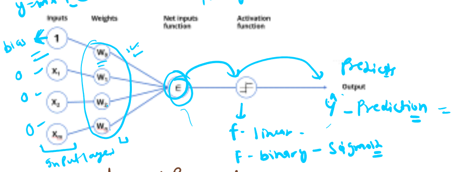
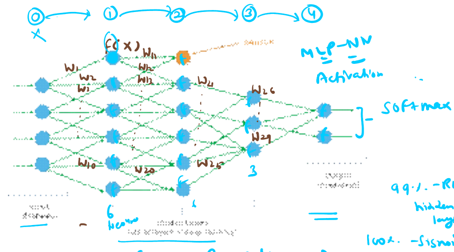
NN Architecture

Friday, September 24, 2021 10:29 AM

1-layer - Perceptron

$y = mx + c$





→ Forward Propagation → $Y \Rightarrow$ Error $= (y - \hat{y})$

Inputs, weights, Activation function, Prediction → Loss Estimation

$(x_1 .. x_m)$, $(w_1 .. w_n)$, $(a_{11}, ... a_{1m})$, $(\hat{y}_1 .. \hat{y}_m)$

Given; Randomly Initialize

$Z = W_1x_1 + ... W_mx_m$?? dot product

(i) Step
(ii) Sign
(iii) Sigmoid
(iv) Softmax
(v) RELU
(vi) Leaky RELU
(vii) tanh
:
So many

Estimation

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

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

99% - RELU hidden layer

100% - Sigmoid
- Softmax

— Forward Propagation →

$Loss = L(y - W^T.x)$ — (1) min Loss wrt W

Actual; unknown - Random; Actual

$Loss = L(y, \hat{y}) = L\left(y - (W^T.x + b)\right)$ → Loss

$= 0$ Reduce

dot Product + b

Objective = 0 Loss or Minloss

(i) MSE - Mean Squared Error - $\frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{m} \Rightarrow$ for linear output (Continuous)

(ii) BCE - Binary Cross entropy - $\sum_{i=0}^{m}[-y_i \log(\hat{y}_i) - (1-y_i)\log(1-\hat{y}_i)]$ - for Sigmoid Activation

(iii) Categorical cross entropy (CCE) $\Rightarrow \sum_{i=0}^{m} \max(0, 1 - y_i \cdot \hat{y}_i)$ - with Softmax Activation

← Back propagation ←

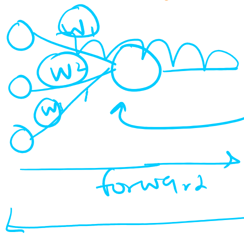
We go backwards - Why??

Min Loss

$Loss = L(y - (W^T.x + b))$ - What is unknown here??

What's the objective here??

GD ??



output $= \hat{y}$ Actual - y

Error $(y - \hat{y})$

Error $= (y - W^T.x)$

Backpropagation = ??

Neural Networks -
1) Architecture
2) Steps
3) Weights
4) Hidden layers
5) Explain via graphs
6) Refer to NN deck

Steps to be explained In NN

Questions -
1) Why weights are required
   a. Theory
   b. Technically