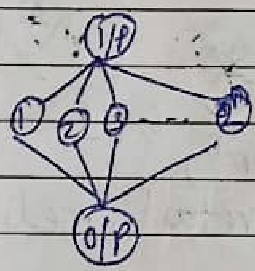
3rd August

DOMS | Page No. | Date / /

ML Lecture.

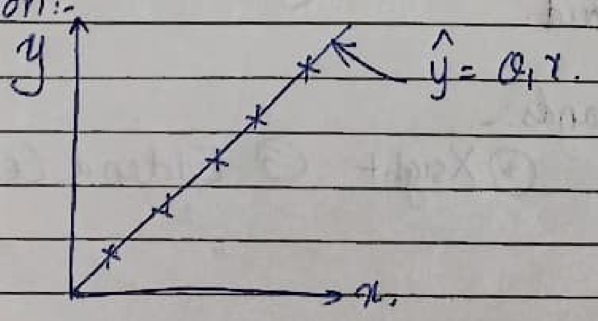
1. MP neuron.
2. MP neuron's achievements.
   AND, OR gate implementation
3. CLASSICAL PERCEPTRON.
   i) Weights & I/p ∈ R
4. Perceptron learning algorithm.
   i) weights are found iteratively.
5. Limitation of single perceptron.
   i) XOR gate can't be implemented.
6. Two boolean input.
   Max $2^{2^n} = 16$ cases to be handled.
7. 2^n



* ML classification :-

1. Supervised ML (labelled dataset) → Regression → task is to fit the func to DATA POINTS
   → Classification → task is to find func which separates classes.
2. Unsupervised ML (unlabelled data) → Clustering

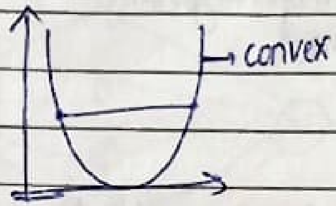
* Regression :-



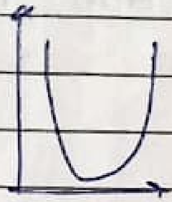
$$MSE = \frac{1}{2M} \sum_{i=1}^{m} (\hat{y} - y_i)^2$$

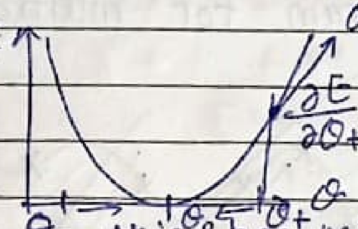
$\hat{y} = \theta^T X = \theta X$ $\hat{y}$ = predicted value.

* Convex ~~funct~~ function



any two pts when connected ~~d~~oesn't intersect any point on curve.





θ_0 = optimal theta.

this has min error.

i) find derivative at θ_+

λ = learning rate

$$w_n \leftarrow w_0 - \lambda \frac{\partial E}{\partial \theta_{+}} \qquad \lambda < 0.01$$

At slope 0, $W_n \leftarrow W_0$.

θ is vector so we take partial derivative.

$$MSE = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y} - y_i)^2$$

2 is used to compensate for the derivative.

$$\frac{\partial E}{\partial \theta} = \frac{2}{2M} (\hat{y}_i - y_i) \times x_i$$

$$W_n \leftarrow W_0 - \lambda \times \frac{1}{m} (\hat{y}_i - y_i) \times x_i$$

x_i = feature vector

θ = parameter.

$x = [x_0, x_1, x_2 .. x_m] \Rightarrow$ m+1 dimension.

- Com

* Assignments:-

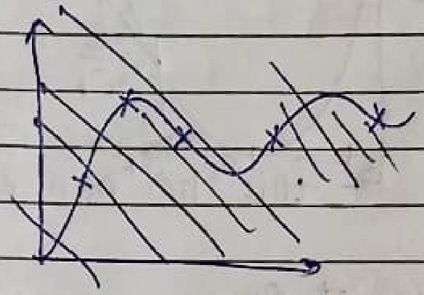
① MP N

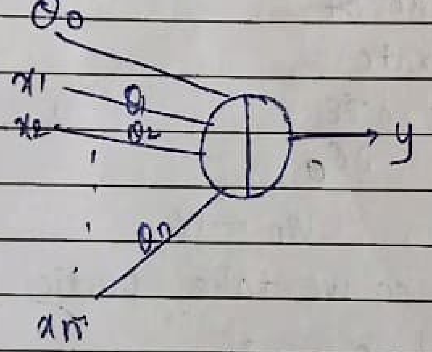
② Peceptron Learning Algorithm

③ Linear regression

④ Classification

* Draw diagram for multidimensional regression



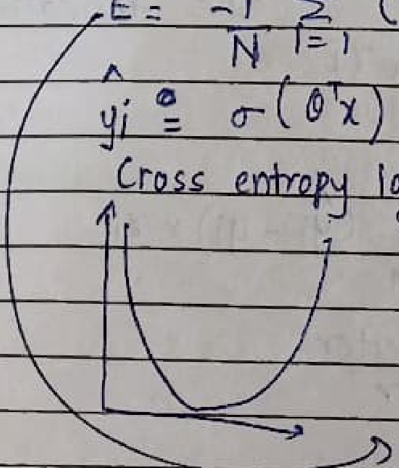


* Classification:

- $E = \frac{-1}{N} \sum_{i=1}^{N} (y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)]$

$\hat{y}_i = \sigma(\theta^T x)$ $\qquad y \in \{0,1\}$

- Cross entropy log loss



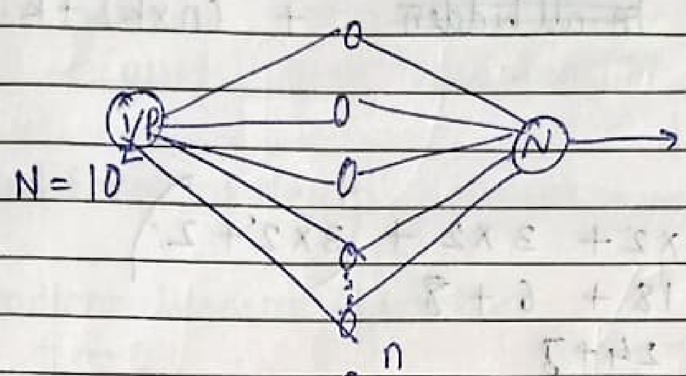
on $\frac{\partial E}{\partial x}$

we get same eqn $\rightarrow$ like regression.

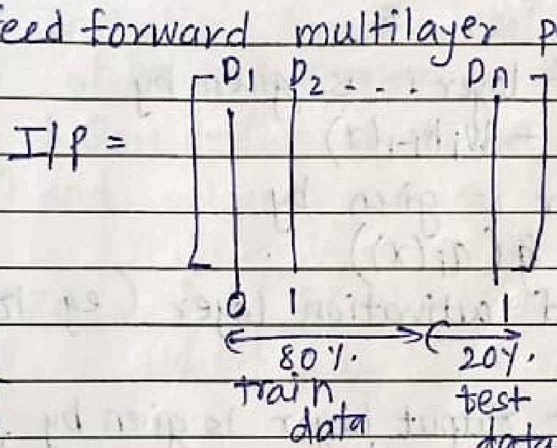
DOMS Page No.
Date / /

* Neural Networks :-

① Use MSE for regression
② Use CELL for classification.



* Feed forward multilayer perceptron :-

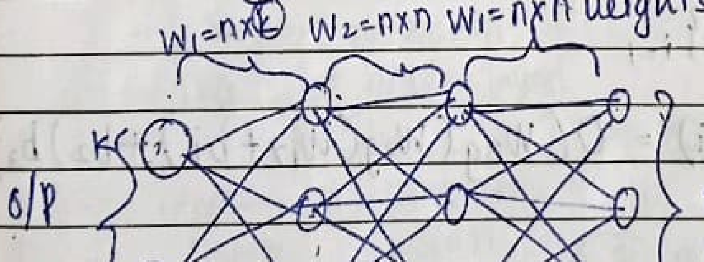


5 marks

10 Aug 2024.

* Backpropagation :-

- Types — ① Batch gradient descent
  ② Stocastic " "



$L-1 = 2$ hidden layers

One o/p layer with k ye neurons.

Here, $n\times k = 3\times 2$

$b_1 = b_2 = 3$

$b_3 = 2$

$n\times n = 3\times 3$

- why is it deep neural network ?
  It has more than 2 layers.
- No. of parameters in hidden layer = $(n\times n)(L-1)$

Here, $W_1 = W_2 = 3\times 3$, $W_3 = 3\times 2$

$6 + 9 + 9 + 2 + 3 + 3 + 3$

Ritesh Kapra

Page No.

Date / /

- i/p layer – 0th Layer   o/p Layer – 0th Layer.

- Total no of parameter $= (n \times n)(L-1) + n(L-1)$
  ~~in all hidden~~ $+ (n \times k) + k.$

$n = 3$

$L = 3$

$k = 2$

$Ans = 9 \times 2 + 3 \times 2 + (3 \times 2 + 2)$

$= 18 + 6 + 8$

$= 24 + 8$

$= 32.$

- The preactivation of layer i is given by
  $a_i(x) = b_i + W_i h_{i-1}(x)$
- The activation layer is given by
  $h_i(x) = g(a_i(x))$
  where g is called activation layer (eg. logistic, tanh, linear)
- The activation at output layer is given by,
  $-f(x) = h_L(x) = O(a_L(x))$
  whr O is the output activation function (eg. softmax, linear)

i) Data: $\{x_i, y_i\}_{i=1}^{N}$

ii) Model:
$\hat{y}_i = f(x_i) = O(W_3 g(W_2 g(W_1 x + b_1) + b_2) b_3)$

iii) Parameters :-
$\theta = W_1 \ldots W_L, b_1, b_2, \ldots b_L \ (L = 3)$

iv) Algorithm :-
- GD with back propagation

1 summation = batch size
2nd " = o/p size.

v) Objective/Loss/Error fuction :-

$$\min \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{k} (\hat{y}_{ij} - y_{ij})^2$$

In general, min $\cancel{L(\theta)}$ $L(\theta)$. where $L(\theta)$ is some spec func of paraneters.

* Feed forward Neural Network :-

- Gradient Descent Algorithm:-

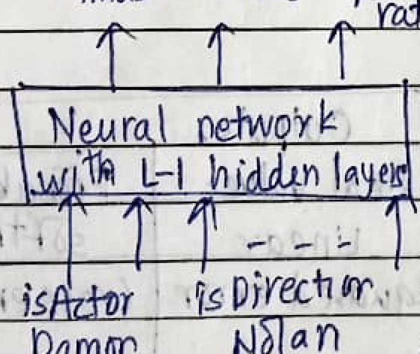
① $t \leftarrow 0$;
② max-iterations $\leftarrow 1000$
③ Initialize $\theta_0 = [w_0, b_0]$.
④ while $t{+}{+} <$ max-iterations do
  $\theta_{t+1} \leftarrow \theta_t - \eta \nabla \theta_t$
⑤ end.

$$\nabla \theta_1 = \left\{ \left[\frac{\partial L(\theta)}{\partial W_{3,t}}, \dots \frac{\partial L(\theta)}{\partial b_{L,1}} \right]^T \right.$$

4 marks

* Example — Describe prob & how will u solve it.

$y_i = \{7.5 \text{ imdb}, 8.2 \text{ critics}, 7.7 \text{ RR}\}$ ratings.



x_i

① Its a regression problem coz o/p is given & we arn't classifying.

② features — isActor, isDirector, etc.

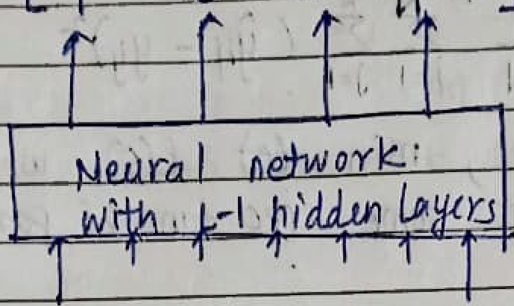
③ dimensions = 4
~~(3 o/p + 1 bias)~~

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{3} (\hat{y}_{ij} - y_{ij})^2.$$

Here $y_i \in \mathbb{R}^3$

Apple Mango Orange Banana.

eg: 2 $y = [1 \quad 0 \quad 0 \quad 0]$.



x_i Apple

Softmax func is used.

eg: $y_1 = \frac{e^1}{e^1+e^2+e^3+e^4} = \frac{1}{1+e^1+e^2+e^3}$ $y_2 = \frac{e^1}{1+e^1+e^2+e^3}$.

$y_3 = \frac{e^2}{1+e^1+e^2+e^3}$ $y_4 = \frac{e^3}{1+e^1+e^2+e^3}$.

$y_1 = \frac{1}{31.159} = 0.047$

$y_2 = 0.0871$

$y_3 = 0.2369$

$y_4 = 0.6439$

	Outputs: Real values	Probabilities
O/p activation	Linear	softmax.
Loss function	Squared error	Cross entropy

12 Aug 2024

DOMS Page No.
Date / /

* Chain rule:-

$$\frac{\partial \ell(\theta)}{\partial W_{111}} = \frac{\partial \ell(\theta)}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial a_{L11}} \times \frac{\partial a_{L11}}{\partial h_{21}} \times \frac{\partial h_{21}}{\partial a_{21}} \times \frac{\partial a_{21}}{\partial h_{11}}$$

$$\times \frac{\partial h_{11}}{\partial a_{11}} \times \frac{\partial a_{11}}{\partial W_{111}}$$

Rate of change of loss func with rt W_{111}

* Is it stochastic or normal gd?
→ No summation hence no stocastic.

* log loss → classification.

5marks

* Fordward propagation algorithm:-

1. for $k = 1$ to $L-1$ do
   $a_k = b_k + W_k h_{k-1}$
   $h_k = g(a_k)$
2. end
3. $a_L = b_L + W_L h_{L-1}$
4. $\hat{y} = O(a_L)$.

Imp 5marks

* Back propagation algorithm:-

Mitali.

23 Aug 2024

DOMS Page No.

Date / /

Class Notes

* PCA :-

- Principal Component Analysis :-
- How to compress an image. using PCA.
- Eigen value & Eigen vector.

max eigen value = 256

$$\underbrace{A}_{\text{matrix}} \underbrace{x}_{\text{Eigen vector}} = \underbrace{\lambda}_{\text{Eigen value}} x$$

- $A = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}$ is any eigen value 0 ?