DSW_ASSIGNMENT-3

Untitled Attachment

1. Explain normal equation method to estimate the parameters of multiple linear regression. Write a function for it

Ans-

The Normal Equation is a method used to find the exact solution to the linear regression problem. It's an analytical approach to Linear Regression with a Least Square Cost Function. We can directly find out the value of θ without using Gradient Descent. Following is the Normal Equation formula:

$\theta=(XTX)-1XTy$θ=(XTX)−1XTy

Here:

- X is the matrix of input features,
- y is the vector of target values,
- θ is the vector of model parameters.

```
import numpy as np
def normal_equation(X, y):def normal_equation(X, y):
# Add a column of ones for the bias term # Add a column of ones for the bias term
X_b = np.c_[np.ones((X.shape[0], 1)), X]X_b = np.c_[np.ones((X.shape[0], 1)), X]
```

```
# Compute theta using the Normal Equation# Compute theta using the Normal
Equation
theta = np.linalg.inv(X_b.T.dot(X_b)).dot(X_b.T).dot(y)theta =
np.linalg.inv(X_b.T.dot(X_b)).dot(X_b.T).dot(y)
return thetareturn theta
X = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
y = np.array([2, 5, 8])
theta = normal_equation(X, y)
print(theta)
```

2. A college professor believes that if the grade for internal examination is high in a class, the grade for external examination will also be high. A random sample of 10 students are selected, and the data is as follows: Marks_internal examination (X)= [15, 23, 18, 23, 24, 22, 22, 19, 19, 16] Marks_external examination (Y) = [49, 63, 58, 60, 58, 61, 60, 63, 60, 52] Find the polynomial regression model of degree 3 and linear regression model from the above data

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures

# Define the data
X = np.array([15, 23, 18, 23, 24, 22, 22, 19, 19, 16]).reshape(-1, 1)
Y = np.array([49, 63, 58, 60, 58, 61, 60, 63, 60, 52])


linear_model = LinearRegression().fit(X, Y)

print("Linear model coefficients:")
print("Slope (a1):", linear_model.coef_[0])
print("Intercept (a0):", linear_model.intercept_)
```

```
poly = PolynomialFeatures(degree=3)
X_poly = poly.fit_transform(X)


poly_model = LinearRegression().fit(X_poly, Y)


print("\nPolynomial model coefficients:")
print("a3, a2, a1, a0:", poly_model.coef_)

Linear model coefficients:
Slope (a1): 1.0416197975253094
Intercept (a0): 37.463442069741276
Polynomial model coefficients:Polynomial model coefficients:
a3, a2, a1, a0: [ 0.00000000e+00 1.57135372e+01 -3.66067751e-01
-2.86182579e-04]
```

3. Explain Gradient Descent method in context of linear regression.

Gradient Descent is an iterative optimization algorithm that tries to find the optimum value (Minimum/Maximum) of an objective function. It is one of the most used optimization techniques in machine learning projects for updating the parameters of a model in order to minimize a cost function.

Steps Required in Gradient Descent Algorithm

- **Step 1**we first initialize the parameters of the model randomly

- **Step 2**Compute the gradient of the cost function with respect to each parameter. It involves making partial differentiation of cost function with respect to the parameters.
- **Step 3** Update the parameters of the model by taking steps in the opposite direction of the model. Here we choose a hyperparameter learning rate which is denoted by alpha. It helps in deciding the step size of the gradient.
- **Step 4**Repeat steps 2 and 3 iteratively to get the best parameter for the defined model

Cost Function

$$J\left(\Theta_0, \Theta_1\right) = \frac{1}{2m} \sum_{i=1}^{m} [h_\Theta(x_i) - y_i]^2$$

$h_\Theta(x_i)$: Predicted Value

y_i: True Value

Gradient Descent

$$\Theta_j = \Theta_j - \alpha \frac{\partial}{\partial \Theta_j} J\left(\Theta_0, \Theta_1\right)$$

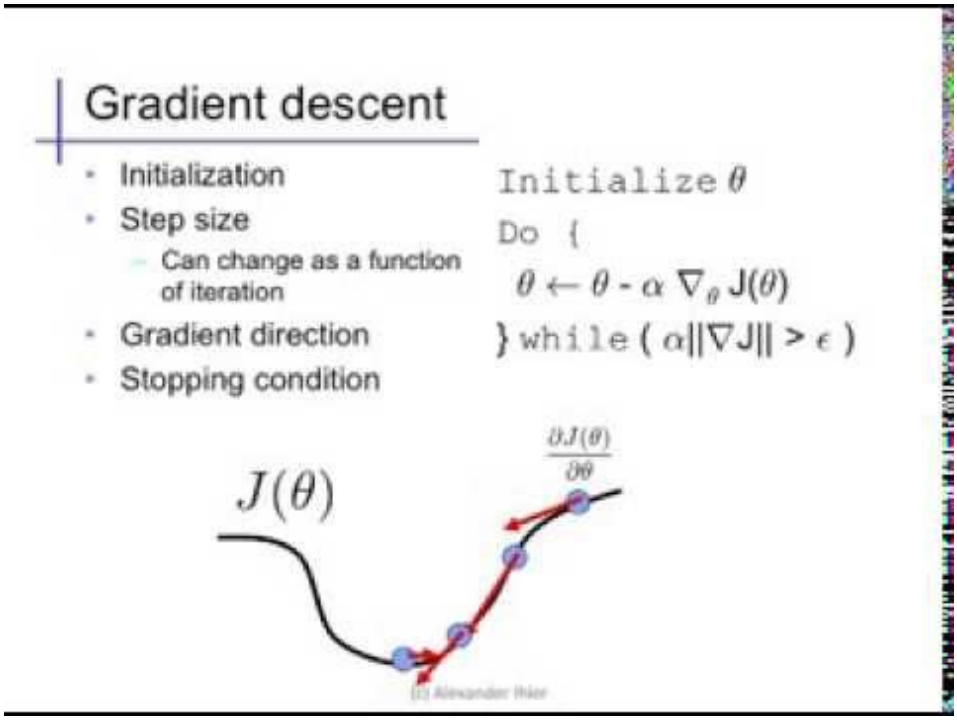
α: Learning Rate

Now,

$$\begin{aligned}
\frac{\partial}{\partial \Theta} J_\Theta &= \frac{\partial}{\partial \Theta} \frac{1}{2m} \sum_{i=1}^{m} [h_\Theta(x_i) - y]^2 \\
&= \frac{1}{m} \sum_{i=1}^{m} (h_\Theta(x_i) - y) \frac{\partial}{\partial \Theta_j} (\Theta x_i - y) \\
&= \frac{1}{m} (h_\Theta(x_i) - y) x_i
\end{aligned}$$

Therefore,

$$\Theta_j := \Theta_j - \frac{\alpha}{m} \sum_{i=1}^{m} [(h_\Theta(x_i) - y) x_i]$$



4. List and explain the issues of Linear Regression model.

Ans-

Linear regression is a widely used algorithm for modeling the relationship between a dependent variable (target) and one or more independent variables (features). However, it has several limitations and issues that need to be considered:

1. Non-Linearity of the Response-Predictor Relationships:
2. Correlation of Error Terms
3. Non-Constant Variance of the Error Term (Heteroscedasticity)
4. Collinearity

5. Write the difference between gradient descent and normal equation method for regression coefficients estimation.

Gradient Descent	Normal Equation
Gradient Descent is an iterative optimization algorithm used to find the values of parameters that minimize a cost function. It starts with random values of parameters and then keeps changing the parameters to reduce the cost function until it reaches a minimum	**Normal Equation** is an analytical approach used for optimization. It directly computes the parameters of the model that minimizes the sum of the squared difference between the actual term and the predicted term of the dataset[1]
Gradient Descent can converge even if the learning rate is kept fixed. However, it requires us to define a learning rate that controls the size of the steps taken towards the minimum of the loss function	**Normal Equation** doesn't require us to define a learning rate because we are not taking iterative steps - we get the results directly. But finding the inverse of the matrix is computationally very expensive in large datasets
Gradient Descent is one of the most used optimization techniques in machine learning projects for updating the parameters of a model in order to minimize a cost function.	**Normal Equation** is very effective when you are working with smaller datasets. It provides a direct solution without the requirement for iterative updates.

6.What is difference between simple linear and multiple linear regressions?

Simple Linear Regression

Simple linear regression is a statistical method used to model the relationship between one independent variable (X) and one dependent variable (Y). The goal is to find the best-fitting linear relationship between the two variables, which is represented by the equation Y = a + bX, where:

- Y is the dependent variable
- X is the independent variable
- a is the intercept (the value of Y when X is 0)
- b is the slope (the change in Y for a one-unit change in X)

The slope and intercept are estimated using a method called least squares regression, which finds the line that minimizes the sum of the squared differences between the actual Y values and the predicted Y values.

Multiple Linear Regression

Multiple linear regression is an extension of simple linear regression tat allows for the modeling of the relationship between one dependent variable (Y) and multiple independent variables (X1, X2, X3, etc.). The equation for multiple linear regression is:

$Y = a + b1X1 + b2X2 + b3X3 + ... + bnXn$

Where:

- Y is the dependent variable
- X1, X2, X3, ..., Xn are the independent variables

- a is the intercept (the value of Y when all of the independent variables are 0)
- b1, b2, b3, ..., bn are the coefficients (the change in Y for a one-unit change in the corresponding independent variable)

Like simple linear regression, the coefficients and intercept are estimated using least squares regression. The goal is to find the set of coefficients that minimizes the sum of the squared differences between the actual Y values and the predicted Y values.

Key Differences

The key differences between simple linear and multiple linear regression are:

1. Number of Independent Variables: Simple linear regression involves only one independent variable, while multiple linear regression involves two or more independent variables.
2. Predictive Power: Multiple linear regression generally has greater predictive power than simple linear regression because it can account for the combined effects of multiple independent variables.

3. Interpretation: The interpretation of the coefficients in multiple linear regression is more complex than in simple linear regression because each coefficient represents the effect of one independent variable while controlling for the effects of the other independent variables.

7. Suppose we have data for a retail company. The company wants to understand how their advertising expenses in various channels (e.g., TV, Radio) impact sales. Predict sales using Multiple Regression model using both TV and Radio advertising expenses for the following dataset.

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression

# create a DataFrame from the data
data = {'TV': [127.44, 135.76, 130.14, 127.24, 121.18, 132.29, 121.88,
144.59, 148.18, 119.17],
'Radio': [66.95, 56.75, 68.38, 74.05, 56.22, 64.40, 64.80, 64.31, 55.58,
73.82],
'Sales': [716.54, 660.01, 658.56, 679.68, 632.94, 751.38, 691.32, 732.85,
691.15, 693.58]}
df = pd.DataFrame(data)

# create a matrix of features (TV and Radio advertising expenses)
X = df[['TV', 'Radio']].values

# create a vector of target values (sales)
y = df['Sales'].values
```

```
# create a LinearRegression object
model = LinearRegression()

# fit the model to the data
model.fit(X, y)

# make predictions on new data
X_new = np.array([[150, 60]]).reshape(1, -1)
y_pred = model.predict(X_new)

# print the predicted sales
print('Predicted sales:', y_pred[0])

Predicted sales: 718.2691255657235
```

/

Untitled Attachment

Untitled Attachment