Vidyalankar Institute of Technology
(Accredited A+ by NAAC)
(Autonomous Institute Affiliated to University of Mumbai)

Mid Semester Examination

Branch	Date	Sem.	Roll No. / Exam Seat No.	Subject	Student's Signature	Junior Supervisor's Name and Sign
CMPN	19/03/24	VIII		DL		

Question No.	A	B	C	D	E	F	G	H	Total	Total out of (20 /30 / 40)
1										
2										
3										
4										

Examiners Signature	Student's Sign (After receiving the assessed answer sheet)

Q1. Solve any five

A) What is gradient

A gradient simply measures the change in all weights with regards to the change in error. The higher the gradient, the steeper the slope and the faster a model can learn.

B) Early stopping helps in reducing over fitting. It improves generalisation. It requires less amount of training data, and takes less time compared to other regularisation models.

c) Dropout is a regularization method approximating concurrent training of many neural networks with various designs. During training, some layer outputs are ignored or dropped at random, reducing the overfitting problems in deep learning.

D) The accuracy of predictions in supervised deep learning models depends to a large extent on the amount of data available to the model during training and the level of diversity in that data.

E yes, gradient descent can escape saddle points. gradient descent updates the parameters in the direction opposite to the gradient of the loss function to minimize it.

F) A high learning rate in gradient descent can cause overshooting, leading to instability and divergence away from the minimum of the loss function.

Momentum in optimization refers to the technique of incorporating past gradients to accelerate convergence and smooth out oscillations. It introduces a velocity term that accumulates ~~p~~ gradients over iterations, allowing for faster progress through flat regions and overcoming saddle points

The two steps of the gradient descent optimization are

(1) Compute gradient :- Calculate the gradient of the loss fuction with respect to the parameters

2. Update parameters :- Update the parameters in the opposite direction of the gradient to minimize the loss function.

Q2 A) Robustness to outliers

(1) L1 regularization (Lasso) is less robust to outliers compared to L2 regularization (Ridge) due to its tendency to shrink coefficients to zero, which can make the model overly sensitive to individual data points

(2) L2 regularization spreads the penalty more evenly across all weights, making it more robust to outliers

Penalty Term

(1) Sum of absolute values of weights

(2) Sum of squared magnitudes of weights

Effect

(1) Encourage sparsity, some weights become zero

(2) Encourages smaller weights, prevents overfitting

Computational Efficiency

L2 regularization is computationally more efficient to optimize compared to L1 regularization especially for large scale datasets, due to its smooth and convex penalty term.

Feature Correlation

L1 regularization tends to select one features, while L2 regularization spreads the penalty equally among them allowing correlated feature to share importance.

2B $x = [1, 2, 3, 4, 5]$

$y_{true} = [2, 4, 6, 8, 10]$

$w_0 = 0.5$ & $w_1 = 0.5$ $\lambda = 0.1$

$y_{pred} = w_0 + w_1 \times x$

$= 0.5 + 0.5 \times [1, 2, 3, 4, 5]$

$= [1.0, 1.5, 2.0, 2.5, 3.0]$

Compute the loss for each data point

$Loss_i = |y_{true} - y_{pred}| + \lambda \times (|w_0| + |w_1|)$

For $i = 1$

$Loss_1 = |2 - (1.0)| + 0.1 \times (|0.5| + |0.5|)$

$= 0.5 + (1.0 \times 1.0) = 0.6$

Similiary $Loss_2 = 2.6$

$Loss_3 = 4.1$

$Loss_4 = 5.6$

$Loss_5 = 7.1$

Finally total loss with L1 regularization is the sum of these individual Losses

$= 0.6 + 2.6 + 4.1 + 5.6 + 7.1$

$= 20.0$

So Total loss with L1 regularization for the given data is $\underline{\underline{20.0}}$

Q2c

1. Initialization: Initialize model parameters θ randomly
2. Iterative optimization
   - Shuffle training data to introduce randomness
   - For each data point (x_i, y_i) or mini-batch
     - Compute gradient $\nabla J(\theta; x_i, y_i)$ of the loss function with respect to the parameters
     - Update parameters:
       $\theta = \theta - \eta \nabla J(\theta; x_i y_i)$,
       where η is the learning rate
     - Repeat until all data points are used.

3. Termination: Repeat until convergence criteria are met or for a fixed number of epochs

SGD updates parameters more frequently benefiting from stochasticity to escape local minima and achieve faster convergence making it efficient for larger dataset

Q3(a)

given data $x = [3,4]$ and $y = [4,5]$ and $w = 0.5$ we have our goal to minimize the MSE (Loss function)

$$L = \frac{1}{n}(y_{pred} - y)^2$$

$$y_{pred} = 0.5 \times 3 = 1.5$$

$$g_t = dL/dw = \frac{2}{1} * (y_{pred} - 4) * x = -15$$

$$g_t^2 = (-15)^2 = 225$$

$$m_1 = 0.9 * 0 + (1 - 0.9) * -15 = -1.5$$

$$v_1 = 0.999 * 0 + (1 - 0.999) * 225 = 0.225$$

$\cancel{\hat{m}_2 = -0.8947}$ $\qquad \hat{m} = \frac{-1.5}{1 - 0.9} = -15$

$\cancel{\hat{v}_1 = 112.811}$ $\qquad \hat{v} = \frac{0.225}{1 - 0.999} = 225$

$$\hat{m} = -15$$

$$w_{t+1} = w_t - \left[\alpha \cdot \frac{1}{\sqrt{\hat{V}_t} + \varepsilon}\right] \times \hat{m}_1$$

$$0.5 - 0.1 \cancel{\left[\frac{-1.5}{\sqrt{225} + 10^{-8}}\right]} = \cancel{-0.04}$$

$$0.5 - \left[\frac{0.1}{\sqrt{225} + 10^{-8}}\right] \times 15$$

$$0.5 - (-0.1) = 0.5 + 0.1 = 0.6$$

$$w_{t+1} = 0.6$$

$y_{pred} = 0.5 \times 4$ — $y_{pred} = 0.6 \times 4 = 2.4$

$g_t = \frac{dL}{dw} = \frac{2}{1} * (2.4 - 5) \times 4 = -20.8$

$g_t^2 = (-20.8)^2 = 432.64$

$m_2 = 0.9 * -15 + (1 - 0.9) * (-20.8) = 11.42$

$v_2 = 0.999 * 225 + (1 - 0.999) * (432.64)$
$= 224.775 + (0.001 * 432.64$
$= 225.20$

$\hat{m}_2 = \frac{11.42}{1 - 0.9} = 114.2$

$\hat{v}_2 = \frac{225.20}{1 - 0.999} = 225200$

$0.6 - \left[\frac{0.1}{\sqrt{225200} + 10^{-8}} \right] \times 114.5$

$0.6 - 0.02$

$= 0.58$

Q3.B)

$$\Delta J(w) = \frac{1}{3}\left[2*(0.5\times[1-2])\times 1 + 2(0.5(2-4))\times 2 + 2(0.5(3-6)\times 3)\right.$$

$$= \frac{1}{3}\left[2\times(-1.5)\times 1 + 2\times(-3)\times 2 + 2\times(-5.5)\times 3\right)$$

$$= \frac{1}{3}\left[-3+(-12)+(-33)\right]$$

$$= \frac{1}{3}\times(-48) = -16$$

Initially
$G=0$ we have $G = 0+(-16)^2 = 256$

update the w using Adagrad update rule

$$w = w - \frac{\alpha}{\sqrt{G+\varepsilon}} \times \nabla J(0)$$

$$0.5 - \frac{0.1}{\sqrt{256+10^{-8}}} \times (-16)$$

$$= 0.5 + 0.1 = 0.6$$

After one iteration using the Adagrad the update weight $w = 0.6$.