Information Engineering and Technology Faculty
German University in Cairo

Report
Machine Learning Assignment 1 & 2

Done by : Nour El-Din Mohamed Samir 40-3993

Contents:

Introduction

Machine learning is the science of getting computers to learn without being explicitly programmed . Computers can learn , adapt and automatically improve through experience and collecting data to train on . From the types of Machine learning are supervised learning and unsupervised learning. The supervised learning is defined by training the algorithms on labeled data that classify data and predict outputs accurately . The unsupervised learning is defined by training the algorithms on unlabeled data where the algorithm works on its own to discover and analyze data to find hidden patterns from given data. Supervised learning has two types:

1. **Regression** : The output is continuous.
2. **classification** : The output is discrete.

Linear Regression is a regression supervised learning algorithm. Linear Regression is a linear approach to model the relation between a scalar response and one or more explanatory variables (dependent and independent variables). Incase of one variable, it is called linear regression with one variable . Incase of multiple variables, it is called linear regression with multiple variables. The relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. On the other hand, the logistic regression which is a classification supervised learning algorithm. Logistic regression aims to learn a mapping from x inputs to y outputs, where y= {1,2,...C}, with being C the number of classes. Moreover, logistic regression is a statistical model that uses a logistic function to model a binary dependent variable through estimating the parameters of the logistic model. In this assignment we implemented the supervised machine learning algorithm Linear Regression and Logistic Regression .Each assignment will be discussed clarifying the steps and techniques applied.

Assignment 1:

In assignment 1 , Linear regression with one variable and multiple variables was applied. At first, I had the outline code of the algorithm and the needed functions predefined and coded in the code. After studying the linear regression algorithm I was able to fill in the missing parts of the code which was mainly the mathematical equations of the functions. At first , reading the dataset of the linear regression with one variable which consisted of two columns one is the feature (x variable) and the other column is the predicted price (y variable). Equation[1] shoes the hypothesis function.

$$h_\theta(x) = \theta^T x = \theta_0 + \theta_1 x_1$$

Equation[1]:Linear regression Hypothesis Function.

H : is the hypothesis
X1: is the variable or feature

Choosing theta 0 and theta 1 in order to make h(x) nearly equal or close to y.
So by applying the following equation(2) the difference between h(x) and y could be

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2$$

Equation[2]: Cost Function

Calculated and our aim is to minimize the J cost function. Using the gradient descent function helped in choosing the optimal thetas as we keep trying different thetas till to reduce the cost function. In the algorithm code there were two functions: the cost function to calculate J and the gradient descent function to try different thetas values. The cost function was called inside the gradient descent function to see the effect of new thetas value. Gradient descent is repeated until convergence and parameters are updated simultaneously. The new values for thetas are calculated through a certain equation which contains the alpha which is the learning rate which controls the sizes of taken steps. On the other hand , the second part of the assignment was the linear regression with multiple variables. The same equations and steps as with the one variable are followed with very small differences in the equations. The difference is that I had more features (x variables) , so more thetas . The hypothesis function is now for n features. So we have ($x1,x2,...xn$) and ($\theta1,\theta2,...\theta n$). For example the dataset had 3 columns , 2 columns for the features ($x1$ and $x2$ variables) and a third column y the predicted output. A columns of ones was concatenated to the feature columns to define the zero feature $x0$ after normalizing the data to scale the features as It is found that during gradient descent if the features are on the same scale then the algorithm converges faster than when the features are not appropriately scaled in the same range. Last but not least, another task was added on this assignment which was adding some of the machine learning techniques on the linear regression multiple variables algorithm. Model selection was added to the dataset of multiple variables (18 features), the dataset was split into training data, validation data and testing data with ratios 60%, 20% and 20% respectively of the original dataset.

- **Training Data :** It is used to calculate the optimal thetas value that gives the minimum cost J for each degree.

- **Validation Data :** It is used to find which degree will give the minimum cost error and the optimal thetas that obtained from each degree at the training will be used as a parameter to call the ComputeCostMulti function.

- **Testing Data:** It is used to test the data at the polynomial degree with minimum error cost obtained from the validation data.

Moreover, a feature reduction technique was applied before splitting the dataset inorder to get rid of the low correlated features and improve the performance and accuracy of prediction. After plotting a map showing the correlation features values with each other and analyzing the correlation values of the features with the price feature (first row in the plot). The features with correlation values above or equal 0.5 were taken and other features were removed from the dataset . Only 5 features were taken instead of 18 features which decreased run time complexity.

Assignment 2:

In assignment 2, a Logistic regression algorithm was applied. At first we had two datasets one was applied on the logistic regression without regularization and the other was applied on the logistic regression using regularization. Datasets were labeled using negative (0) and positive (0) classes. Starting with logistic regression without regularization, the aim was to predict the hypothesis function h(x), where 0<= h(x) <= 1. The predicted h(x) is compared to a threshold, if h(x) < threshold, then it should predict label=1 and if h(x) > threshold, then it should predict label=0. First, the sigmoid is calculated using the sigmoid function in the code as the logistic regression hypothesis is defined as the following equation[3]:

$$h_\theta(x) = g(\theta^T x)$$

Equation[3]: Logistic Regression Hypothesis

Then, thetas values are choses to start with and these values keep changing using gradient descent to reach optimal values of thetas to decrease error cost to minimum. Gradient descent is used till convergence is reached. In the code i had a function called costFunction where gradient descent is applied on thetas and error cost is calculated after gradient descent. The minimum cost was calculated using the optimize.minimize pre defined python function giving it the parameters initial thetas values and costFunction function. The optimize.minimize function outputs the minimum cost and the optimal theta values to obtain this minimum cost. Moreover, the predict function takes these previously mentioned optimal thetas and feature columns (x) and calculates out the predicted label or class (0 or 1). This predicted label is compared to the actual label (y) in the dataset and the accuracy of these predictions is calculated. Moreover , the dataset was then split into train and test datasets as there was no need for the validation as there was only thetas parameters, so data is trained to get optimal thetas for lowest error cost . Then these thetas values were used in the testing dataset to predict the price(y value 0 or 1). On the other hand, the linear regression with regularization was applied on the second dataset. Regularization makes the hypothesis simpler. Regularization keeps all the features, but it reduces the parameters of these features by adding the regularization term to cost function as in the following equation[4]:

$$J(\theta_0, \theta_1, .., \theta_m) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \frac{\boldsymbol{\lambda}}{\mathbf{2m}}\sum_{j=1}^{n}\boldsymbol{\theta}_j^2$$

Equation[4]: Cost Function of logistic regression with regularization

The lambda is the regularization parameter. The linear regression with regularization follows exactly the same steps and logic as the linear regression without regularization. Lastly, The dataset applied on the linear regression with regularization was split to train data, validate data and test data with ratios 60%, 20% and 20% respectively.

- **Training Data :** It is used to calculate the optimal thetas value that gives the minimum cost J. Different values of lambda were used , so from the training data the lambda value with the least cost is known with its optimal thetas values.

- **Validation Data :** Using the thetas obtained from the training data we compute the minimum cost error on the validation data with lambda equals zero.

- **Testing Data:** Using the best combo of lambda and thetas from training and validation data to test the testing data.

Results:

The results were more obvious and comparable at assignment 2. The logistic regression algorithm without regularization showed accuracy of 89% , then it showed 86.67% after splitting data into training data and testing data. On the other hand, the logistic regression with regularization showed accuracy of 89%, then it showed 60% accuracy after splitting data into training, validation and testing datasets.