6) Reinforcement Learning

* Reinforcement learning :-
  - is a feedback-based machine learning technique in which an agent learns to behave in an environment by performing actions & seeing the results of actions
  - For each good action, the agent gets +ve feedback, & for each bad action, the agent gets -ve feedback or penalty.

ex) game-playing, robotics.

* Elements of RL :-

1) Policy :-
   - Policy defines the learning agent behavior for given time period.
   - It is mapping from perceived states of env. to actions to be taken when in those states.

2) Reward function :-
   - Reward function is used to define a goal in a reinforcement learning problem.
   - A reward fn is a function that provides a numerical score based on the state of env.

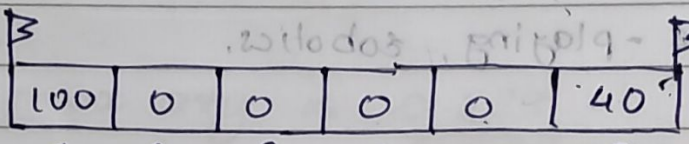
3) Value function :-
   - specify ~~when~~ what is good in long run.
   - the value of state is total amount of reward an agent can expect to accumulate over the future, starting from that state.

4) Model :-
   - with the help of model, one can make inferences about how the environment will behave.

-such as, if a state and action are given, then a model can predict the next state & reward.

* Discount Factor:- (γ) :- Ranges from 0 to 1
- cost of travelling from one state to another.

Ex)



consider: $\gamma = 0.5$

From 1:-

i) To left:-

Return = 100

ii) To right:-

Return = $\gamma^5 . 40 = 1.25$

From 2:-

1) To left:-

Return = $\gamma^1 . 100$

$= 0.5 * 100$

$= 50$

2) To right:-

Return = $\gamma^4 . 40$

$= 0.5^4 * 40$

$= 2.5$

* State-action value function:

$Q(S,a)$ = Return.

S = state a = action.

optimal behaviour:-

100	← 0	← 0	← 0	→ 0	40
1	2	3	4	5	6

$\gamma = 0.5$

1) $Q(3, \rightarrow) = \gamma^0 + \gamma^1 \cdot 0 + \gamma^2 \cdot 0 + \gamma^3 \cdot 0 + \gamma^4 \cdot 100$

$= \gamma^4 \cdot 100$

$= 0.5^4 \cdot 100$

$= 6.25$

2) $Q(5, \leftarrow) = \gamma^4 \cdot 100$

$= 0.5^4 \cdot 100$

$= 6.25$

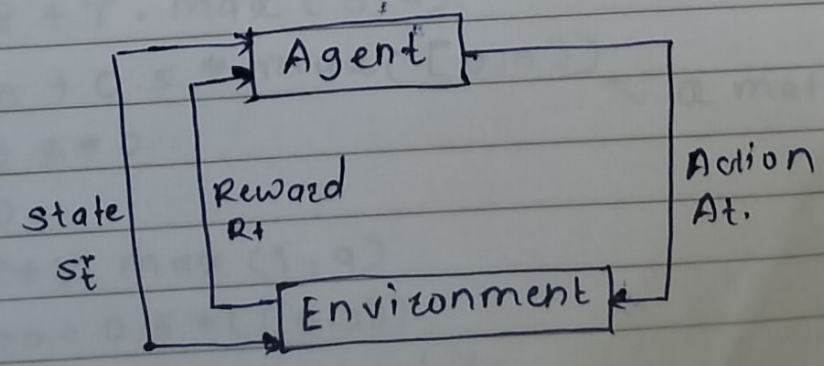
3) $Q(3, \leftarrow) = \gamma^2 \cdot 100$

$= 0.5^2 \cdot 100$

$= 25$

Now,
max $Q(s, a)$ for all a it is called Q^*

* Markov decision process :-
- a mathematical framework used for modelling decision-making processes in situations where outcomes are partly random & partly under the control of a decision-maker.

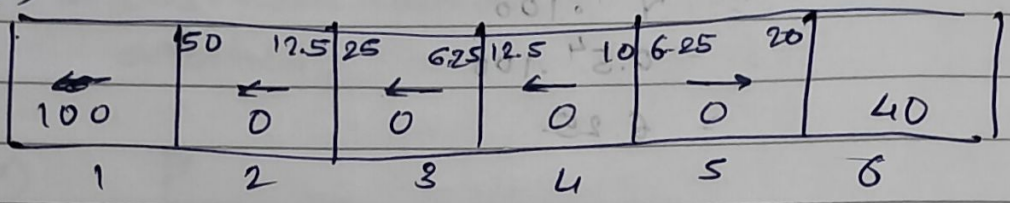


* Bellman eqn :-

- Helps to find out Q(s,a)

$$Q(s,a) = R(s) + \gamma \max Q(s', a')$$

Ex)



$\gamma = 0.5$

checking values using Bellman eqn.

① $Q(3, \rightarrow) = R(3) + \gamma \max([12.5, 10])$

$= 0 + 0.5 * 12.5$

$= 6.25$

② $Q(4, \rightarrow) = R(4) + 0.5 * \max([6.25, 20])$

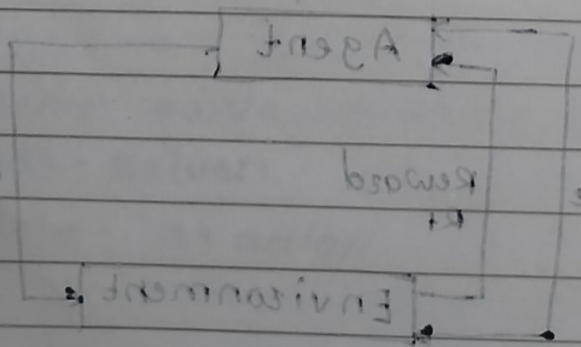
$= 0 + 0.5 * 20$

$= 10$

③ $Q(5, \leftarrow) = R(5) + 0.5 * \max([12.5, 10])$

$= 0 + 0.5 * 12.5$

$= 6.25$



Q-Learning :-

- enables model to iteratively learn & improve over time by taking the correct action.

Ex)

Reward matrix :-

	1	2	3	4	5	6
1	-1	0	-1	+1	-1	-1
2	100	-1	0	-1	-1	-1
3	-1	0	-1	0	-1	-1
4	-1	-1	0	-1	0	-1
5	-1	-1	-1	0	-1	40
6	-1	-1	-1	+1	0	-1

1) Initialize with all zeros : (Q-matrix)

	1	2	3	4	5	6
1	0	0	0	0	0	0
2	100	0	0	0	0	0
3	0	50	0	0	0	0
4	0	0	25	0	0	0
5	0	0	0	12.5	0	40
6	0	0	0	0	0	0

Now, by Bellman eqn.

$Q(2,3) = R + \gamma \cdot \max(3, a)$

$= 0 + 0.5 * \max([0,0])$ ← Q matrix

$= 0.5 * 0$

$= 0.$

$Q(2,1) = R + \gamma \cdot \max(1, a)$

$= 100 + 0.5 * ([0,0])$

$= 100 + 0 = 100$ update

$Q(3,2) = R + \gamma \cdot \max(2,a)$

$= 0 + 0.5 * \max([100, 0])$

$= 0.5 * 100$

$= 50$ – update.

$Q(4,3) = R + \gamma \cdot \max(3,a)$

$= 0 + 0.5 * \max([50, 0])$

$= 0.5 * 50$

$= 25$ – update.

$Q(5,4) = R + \gamma \cdot \max(4,a)$

$= 0 + 0.5 * \max([25, 0])$

$= 0.5 * 25$

$= 12.5$ – update

$Q(5,6) = R + \gamma \cdot \max(6,a)$

$= 40 + 0.5 * \max([0, 0])$

$= 40 * 0$

$= 40$ – update

MOD 4: Supervised learning Advanced.

class imbalance problem:-

- this problem occurs when one class in a classification problem has significantly fewer samples than the others.
- This can lead to the model ~~being~~ being biased towards the majority class, resulting in poor performance, especially for the minority class.

Example: suppose we have a dataset with 1000 transactions, out of which only 20 are fraudent & 980 are non-fraudent. This represents class imbalance prob.

* ~~Other~~ More on evalution matrix:-

1) sensitivity:-
   - known as Recall, True +ve rate
   - measure the proportion of actual +ve cases that were correctly identified by the model.

$$\text{sensitivity} = \frac{TP}{TP+FN}$$

2) Specificity:- (True -ve rate)
   - measures the proportion of actual -ve cases that were correctly identified by the model.

$$\text{Specificity} = \frac{TN}{TN+FP}$$

3) Area Under the ROC curve (AUC-ROC) :-

- the ROC curve is graphical representation of the True +ve rate vs False +ve rate for different classification thresholds.
- AUC-ROC quantifies the models ability to distinguish betn classes.
- An AUC-ROC score of 1 indicates perfect model, while score 0.5 indicates random model.

* Logistic regression with single neuron :-

- supervised learning algorithm used for classification.
- uses sigmoid fn.

* Sigmoid fn :-

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

* Logistic regression model :-

$$z = w^T x + b.$$

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

where,

x = input feature vector.

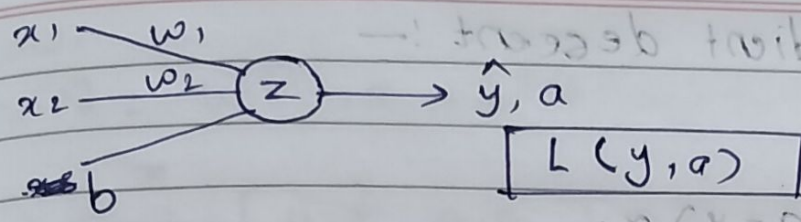
w = weight vector

b = bias.

z = combination of input features & weights.

σ = sigmoid fn

$a, \hat{y}$ = predicted output.



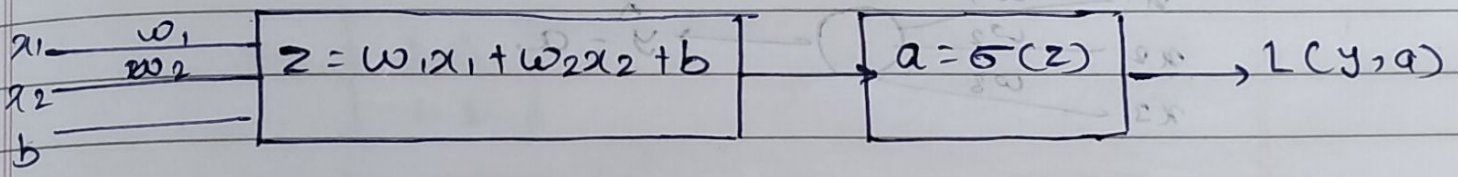
so,

$z = w_1 x_1 + w_2 x_2 + b.$

$a = \sigma(z)$

$L(y, a) = -y \log_2 a - (1-y) \log_2 (1-a)$

It can shown like this :—



For back propagation,

$$\frac{\partial z}{\partial w_1} \cdot \frac{\partial z}{\partial w_2} \cdot \frac{\partial z}{\partial b} \leftarrow \frac{\partial a}{\partial z} \leftarrow \frac{\partial L}{\partial a}$$

updating weights :-

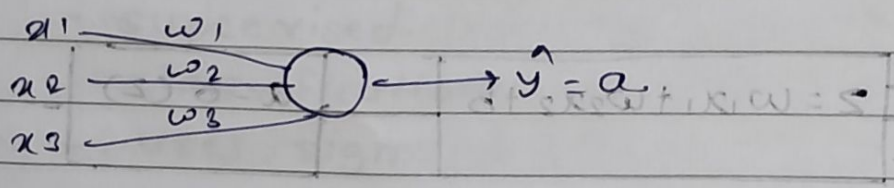
$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w_1}$$

$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial z}{\partial w_2}$$

$$\frac{\partial L}{w_1} = dw_1 \qquad \frac{\partial L}{\partial w_2} = dw_2 \qquad \frac{\partial L}{\partial b} = db.$$

Now, Gradient descent :—

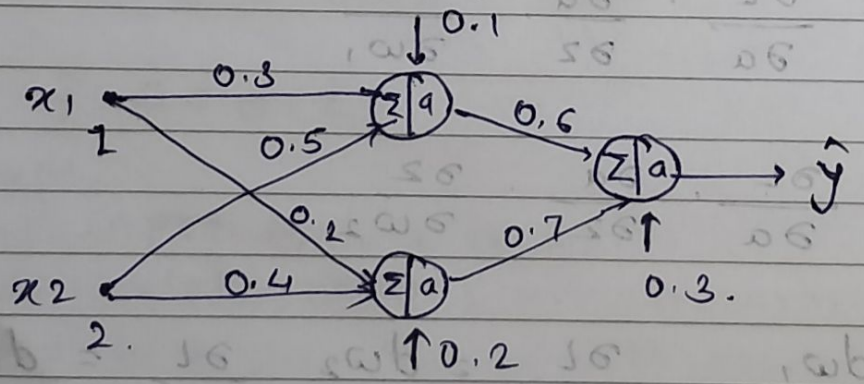
Repeat
{
$dw_1 = (a-y)\, x_1$
$dw_2 = (a-y)\, x_2$
$db = (a-y)$
$w_1 = w_1 - \alpha\, dw_1$
$w_2 = w_2 - \alpha\, dw_2$
$b = b - \alpha\, db$.
}.



Weight matrix = $[w_1, w_2, w_3]$

Input matrix = $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

Ex)



$w^{[1]} = \begin{bmatrix} 0.3 & 0.5 \\ 0.2 & 0.4 \end{bmatrix}$ $w^{[2]} = [0.6 \quad 0.7]$

$x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ $b^{[1]} = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$ $b^{[2]} = [0.3]$

Now,

$$z^{[1]} = w^{[1]} \cdot x + b^{[1]}$$

$$= \begin{bmatrix} 0.3 & 0.5 \\ 0.2 & 0.4 \end{bmatrix} * \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$$

$$= \begin{bmatrix} 0.3+1 \\ 0.2+0.8 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$$

$$= \begin{bmatrix} 1.3 \\ 1.0 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$$

$$= \begin{bmatrix} 1.4 \\ 1.2 \end{bmatrix}$$

Now apply sigmoid,

$$a = \sigma(z)$$

$$a^{[1]} = \begin{bmatrix} 1/(1+e^{-1.4}) \\ 1/(1+e^{-1.2}) \end{bmatrix} = \begin{bmatrix} 0.8021 \\ 0.7685 \end{bmatrix}$$

Now,

$$z^{[2]} = w^{[2]} \cdot a^{[1]} + b^{[2]}$$

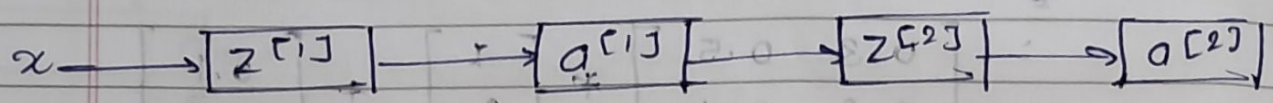
$$= [0.6 \quad 0.7] * \begin{bmatrix} 0.8021 \\ 0.7685 \end{bmatrix} + [0.3]$$

$$= [0.6*0.8021 + 0.7*0.7685] + [0.3]$$

$$= 1.31$$

$$a^{[2]} = \frac{1}{1+e^{-1.31}} = 0.78 \quad \therefore \hat{y} = 0.78.$$

Now,
For backpropagation,



$z^{[2]} = w^{[2]T} \cdot a^{[1]} + b^{[2]}$

$a^{[2]} = \sigma(z^{[2]})$

$L = -y \log_2 a^{[2]} - (1-y) \log_2 (1 - a^{[2]})$

$dz^{[2]} = a^{[2]} - y$

$dw^{[2]} = dz^{[2]} \cdot a^{[1]T}$

$db^{[2]} = dz^{[2]}$

Now,

$z^{[1]} = w^{[1]T} \cdot x + b^{[1]}$

$a^{[1]} = \sigma(z^{[1]})$

$L = -y \log_2 a^{[1]} - (1-y) \log_2 (1 - a^{[1]})$

$dz^{[1]} = w^{[2]} \cdot dz^{[2]} * g^{[1]'}(z^{[1]})$

$dw^{[1]} = dz^{[1]} \cdot x^T$

$db^{[1]} = dz^{[1]}$

$$w^{[i]}_{new} = w^{[i]}_{old} - \alpha \, dw^{[i]}$$
$$b^{[i]}_{new} = b^{[i]}_{old} - \alpha \, db^{[i]}$$

* For above example,
IF $y_{actual} = 1$, $\alpha = 0.1$. Find $w^{[1]}_{new}$, $w^{[2]}_{new}$, $b^{[1]}_{new}$, $b^{[2]}_{new}$.

→ Finding $w^{[2]}_{new}$:—

$dz^{[2]} = a^{[2]} - y$

$= 0.78 - 1$

$= -0.22$

$dw_1^{[2]} = dz^{[2]} \cdot a_1^{[1]}$
$= -0.22 * 0.8021$
$= -0.1764$

$dw_2^{[2]} = dz^{[2]} \cdot a_2^{[1]}$
$= -0.22 * 0.7685$
$= -0.1690$

$db^{[2]} = dz^{[2]}$
$= -0.1764 - 0.22$

Now,

$w_1^{[2]}new = w_1^{[2]}old - \alpha\, dw_1^{[21}$
$= 0.6 - 0.1(-0.1764)$
$= 0.6176.$

$w_2^{[2]}new = w_2^{[2]}old - \alpha\, dw_2^{[21}$
$= 0.7 - 0.1(-0.1690)$
$= 0.7169$

$b_{new}^{[2]} = b_{old}^{[2]} - \alpha\, db^{[2]}$
$= 0.3 - 0.1(-0.22)$
$= 0.322$

$\therefore w^{[2]}new = [\,0.6176 \quad 0.7169\,]$

$b^{[2]}new = [\,0.822\,]$

Finding $w^{[1]}new$ & $b^{[1]}new$.

$dz^{[1]} = w^{[2]} \cdot dz^{[2]} * g^{[1]'}(z^{[1]})$
$= [0.6 \quad 0.7) * (-0.22) * \begin{bmatrix} 0.8021 \\ 0.7685 \end{bmatrix}$
$= [-0.132 \quad -0.154] * \begin{bmatrix} 0.8021 \\ 0.7685 \end{bmatrix}$
$= -0.2242.$