AI ASSIGNMENT 03

Bilal Ahmed Khan, Roll NO: 20K-0183

Sec: B

QUESTION NO. 01

① $\delta_n^K = f\left(\sum_{i=1}^{m} w_i^K a_i + w_f^K\right)$

For $\delta_1 \rightarrow$ $w_1' = w_f^K = -2$

$w_1 = [0.2, 0.4, 0.6]$

$a = [3, 5, 2]$

$\delta_1' = f\left([0.2 \times 03 + 0.4 \times 5 + 0.6 \times 2] - 2\right)$

$\delta_1' = f(1.8)$

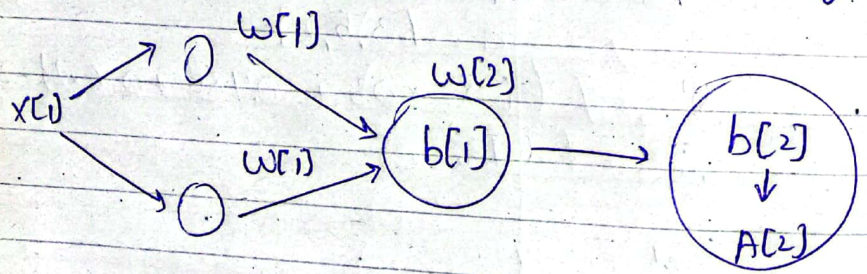
$\delta_1' = \frac{1}{(1 + e^{-1.8})}$

$\boxed{\delta_1' = 0.858}$

for $_1S^2 \rightarrow w_2^1 = w_b^k = 2.$

$$W_2 = [0.1, 0.9, 0.7]$$
$$a = [3, 5, 2]$$
$$\delta_2^1 = f([0.1\times3+0.9\times5+0.7\times2]+2)$$
$$\delta_2^1 = f(8.2)$$
$$\delta_2^1 = \frac{1}{1+e^{8.2}}$$
$$\boxed{\delta_2^1 = 0.998}$$

Hidden layer   Output Layer



Here,
- → $x[1]$ is input matrix of size $(n \times m)$
- → $W[1]$ & $W[2]$ are weights accredited with inputs
- → $b[1]$ & $b[2]$ represents bias on $Z[1]$ & $Z[2]$
- → $A[2]$ is the activation energy on layer $Z[2]$
- → $Z[1]$ hidden layers having p no. of neutrons
- → $Z[2]$ output layer having q no. of neutrons

$\partial L / \partial B[1]$

Chain Rule $\Rightarrow \frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx}$

The loss(L) depends upon $Z[2]$ which depends upon $A[2]$ & $A[2]$ depends on $Z[1]$ which itself depends on $B[1]$

$$\frac{\partial L}{\partial B[1]} = \frac{\partial L}{\partial B[2]} \frac{\partial L}{\partial Z[2]} \frac{\partial Z[2]}{\partial A[2]} \frac{\partial A[2]}{\partial Z[1]} \frac{\partial Z[1]}{\partial B[1]}$$

→ In the above equations we replace the symbols with exact function which are missing.

→ Back propagation allows updation of weights & biases based on error rate. it also reduces computational & memory cost for neural ~~networking~~ training.

—|—|

QUESTION NO.02

Initial Matrix

$$\begin{array}{c|cccccc} & A & B & C & D & E & F \\ \hline A & 0 & 0 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 0 & 0 & 0 \\ D & 0 & 0 & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 0 & 0 \\ F & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Q Matrix

Starting Pos ⇒ B
Next Step ⇒ F
Exploration rate = 0.
reward = 100

$$Q(\text{state, action}) = R(\text{state, action}) + \gamma \cdot \max[Q(\text{next state, allocation})]$$

$$Q(B,F) = R(B,F) + \gamma \cdot \max[Q(F,B) + Q(F,E) + Q(F,F)]$$

$$= 100 + 0.8 \max(0,0,0)$$

$$= 100 + 0.8(0)$$

$$\boxed{Q(B,F) = 100}$$

Updated Matrix Q

$$\begin{array}{c|cccccc} & A & B & C & D & E & F \\ \hline A & 0 & 0 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 0 & 0 & 0 & 100 \\ C & 0 & 0 & 0 & 0 & 0 & 0 \\ D & 0 & 0 & 0 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 0 & 0 \\ F & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$