Om Shree
2006077
IT - 02

ML Assignment - 01

Ans 1)

<a>

Gradient descent in an optimization algorithm used to minimise the loss function of an ML model. The gradient is a vector of partial derivatives that represents the rate of change of loss function w.r.t. weights. By updating the weights in the direction of negative gradient, the algorithm moves towards the minimum of the loss function.

<b>

Batch gradient descent		Stochastic gradient descent
deterministic in nature	(1)	stochastic in nature
Convergence is slow	(2)	convergence is fast
slow & computationaly expensive	(3)	fast & less expensive
Uses the whole training sample	(4)	Uses a single training sample

<c>

Gradient descent fails to converge when :-

① Chosing a very large learning rate which can cause gradient descent to overshoot the minimum & diverge.

② Chosing a very small learning rate, it will take a long time to reach the minimum.

③ Stochastic gradient descent may lead to slow convergence.

④ Encountering a non-convex function, which can have multiple local minima that trap the gradient descent.

⑤ A high dimensional parameter space which increases the complexity of finding the global minima.

<d>

Learning rate determines how much we adjust weights w.r.t. loss gradient descent. It affects how fast or slow we move towards the optimal weights.

If it's too small, we may never converge & if it's too big we may overshoot & increase our training error.

Ans2)

$n = 5$

x	y	xy	x^2	y^2
0	72	0	0	NA
5	66	330	25	NA
10	70	700	100	NA
15	64	960	225	NA
20	60	1200	400	NA
$\Sigma x = 50$	$\Sigma y = 332$		$\Sigma x^2 = 750$	

$\Sigma xy = 3190$

Equations for linear regression line :-

① $na + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$

② $a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i$

substituting values :-

$[5(a) + b(50) = 332] \times 10$

$50(a) + b(750) = 3190$

$250(b) = 3190 - 3320 = -130$

$b = -\frac{130}{250} = -\frac{13}{25}$

$\therefore a = \frac{332 - 50\left(-\frac{13}{25}\right)}{5} = \frac{332 + 26}{5}$

$\Rightarrow a = 71.6$

$\therefore$ Line := $y = 71.6(x) + \left(-\frac{13}{25}\right)$

Ans 3)

$\boxed{k=2}$

	x	y
A :	5	7
B :	11	45
C :	10	6
D :	18	29
E :	10	25
F :	4	3

Assume B and F as initial clusters :-

Point	x	y	D_B	D_F
A	5	7	$\sqrt{1444}$	$\sqrt{17}$
B	11	45	0	$\sqrt{1813}$
C	10	6	$\sqrt{1522}$	$\sqrt{45}$
D	18	29	$\sqrt{305}$	$\sqrt{872}$
E	10	25	$\sqrt{401}$	$\sqrt{520}$
F	4	3	$\sqrt{1813}$	0

Now ;

$A \rightarrow F$

$B \rightarrow B$

$C \rightarrow F$

$D \rightarrow B$

$E \rightarrow B$

$F \rightarrow F$

$\therefore$ New clusters = ACF BDE

ACF: $\frac{19}{3}, \frac{16}{3}$

BDE: 13, 33

For 2^{nd} iteration :-

Point	x	y	D_{ACF}	D_{BDE}
A	5	7	2.134	27.202
B	17	45	39.94	12.16
C	10	6	3.73	27.166
D	18	29	26.39	6.403
E	10	25	20.009	8.344
F	4	3	3.295	31.320

After 2^{nd} iteration :-

A → ACF
B → BDE
C → ACF
D → BDE
E → BDE
F → ACF

∴ Two clusters after 2 iterations are :-

$\boxed{ACF}$ and $\boxed{BDE}$

Ans 4) Target = (3, 7)

	x	y	Label	D_{target}
A	7	7	1	4
B	7	4	1	5
C	3	4	2	3
D	1	4	2	$\sqrt{13} = 3.8$

Label of 3 nearest neigbors is:-

2, 2, 1

$$\text{Target Label} = \frac{2+2+1}{3} = 2$$

Ans 5) **Regularization** : technique used to reduce the errors by fitting the function appropriatly on the given training set & avoid overfitting.

L1 regularization

L2 regularization

Dropout regularization

L1 regularization (LASSO) : L1 adds absolute value of magnitude of coefficient as penalty term to the loss function.

L2 regularization (Ridge) : L2 adds squared magnitude of co-efficient as penalty term to the loss function.

Ans continued :-

- $RSS = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{n} \beta_j x_{ij} \right)^2$

Ridge regression adds the shrinkage quantity to modify the RSS :-

$$\sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{n} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{n} \beta_j^2 = RSS + \lambda \sum_{j=1}^{n} \beta_j^2$$

at $\lambda = 0$, the penalty term has no effect but as $\lambda \to \infty$ the penalty influence increases.

$\Rightarrow$ $\lambda \to 0$ == $\hat{\beta}_{ridge} = \hat{\beta}_{OLS}$

$\Rightarrow$ $\lambda \to \infty$ == $\hat{\beta}_{ridge} \to 0$

In case of Lasso :-

- Penalty term is changed from β^2 to $|\beta|$:-

$$\sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{n} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{n} |\beta_j| = RSS + \lambda \sum_{j=1}^{n} |\beta$$

↳ for ridge : $RSS + \lambda \sum_{j=1}^{n} \beta_j^2$

vs

↳ for Lasso : $RSS + \lambda \sum_{j=1}^{n} |\beta_j|$

Ans6) Derivation of linear regression's loss function :-

Linear function b/w dependent & independent variables :-

$$\hat{y}^{(i)} = W_0 + W_1 x_1^{(i)} + W_2 x_2^{(i)} + \ldots\ldots W_n x_n^{(i)}$$

Some notations :-

N : no. of datapoints

n : no. of features

$x_j^{(i)}$: j^{th} feature in i^{th} observation / datapoint

Some terms :-

cost f^n : calculates the error of the entire dataset

Loss f^n : calculates error on every observation of the dataset.

$$\text{Error}^{(i)} = e^{(i)} = y^{(i)} - \hat{y}^{(i)}$$

$$e^{(i)} = y^{(i)} - \sum_{j=0}^{n} W_j x_j^{(i)} \; ; \; x_0^{(i)} = 1$$

A.c.t. least square method :-

$$\left(e^{(i)}\right)^2 = \left(y^{(i)} - \sum_{j=0}^{n} W_j x_j^{(i)}\right)^2 \; ; \; x_0^{(i)} = 1$$

$$RSS = E = \sum_{i=1}^{N} \left(e^{(i)}\right)^2 = \sum_{i=1}^{N} \left(y^{(i)} - \sum_{j=0}^{n} W_j x_j^{(i)}\right)^2 \; ; \; x_0^{(i)} = 1$$

$$X = \begin{bmatrix} x_0 & x_0 & \cdots & x_0 \\ x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(N)} \\ x_2^{(1)} & x_2^{(2)} & & x_2^{(N)} \\ \vdots & \vdots & & \vdots \\ x_n^{(1)} & x_n^{(2)} & & x_n^{(N)} \end{bmatrix} \Rightarrow X = [\bar{x}^{(1)}\ \bar{x}^{(2)} \ldots \bar{x}^{(N)}]$$

$$\forall\ \bar{x} = [1\ x_1\ x_2 \ldots x_n]^T$$

$$Y = [y^{(1)}\ y^{(2)} \ldots y^{(N)}]^T$$

$$\bar{w} = [w_0\ w_1\ w_2 \ldots w_n]^T$$

for optimum sol^n :-

$$y^{(1)} = \bar{w}^T \bar{x}^{(1)}$$

$$\vdots$$

$$y^{(N)} = \bar{w}^T \bar{x}^{(N)}$$

$$[y^{(1)}\ y^{(2)} \ldots y^{(N)}] = \bar{w}^T [\bar{x}^{(1)}\ \bar{x}^{(2)} \ldots \bar{x}^{(N)}] = \bar{w}^T X$$

or $\quad Y = (\bar{w}^T X)^T$

2) $E(\bar{w}) = \bar{w}^T [X X^T] \bar{w} - 2\bar{w}^T X y + y^T y$

↳ Find $\bar{w}^*$ which reduces the error to 0 :-

$$\frac{\partial E(\bar{w})}{\partial \bar{w}} = 2(XX^T)\bar{w} - 2Xy = 0$$

$$\Rightarrow \bar{w}^* = (XX^T)^{-1} X y$$

fitted output :-

$$\hat{y} = X^T \bar{w}^* = \underbrace{X^T (XX^T)^{-1} X y}_{X^+ y}$$

$$\Rightarrow \bar{w}^* = X^+ y$$

Ans 7) Weight updating method in gradient descent :-

(•) $\hat{y}^{(i)} = \sum_{j=1}^{n} W_j x_j^{(i)} + W_0$

(•) $Error^{(i)} = y^{(i)} - \hat{y}^{(i)} = y^{(i)} - \left[\sum_{j=1}^{n} W_j x_j^{(i)} + W_0\right] = e^{(i)}$

$$E = \sum_{i=1}^{N} e^{(i)} = \sum_{i=1}^{N} \left[y^{(i)} - \sum_{j=1}^{n} W_j x_j^{(i)} + W_0\right]$$

(•) Applying least cost squared :-

$$E \rightarrow \frac{1}{2} E^2$$

$$\Rightarrow E = \frac{1}{2} \sum_{i=1}^{N} \left[y^{(i)} - \left[\sum_{j=1}^{n} W_j x_j^{(i)} + W_0\right]\right]^2$$

↳ E is our cost function, we must minimise 'E' :-

① Gradient descent updates every weight so we must find out how much every weight affects E :-

$$\frac{\partial E}{\partial W_j} = \frac{\partial \left[\frac{1}{2} \sum_{i=1}^{N} \left[y^{(i)} - \left[\sum_{j=1}^{n} W_j x_j^{(i)} + W_0\right]\right]^2\right]}{\partial W_j}$$

We know, $y^{(i)} - \left[\sum_{j=1}^{n} W_j x_j^{(i)} + W_0\right] = e^{(i)}$

$$\Rightarrow \frac{\partial E}{\partial W_j} = \frac{\partial \left[\frac{1}{2} \sum_{i=1}^{N} (e^{(i)})^2\right]}{\partial W_j}$$

$$\Rightarrow \frac{\partial E}{\partial w_j} = \frac{1}{2} \sum_{i=1}^{N} \frac{\partial (e^{(i)})^2}{\partial w_j}$$

remember, chain rule :-

$$\frac{d(f(x))^2}{dx} = 2f(x) \cdot f'(x)$$

$$\frac{\partial E}{\partial w_j} = 2 \cdot \frac{1}{2} \sum_{i=1}^{N} e^{(i)} \cdot \frac{\partial e^{(i)}}{\partial w_j}$$

$$\frac{\partial E}{\partial w_j} = \sum_{i=1}^{N} e^{(i)} \cdot \frac{\partial e^{(i)}}{\partial w_j}$$

for $$\frac{\partial e^{(i)}}{\partial w_j} = \frac{\partial \left[y^{(i)} - \left[\sum_{j=1}^{n} w_j x_j^{(i)} + w_0 \right] \right]}{\partial w_j}$$

$$= 0 - x_j \quad = -x_j^{(i)}$$

$$\therefore \frac{\partial E}{\partial w_j} = \sum_{i=1}^{N} e^{(i)} \cdot (-x_j^{(i)})$$

$$= - \sum_{i=1}^{N} e^{(i)} x_j^{(i)}$$

$$\Rightarrow \frac{\partial E}{\partial w_j} = - \sum_{i=1}^{N} \left[y^{(i)} - \left[\sum_{j=1}^{n} w_j x_j + w_0 \right] \right] x_j^{(i)}$$

Now, $\frac{\partial E}{\partial w_0} = \frac{\partial \left[\frac{1}{2} \sum_{i=1}^{N} (e^{(i)})^2 \right]}{\partial w_j} \quad ; j = 0$

$= 2 \cdot \frac{1}{2} \sum_{i=1}^{N} e^{(i)} \cdot \frac{\partial e^{(i)}}{\partial w_j} \quad ; j = 0$

$\Rightarrow \frac{\partial E}{\partial w_0} = \sum_{i=1}^{N} e^{(i)} \cdot \frac{\partial e^{(i)}}{\partial w_0}$

for, $\frac{\partial e^{(i)}}{\partial w_0} = \frac{\partial \left[y^{(i)} - \sum_{j=1}^{n} w_j x_j + w_0 \right]}{\partial w_0}$

$= 0 - 0 + 1$

$\Rightarrow \frac{\partial E}{\partial w_0} = \sum_{i=1}^{N} e^{(i)} \cdot 1$

Now, weight updates :-

$w_j \leftarrow w_j + \eta \frac{\partial E}{\partial w_j}$

$\Rightarrow w_j = w_j + \eta \sum_{i=1}^{N} \left[y^{(i)} - \left[\sum_{j=1}^{n} w_j x_j + w_0 \right] \right] x_j^{(i)}$

$w_0 \leftarrow w_0 + \eta \frac{\partial E}{\partial w_0}$

$\Rightarrow w_0 = w_0 + \eta \sum_{i=1}^{N} \left[y^{(i)} - \left[\sum_{j=1}^{n} w_j x_j + w_0 \right] \right]$