Question 1 (i): Least Squares Solution using Analytical Method

Analytical Least Squares Solution

The least squares solution finds the optimal weights that minimize the residual sum of squares in a linear regression problem. The closed-form solution for the weights is given by the normal equation:

w_ML = (X^T X)^-1 X^T y

Where:

- X is the matrix of input features (with an added bias term),

- y is the target vector (actual outputs),

- w_ML is the vector of weights that minimizes the error.

In this part, we will implement the least squares solution from scratch using matrix operations.

Steps

In the following code, we will:

1. Load the dataset.
2. Prepare the input matrix X, by adding a bias term.
3. Compute the least squares solution using the formula w_ML = (X^T X)^-1 X^T y

Question 1 (ii): Gradient Descent Algorithm

Gradient Descent is an iterative optimization algorithm used to minimize a function by iteratively moving towards the direction of the steepest descent. In the context of linear regression, the goal is to minimize the cost function, which is the mean squared error between the predicted and actual target values.

Cost Function:

The cost function J(w) is defined as:

J(w) = (1/2m) * sum((y^(i) - X^(i)w)^2)

Where:

- m is the number of training examples,

- X^(i) are the features for the i-th training example,

- y^(i) is the actual target for the i-th example,

- w is the weight vector we are optimizing.

Gradient Descent Update Rule:

The update rule for Gradient Descent is:

w_t+1 = w_t - eta * gradient(J(w_t))

Where:

- w_t is the weight vector at iteration t,

- eta is the learning rate,

- gradient(J(w_t)) is the gradient of the cost function.

Question 1 (iii): Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD) is a variation of gradient descent where the model updates weights based on small subsets of the dataset, rather than the full dataset, in each iteration.

Advantages of Stochastic Gradient Descent:

1. Faster Convergence: Since SGD uses mini-batches of data instead of the full dataset, it updates weights more frequently, which leads to faster convergence compared to batch gradient descent.

2. Works Well for Large Datasets: For large-scale machine learning tasks, using the entire dataset for each update can be computationally expensive. By using smaller mini-batches, SGD is more computationally efficient.

Disadvantages of Stochastic Gradient Descent:

1. Noisy Updates: The mini-batch updates can introduce noise in the gradient, making the path to convergence less stable compared to batch gradient descent.

2. Hyperparameter Sensitivity: The learning rate and mini-batch size must be carefully chosen for optimal performance.

Steps in SGD:

1. Randomly shuffle the dataset to ensure randomness in mini-batch selection.

2. Select a mini-batch of 100 samples and compute the gradient for the mini-batch.

3. Update the weights using the gradient of the mini-batch.

4. Repeat the process until the weights converge.

Question 1 (iv): Ridge Regression with Gradient Descent

Ridge regression, also known as Tikhonov regularization, introduces a penalty term to the cost function to prevent overfitting. The penalty is proportional to the magnitude of the weights.

Cost Function for Ridge Regression:

J_ridge(w) = (1/2m) * sum((y^(i) - X^(i)w)^2) + lambda * |w|^2

Where:

- lambda is the regularization parameter.

Benefits of Ridge Regression:

1. Prevents Overfitting: The regularization term helps control the complexity of the model by penalizing large weights, leading to better generalization.

2. Improves Model Stability: Ridge regression can stabilize the model when multicollinearity is present in the dataset.

Steps in Ridge Regression with Gradient Descent:

1. Compute the gradient of the cost function with the added regularization term.

2. Update the weights using gradient descent.

3. Use cross-validation to find the optimal regularization parameter (lambda).

Question 1 (v): Kernel Ridge Regression

Kernel Ridge Regression is a powerful tool for modeling non-linear relationships by applying a kernel function to transform the data into a higher-dimensional space.

Key Concepts:

1. Kernel Function: The kernel function allows us to compute the similarity between data points in the transformed space without explicitly transforming the data.

2. Radial Basis Function (RBF) Kernel: One of the most commonly used kernels, it is effective at capturing non-linear relationships.

The RBF Kernel is defined as:

K(x, x') = exp(-(||x - x'||^2) / (2*sigma^2))

Advantages of Kernel Ridge Regression:

1. Captures Non-Linearity: By using the RBF kernel, we can model complex non-linear relationships that cannot be captured by standard linear regression methods.

2. Flexibility: The kernel trick allows us to use a variety of kernels, such as polynomial or sigmoid kernels, to fit the data more effectively.

Steps in Kernel Ridge Regression:

1. Compute the kernel matrix using the RBF kernel.

2. Solve the ridge regression problem in the kernel space by optimizing the weights.

3. Use cross-validation to determine the best hyperparameters, such as the regularization parameter (lambda) and the kernel bandwidth (sigma).

Note – I have given more details in Jupyter Notebook. (I am more familiar with that. So, I have used the text blocks of jupyter to explain.