Assignment-1

Solutions

1. Given dataset

$(x, y) = \{(1,1), (2,2), (3,2), (4,5)\}$

Model : $\hat{y} = \theta_1 x + \theta_2$

Residual : $r = \hat{y} - y$

$MSE = \frac{1}{n} \sum r^2$

a) $\theta = (1,0) \Rightarrow \hat{y} = x$

$x=1 : \hat{y} = 1 ; r = 1-1 = 0 ; r^2 = 0$

$x=2 : \hat{y} = 2 , r = 2-2 = 0 ; r^2 = 0$

$x=3 ; \hat{y} = 3 ; r = 3-2 = 1 ; r^2 = 1$

$x=4 : \hat{y} = 4 ; r = 4-5 = -1 , r^2 = 1$

$MSE = \frac{0+0+1+1}{4} = 2/4 = 0.5$

b) $\theta = (0.5, 1) \Rightarrow \hat{y} = 0.5x + 1$

$x=1 : \hat{y} = 1.5 , r = 1.5-1 = 0.5 , r^2 = 0.25$

$x=2 ; \hat{y} = 2 , r = 2-2 = 0 , r^2 = 0$

$x=3 ; \hat{y} = 2.5 , r = 2.5-2 = 0.5 , r^2 = 0.25$

$x=4 ; \hat{y} = 3.0 , r = 3.0-5 = -2.0 , r^2 = 4.00$

$MSE = \frac{0.25+0+0.25+4}{4} = \frac{4.5}{4} = 1.125$

Best fit : (1,0) because $0.5 < 1.125$

2. Given cost function

$J(\theta_1, \theta_2) = 8(\theta_1 - 0.3)^2 + 4(\theta_2 - 0.4)^2$

a) $J(0.1, 0.2) = 8(0.1 - 0.3)^2 + 4(0.2 - 0.4)^2$
$= 8(0.04) + 4(0.25)$
$= 0.32 + 1.00 = 1.32$

$J(0.5, 0.9) = 8(0.2)^2 + 4(0.2)^2$
$= 8(0.04) + 4(0.04) = 0.32 + 0.16 = 0.48$

b) closer to the minimum $(0.5, 0.9)$ since $0.48 < 1.32$

c) The parameter space is continuous & typically high-dimensional; picking pnts uniformly at random has a tiny chance of landing near the optimum, Gradient-based methods Exploit curvature (the gradient) to move toward lower cost systematically, whereas random guses throw away that information.

3.

Dataset $(1,3)(2,4),(3,6),(4,5)$

start $\theta^{(0)} = (0,0)$, step $\alpha = 0.01$

use MSE : $J = \frac{1}{2}\sum r_i^2$ with $r_i = \hat{y}_i - y_i$, $\hat{y}_i = \theta_1 x_i + \theta_2$

$$\nabla J = \begin{bmatrix} \frac{2}{n}\sum x_i r_i \\ \frac{2}{n}\sum r_i \end{bmatrix}$$

a) From $\theta^{(0)} = (0,0)$

Predictions all 0

Residuals r : $-3,-4,-6,-5$

Sums $\sum r = -18$;

$\sum xr = 1(-3) + 2(-4) + 3(-6) + 4(-5)$

$= -3 - 8 - 18 - 20 = -49$

Gradient

$$\nabla J^{(0)} = \left(\frac{2}{4}(-49), \frac{2}{4}(-18)\right) = (-24.5, -9)$$

update

$$\theta^{(1)} = \theta^{(0)} - \alpha \nabla J^{(0)} = (0,0) - (0.01)(-24.5, -9)$$
$$= (0.245, 0.09)$$

costs :-

$$J(\theta^{(0)}) = \frac{9+16+36+25}{4} = \frac{86}{4} = 21.5$$

$J(\theta^{(1)}) \approx 15.2560$ (computed from the new residuals)

b) Second step starting at $\theta^{(1)} = (0.245, 0.09)$

Predictions : $\hat{y} = 0.335, 0.58, 0.825, 1.07$

Residuals $r = \hat{y} - y$, $-2.665, -3.42, -5.175, -3.93$

Sums : $\Sigma r = -15.19$; $\Sigma xr = 1(-2.665) + 2(-3.42) + 3(-5.175) + 4(-3.93) = -40.75$

Gradient :

$$\nabla J^{(1)} = \left(\frac{2}{4}(-40.75), \frac{2}{4}(-15.19)\right) = (-20.375, -7.595)$$

Update

$$\theta^{(2)} = \theta^{(1)} - \alpha \nabla J^{(1)} = (0.245, 0.09) - 0.01\,(-20.375, -7.595)$$
$$= (0.448, 0.16595)$$

costs :-

$J(\theta^{(1)}) = 15.2560$

$J(\theta^{(2)}) = 10.9223$ (decreased again)

4. Given

dataset (1,2), (2,2), (3,4), (4,6)

MSE $J(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left(y^{(i)} - (\theta_1(x^{(i)}) + \theta_2)\right)^2$

a)

$(\theta_1, \theta_2) = (0.2, 0.5)$

$\hat{y} = (0.4, 0.9, 1.1, 1.3) \Rightarrow r = (-1.3, -1.1, -2.9, -4.7)$

$\Rightarrow r^2 = (1.69, 1.21, 8.41, 22.09)$

$$MSE = \frac{1.69 + 1.21 + 8.41 + 22.09}{4} = 8.35$$

$(\theta_1, \theta_2) = (0.9, 0.1)$

$\hat{y} = (1.0, 1.9, 2.8, 3.7) \Rightarrow r = (-1.0, -0.1, -1.2, -2.3)$

$\Rightarrow r^2 = (1.00, 0.01, 1.44, 5.29)$

$$MSE = \frac{1.00 + 0.01 + 1.44 + 5.29}{4} = 1.935$$

b) First GD step from (0,0), $\alpha = 0.01$

Residuals at (0,0): $-2, -2, -4, -6$; $\Sigma r = -14$; $\Sigma xr = -4$

$\nabla J = \left(\frac{2}{4}(-42), \frac{2}{4}(-14)\right) = (-21, -7)$

update: $\theta = (0.21, 0.07)$

MSE at $(0.21, 0.07) \cong 10.509$

c) The random guess (0.9, 0.1) got 1735, which beats the first GD step (≈ 10.51). A single GD step improves from the start but may still be far; random guesses can occasionally land closer to the optimum by luck.

5.

a) Underfitting

b) * Underfitting occurs when the model is too simple (or) not trained enough, so it cannot capture the underlying patterns in the training data.

* Because the model fails to fit even the training data well, both training error & test error remain high

* Common causes: model with low capacity (Eg: linear model for non-linear data), too much regularization (or) insufficient training.

c) * Increase model capacity - use a more complex model (Eg: add more features, use polynomial terms, deeper neural Network)

* Reduce regularization / train longer - relax constraints that prevent the model from fitting (Eg :- lower regularization strength, increase epoch, tune learning rate)

6.

a) Model A → Overfitting (training error = 0, test error high)

Model B → Underfitting (training error high, test error high)

b) Model A (Overfitting):

⇒ low bias → it learns training data very well

⇒ High variance → fails to generalize to unseen data

Model B (Underfitting)

⇒ High bias → model is too simple, can't capture patterns.

⇒ low variance → but still poor on both training and test.

c) Model A (Overfitting)

→ Add regularization (L1/L2, dropout, Early stopping)

→ Reduce model complexity (simple architecture / fewer features)

→ Get more diverse training data (or) use data augmentation.

Model B (Underfitting)

- → Use a more complex model (Eg: deeper NN, higher degree polynomial)
- → Train longer (or) regular reduce regularization
- → Improve feature engineering (add relevant predictors).