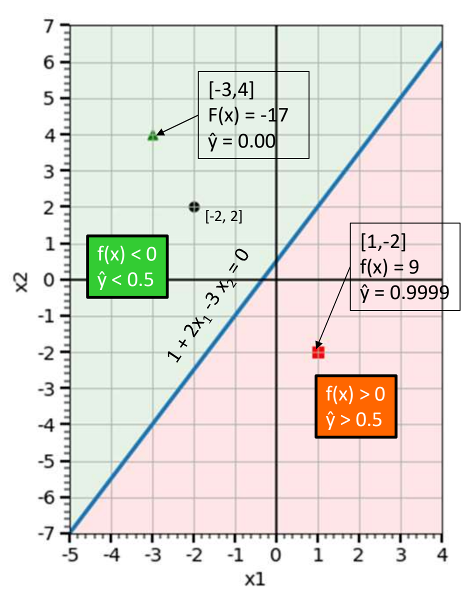
- Given:
  - W = $\begin{bmatrix} 2 \\ -3 \end{bmatrix}$ and b = 1
  - ŷ = σ ([x1, x2].$\begin{bmatrix} 2 \\ -3 \end{bmatrix}$ + 1)
  - ŷ = $\sigma\underbrace{(1 + 2 * x1 - 3 * x2)}_{z}$
- ŷ = σ (z) ;
- Lets use sigmoid function for σ.
  - ŷ = $\frac{1}{(1+e^{-z})}$

pra-sâmi

32

Perceptron with non-linear activation function

- ŷ = $\sigma\ (1 + 2 * x_1 - 3 * x_2)$
- 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

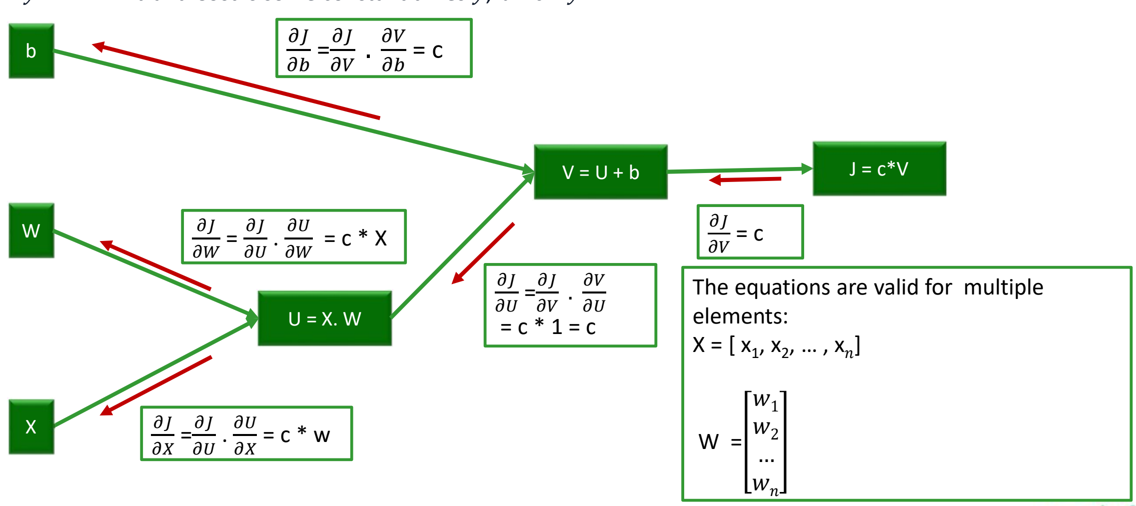


pra-sâmi

34

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 * $\hat{y}$



1/3/2024

pra-sâmi

35

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

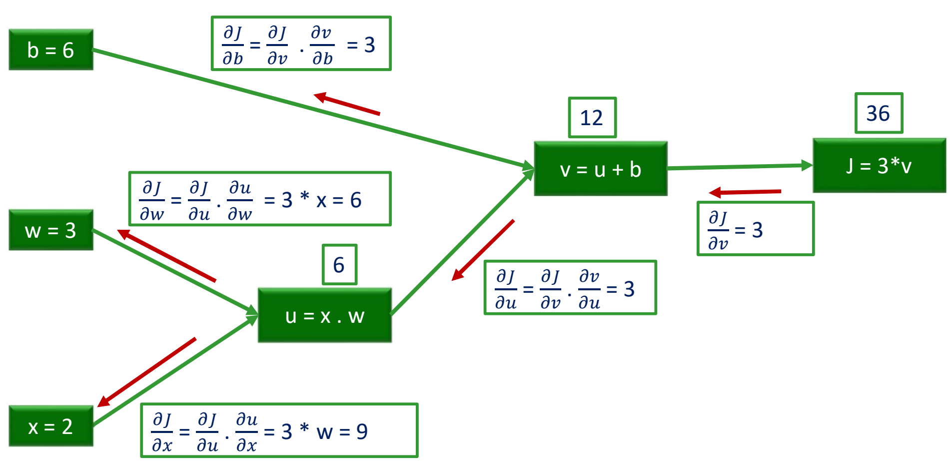
1/3/2024

pra-sâmi

36

Exercise - Solution

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



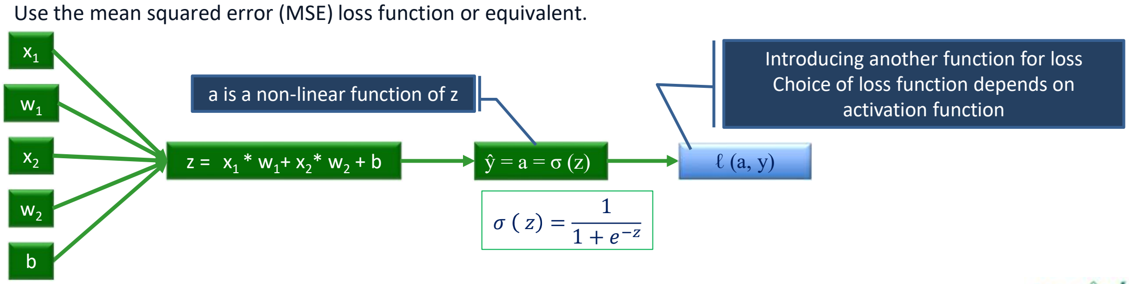
1/3/2024

pra-sâmi

37

Consider Single Path… MLE

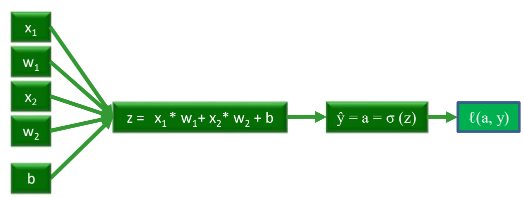
- Maximum likelihood estimation, or MLE, is a framework for inference for finding the best statistical estimates of parameters from historical training data
  - Exactly what we are trying to do with the neural network
- In Classification, output is probability of it belonging to a class
  - Maximum likelihood 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.



1/3/2024 pra-sâmi

38

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 $\ell(\hat{y}, y) = -[y * \log(\hat{y}) + (1-y) * \log(1-\hat{y})]$

1/3/2024 pra-sâmi

39

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 '$\hat{y}$' (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)= $\frac{1}{m}$ (Σ Loss($\hat{y}$, y))
  - = - $\frac{1}{m}$ (Σ (y . log $\hat{y}$ + (1 – y) . log (1 - $\hat{y}$))
- For binary classification:
  - J (W, b) = $\frac{1}{m}$ (Σ Loss($\hat{y}$, y))
  - = - $\frac{1}{m}$ (Σ (y . log $\hat{y}$)

40

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

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

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

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

z = X * W + b

$\hat{y}$ = a = σ (z)

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

ℓ(a, y) = - [y * log(a) + (1-y) * log(1-a)]

For binary classification:

ℓ (a, y) = -y * log(a)

⟹

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

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

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

⟹

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

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

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

Where α is learning rate. The cost function is

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

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

41

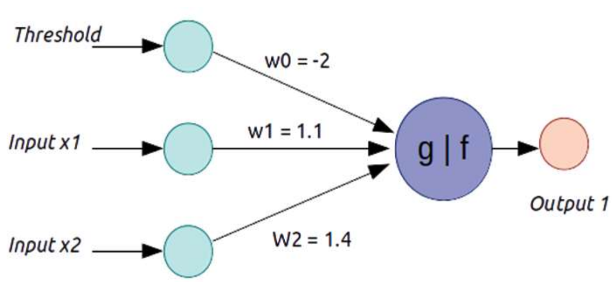
So where are the hidden layers!!!

pra-sâmi

42

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

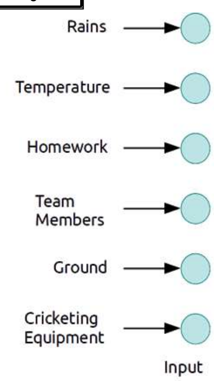


pra-sâmi

43

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

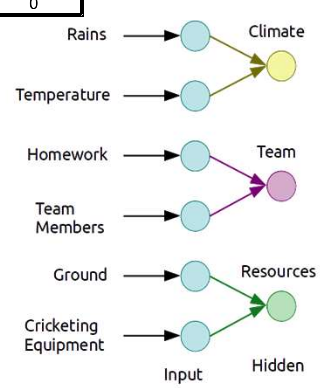


1/3/2024

44

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

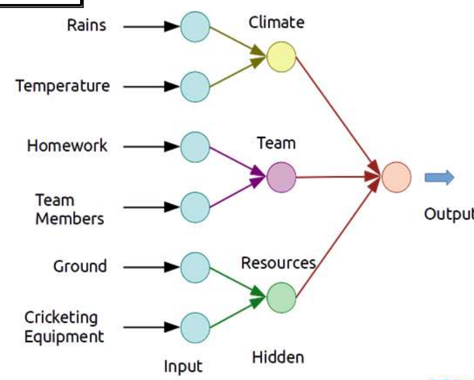


1/3/2024

45

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

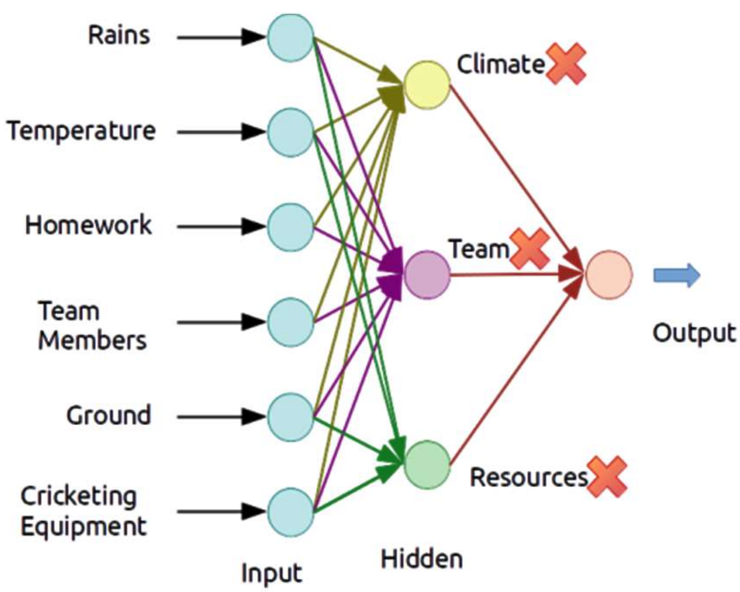


1/3/2024

pra-sâmi

46

Hidden Layers



1/3/2024

pra-sâmi

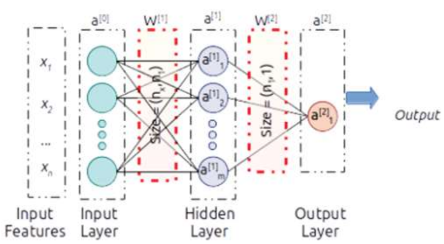
47

Hidden Layers

pra-sâmi

48

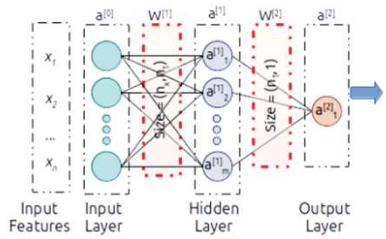
Two Major Conventions

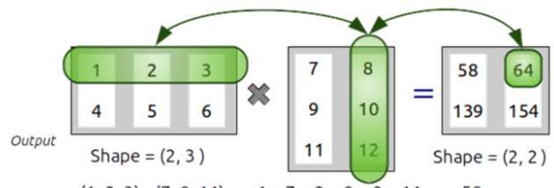


pra-sâmi

Two Major Conventions

49





(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

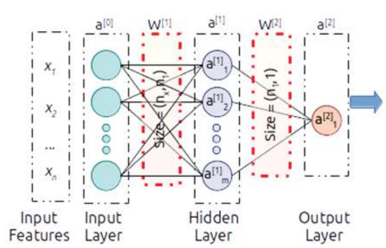
(4, 5, 6) • (7, 9, 11) = 4 × 7 + 5 × 9 + 6 × 11 = 139

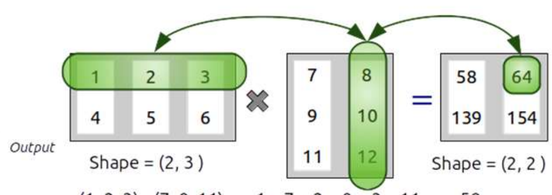
(4, 5, 6) • (8, 10, 12) = 4×8 + 5×10 + 6×12 = 154

pra-sâmi

Two Major Conventions

50



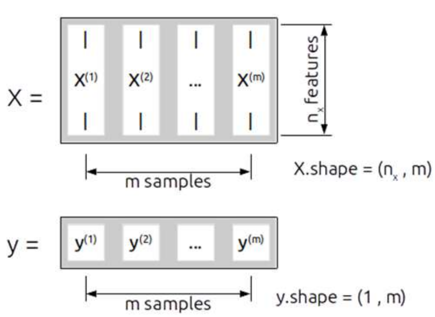


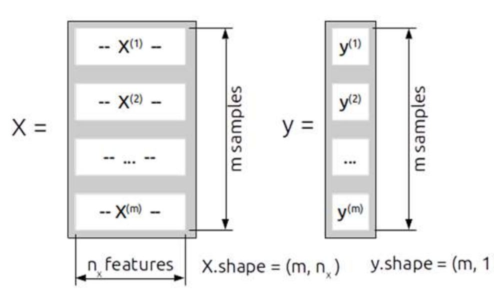
(1, 2, 3) • (7, 9, 11) = 1 × 7 + 2 × 9 + 3 × 11 = 58

(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

51

Two M

64

154

(2, 2)

X =

y =

(m, 1)

1/3/2024

pra-sâmi

52

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

1/3/2024

pra-sâmi

53

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

54

Next Session - Coding Perceptron Model in Python

55

1/3/2024

pra-sâmi

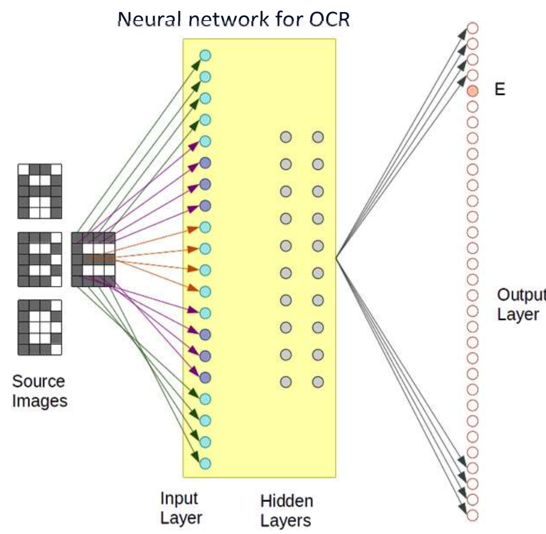
EXTRA MATERIAL

pra-sâmi

57

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



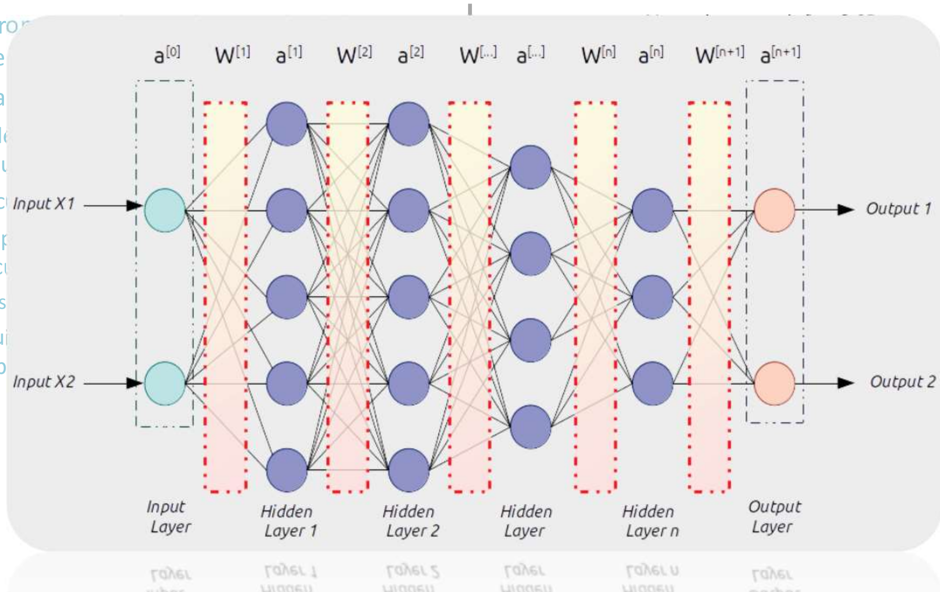
1/3/2024

pra-sâmi

58

Applications

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



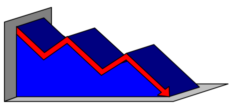
1/3/2024

pra-sâmi

59

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

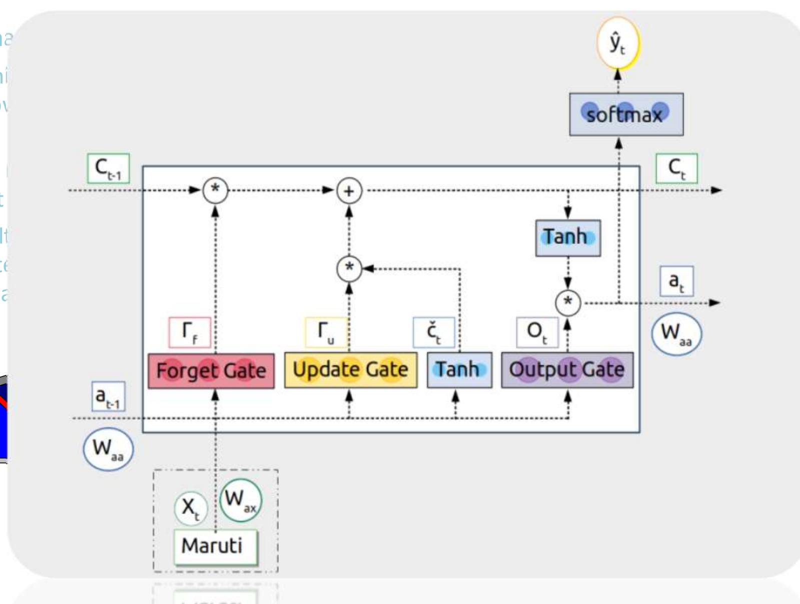
1/3/2024

pra-sâmi

60

Applications

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



ividual.
cases.
ained to make
nions of expert
% reduction in
human experts.

1/3/2024

pra-sâmi

61

Applications

- ALVINN: Autonomous Land Vehicle In a Neural Network

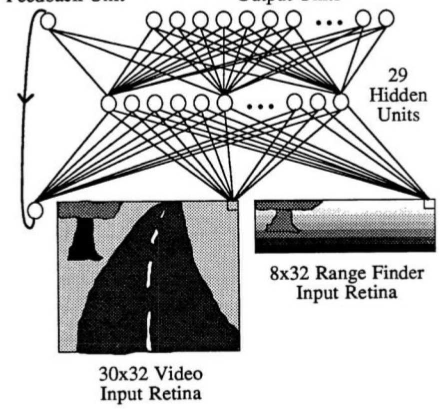
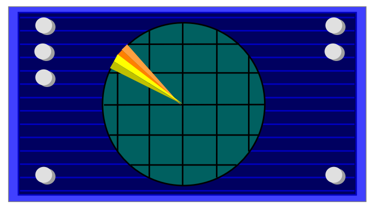


Figure 1: ALVINN Architecture

- **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.



1/3/2024

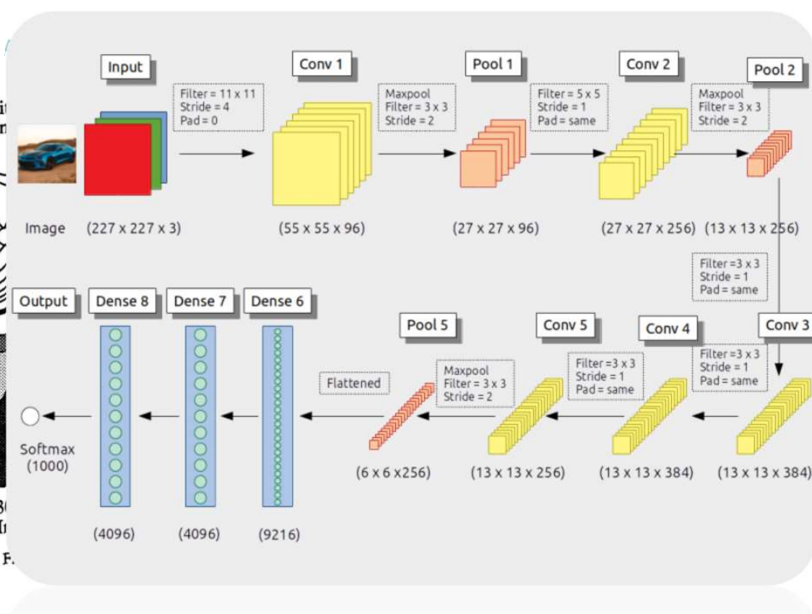
pra-sâmi

62

Applications

- ALVINN:
  Network

Road Intensi
Feedback Un



sea-bed
with a large
e extracted from
of signals from

1/3/2024

pra-sâmi

63

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.

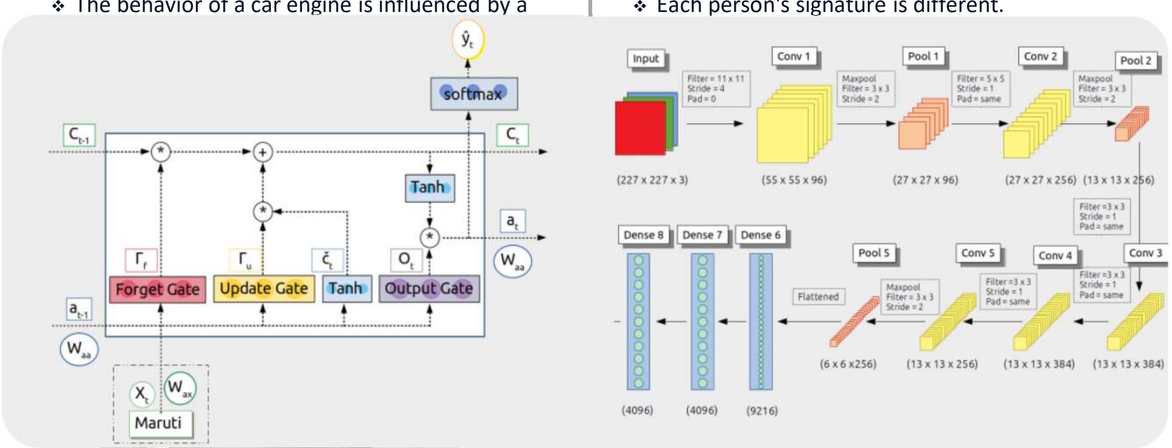
1/3/2024

pra-sâmi

64

Applications

- Engine management
  - The behavior of a car engine is influenced by a
- Signature recognition
  - Each person's signature is different.



1/3/2024

pra-sâmi

65

Derivation of Sigmoid

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

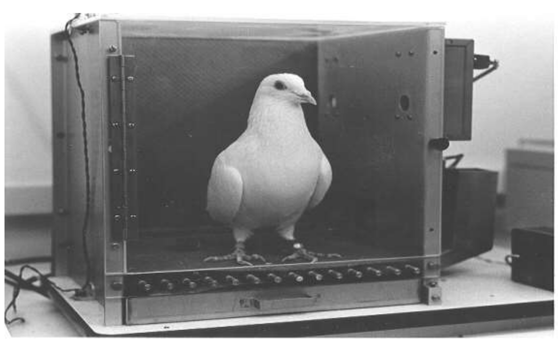
pra-sâmi

66

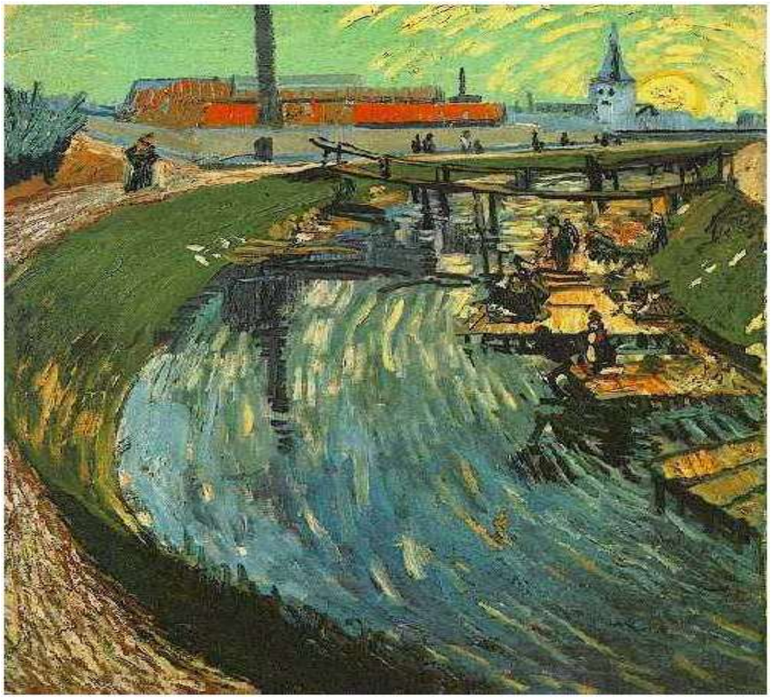
Biological Neural Nets

- Pigeons as art experts (Watanabe et al. 1995)

- Experiment:
  - Pigeon in Skinner box
  - Present paintings of two different artists (e.g. Chagall / Van Gogh)
  - Reward for pecking when presented a particular artist (e.g. Van Gogh)



pra-sâmi



69

Biological Neural Nets

- Pigeons were able to discriminate between Van Gogh and Chagall
  - With 95% accuracy on train set (when presented with pictures they had been trained on)
  - Discrimination, still 85% successful for previously unseen paintings of the artists

- Pigeons do not simply memorise the pictures
- They can extract and recognise patterns (the 'style')
- They generalise from the already seen to make predictions

- This is what neural networks (biological and artificial) are good at (unlike conventional computer)

70

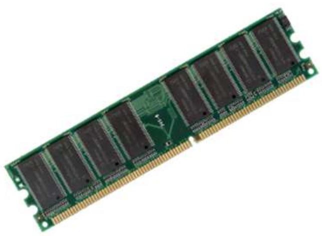
Brain and Machine

- The Brain
  - Pattern Recognition
  - Association
  - Complexity
  - Noise Tolerance

- The Machine
  - Calculation
  - Precision
  - Logic

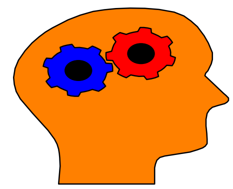
71

The contrast in architecture



- The Von Neumann architecture uses a single processing unit;
  - Tens of millions of operations per second
  - Absolute arithmetic precision

- The brain uses many slow unreliable processors acting in parallel



72

The biological inspiration

- Features of the Brain
  - Ten billion (10^{10}) neurons
  - On average, several thousand connections
  - Hundreds of operations per second
  - Die off frequently (never replaced)
  - Compensates for problems by massive parallelism

- The brain has been extensively studied by scientists

- Vast complexity prevents all but rudimentary understanding

- Even the behavior of an individual neuron is extremely complex

- Single “percepts” distributed among many neurons

- Localized parts of the brain are responsible for certain well-defined functions (e.g. vision, motion).