Assignment-3

Ginjala Lohith Reddy(210050054)

May 22, 2023

1 What I have learnt

In this week we dive little bit deep in Machine Learning. This week I have learnt about **Linear regression** and **Logistic regression**. I have learnt about the cost functions, derivatives of cost functions with respect to corresponding features, learning rate etc for both methods. I have learnt about Gaussian noise, Mean Squared Error, partial derivatives etc. I have learnt about different in-built functions of **NumPy** Library in Python like **np.mean()**, **np.dot()**, **np.array()**, **np.exp()**, **np.random.normal()**, **np.random.uniform()** etc. I have learnt about **graph descent algorithm**, Support Vector Machines(**SVM**), **kernels** and **Naive Bayes**.

2 Linear Regression

The relationship between a dependent variable and one or more independent variables can be modelled statistically using linear regression. The relationship between the variables is assumed to be linear, which means that changes in the independent variable(s) will directly affect changes in the dependent variable. Finding the best-fitting line to depict the relationship between the variables is the aim of linear regression. The slope and intercept coefficients that minimise the discrepancy between the expected values of the dependent variable and the actual observed values are estimated to produce this line.
In this assignment we have taken linear combination with 1 input features with slope = 2 and intercept = 5 i.e. (Y = 2*X + 5). Then we have manually added a varying gaussian noise with range -20 to 20 to our Y. The obtained values are considered as our **data**. Now I have computed a cost function for linear regression as mentioned in the video. I have taken mean instead of sum to omit overflow of cost function. Then I have written functions for partial derivatives with respect to A, B(Parameters of the Assignments) using the above computed cost function. Now we initialise A, B to '0' and two lists for storing A,B values. Learning rate variable is taken uniformly from **0.0001** to **0.0006** and modified while iterating through number of iterations and printing cost for all iterations. Now we compute initial cost with above A,B values and stores '5' A, B values with changing A, B values using the partial derivatives with respect to A, B and learning rate. Now we print the cost values and we can see that cost is decreasing for every iteration. Now I have computed models for input feature(x) with the values of A, B from the lists

computed. After computing we plot the graphs for **y_true** with **Black color** and other models with other colors. As we can see the models coincide with each other and have very slight deviations from **y_true**. I have tried with different A, B values, changed the number of iterations and also changed the learning rates. **Learning Rate:** the learning rate is a **hyperparameter** that controls the step size at which a model's parameters are updated during the training process. It determines how quickly or slowly the model learns from the data. Learning rate should not be very low or very high because model learning may take many iterations or model might leave some important points during iterations. I have observed that **learning rate $>$ 0.0006** and **learning rate $<$ 0.0001** the cost values increase as we iterate through loop. I have changed the number of iterations from 2000 to 1500, 1200, 1000, 900 and sometimes 100 also works out(for some values of A, B) and I haven't seen significant differences from **number of iterations = 2000** from the changed values. This may be due to the precise selection of learning rate. Hence, we can say that selecting correct learning rate will decrease the number of iterations to train our model.

3 Logistic Regression

Similar to what we have done in the linear regression we find the data points and compute cost function, partial derivatives of input features as explained in the video and learning the models and plotting the graphs of true values and different models. But there is difference between cost function which inturn lead to change in partial derivatives functions. There is a introduction of **sigmoid** function for computing h(x) function and from there we compute cost function and minimize the cost function. Similarly we plot the graphs and see the difference of models in plots.