Machine Learning

700772579

Assignment:

1) Function Approximation by Hand

Given dataset,

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

$$\hat{y} = \theta_1 x + \theta_0$$

1. Model 1: $\theta = (1,0)$

$$\hat{y} = 1 \cdot x + 0 = x$$

X	Y	Prediction ($\hat{y}$)	Residuals ($y - \hat{y}$)	Square Residuals
1	1	1	0	0
2	2	2	0	0
3	2	3	-1	1
4	5	4	1	1

$$M.S.E = \frac{0+0+1+1}{4} = 0.5$$

2. Model 2 . $\theta = (0.5, 1)$

$$\hat{y} = 0.5x + 1$$

X	Y	Prediction ($\hat{y}$)	Residuals ($y - \hat{y}$)	Squared Resid...
1	1	1.5	-0.5	0.25
2	2	2	0	0
3	2	2.5	-0.5	0.25
4	5	3	2	4.6

$$m.s.E = \frac{4.5}{4} = 1.125$$

3. Conclusion

Model $\theta = (1, 0)$ has 0 lower m.s.E (0.5) compared to model $\theta = (0.5, 1)$ with m.s.E 1.125 so, model 1 fits the data

Activity-2 :- Random guessing. practical cost function

$$J(\theta_1, \theta_2) = 8(\theta_1 - 0.3)^2 + 4(\theta_2 - 0.7)^2$$

This is a quadratic cost function with a minimum at $\theta_1 = 0.3$, $\theta_2 = 0.7$

- Try guess: $\theta = (0.1, 0.2)$

$$J = 8(0.1 - 0.3)^2 + 4(0.2 - 0.7)^2 = 0.32 + 1 = 1.32$$

Try Guess : $\theta = (0.5, 0.9)$

$$J = 8(0.5 - 0.3)^2 + 4(0.9 - 0.7)^2$$

$$= 0.32 + 0.16$$

$$= 0.48$$

2. Which is better

→ $(0.5, 0.4)$ is closer to the true minimum & has a lower cost.

→ Random Guessing can sometimes land near the minimum, but it inefficient & unreliable.

3. Why is random guessing inefficient?

→ It doesn't use any information about the shape of the cost function

→ You're essentially throwing darts in the dark no guarantee of improvement

→ Gradient descent, by contrast, uses the slope to move intelligently towards the minimum.

3) Activity 3 - First Gradient Descent iteration

Dataset:

$(x, y) = \{(1,3), (2,4), (3,6), (4,5)\}$

We're fitting a linear model:

$\hat{y} = (\theta_0 + \theta_1 x)$

Start with:

→ $\theta^{(0)} = [0, 0]$

→ learning rate $\alpha = 0.01$

Step 1: predictions at $\theta(0)$

$\hat{y} = 0.x + 0 = 0$

Step 2: Residuals $(y - \hat{y})$

$r = [3, 4, 6, 5]$

Step 3: Compete Σr and $\Sigma x r$

$\Sigma r = 8 + 4 + 6 + 5 = 18$

$\Sigma x \cdot r = 1 \cdot 8 + 2 \cdot 4 + 8 \cdot 6 + 4 \cdot 5$

$= 8 + 8 + 18 + 20$

$= 49$

Step 4: Gradient

$\nabla 1 = \left[\frac{-2}{4} \Sigma r_i - \frac{2}{4} \Sigma x r\right]$

$= [-9, -24.5]$

Step 5: Update $\theta(1)$

$$\theta^{(1)} = \theta^{(0)} - \alpha J_1 = [0,0] - 0.01\,[-9, -24.5]$$

$$= [0.09, 0.245]$$

Step 6 : Compute Cost Before and After

Before ($\theta = [0,0]$):

$$J = \frac{1}{4}\sum x^2 = \frac{1}{4}(9+16+36+25) = \frac{86}{4} = 21.5$$

After ($\theta = [0.09, 0.245]$):

predictions :

$$\hat{y} = 0.09 + 0.245x$$

For $x = [1,2,3,4] \rightarrow \hat{y} \simeq [0.335, 0.58, 0.825, 1.07]$

Residuals :

$$y - \hat{y} = [2.665, 3.42, 5.175, 3.93]$$

Squared residuals $= [7.1, 11.7, 26.8, 15.4]$

$$= 61$$

$$J = \frac{61}{4} = 15.25$$

Result

→ Cost dropped from $21.5 \rightarrow 15.25$ in one step

→ Gradient descent is working - its moving toward minimum

Activity 4 : Compare Random Guessing & Gradient Descent

Dataset : $(x, y) = \{(1,2), (2,2), (3,4), (4,6)\}$

Random Guess1: $\theta = (0.9, 0.5)$

$\hat{y} = 0.5x + 0.2 \rightarrow [0.7, 1.2, 1.7, 2.2]$

Residuals : $[1.3, 0.8, 2.3, 3.8]$

Squared residuals : $[1.69, 0.64, 5.29, 14.44]$

$MSE = \frac{21.76}{4} = 5.49$

Random Guess 2 : $\theta = (0.9, 0.1)$

$\hat{y} = 0.1x + 0.9 \rightarrow [1.0, 1.1, 1.2, 1.3]$

Residuals : $[1.0, 0.9, 2.8, 4.7]$

Squared residuals : $[1.0, 0.81, 7.84, 22.09]$

$MSE = \frac{31.74}{4} = 7.94$

Gradient Descent $(\theta = [0,0], \alpha = 0.01)$

Initial predictions : $[0,0,0,0]$

Residuals : $[2,2,4,6]$

$\Sigma r = 14, \Sigma x r = 42$

Gradient : $[-7, -21]$

$b(1) = [0.07, 0.21]$

New predictions : $[0.28, 0.49, 0.7, 0.91]$

Residuals : $[1.72, 1.51, 1.3, 5.09]$

Squared residuals : $[2.96, 2.28, 10.87, 25.9]$

$$MSE = \frac{42.03}{4} = 10.51$$

Comparision

- Random Guess had the lowest Error (5.49) MSE
- Gradient Descent start rough but improves with each step

1. Random guessing can occasionally land near the minimum, but its unreliable
- Gradient Descent is systematic - it guarantees improvement over time
- With more iterations, Gradient Descent Outperform any random guess

Activity 5 - Recognizing Underfitting and Overfitting

1. This is Underfitting

2. Why does this happen?

Underfitting occurs when the model is too simple to capture the underlying patterns in the data. It fails to learn from the training data, which leads to:

→ High training error: The model doesn't even fit the data it was trained on.

→ High test error: Since it didn't learn the pattern, it performs poorly on new data too.

3. Two possible fixes:

→ Increase model complexity

→ Improve feature Engineering

Activity 6 - Comparing Models

1. Model A is Overfitting: It memorizes the training data but fails to generalize

Model B is Underfitting: It doesn't learn enough from the training data and perform poorly overall.

2. Bias - Variance Tradeoff :-

Model	Bias	Variance
Model A	Low bias	High Variance
Model B	High bias	low Variance

→ Overfitting : low bias, high variance : model is too sensitive to training data.

→ Underfitting : High bias, low variance : model is too rigid to learn pattern

3. Recommendations to improve each model :

→ Model A (Overfitting) :-

i) Add regularization (L1 or L2)

ii) Reduce model complexity

iii) Use Cross - Validation to tune hyperparameters

iv) Collect more training data

→ Model B (Underfitting) :-

i) Use a more Expressive model

ii) Improve feature selection or transformation

iii) Reduce Regularization

iv) Train longer or optimize better.