PROJECT REPORT

Mentors:

Aakash Rana Sir

Sarjita Patra Ma'am

Aryan Rajani

IIT (ISM) Dhanbad – 826004

Mathematics and Computing

First Year

Contents

Overview

The problem statement for ML Bootcamp lasted just over 5 weeks where we as students were given the opportunity to learn and implement machine learning algorithms in Python We learnt how to manage and work with huge datasets and apply machine learning algorithms like Linear Regression, Polynomial Regression, Logistic Regression, K Nearest Neighbours (KNN), K-Means Clustering and Neural Networks on them.

Features

- **Linear Regression (Polynomial Regression)**

  I started working with a single feature and about a hundred training examples to understand the core functionality of the training process of the model. During this I broadened my scope to two training features and learnt the vectorized approach towards the calculation of cost and hypothesis. This vectorized approach decreased the effort and computational time required, by using matrices and its properties. When I felt comfortable with this vectorized approach, I used the entire data to train the model, and then calculated the Room Mean Square Error (RMSE) and concluded linear regression.

- **Polynomial Regression**

  Using the knowledge gained in linear regression, I trained my model. The only added part was adding features which was done using np.multiply function where all possible combinations were taken up till n^{th} degree. I implemented L2 regularization in my model since I encountered the problem of overfitting of my data.

  RMSE (Degree 1) = 5.769233409555011
  RMSE (Degree 2) = 4.158575950667938
  RMSE (Degree 3) = 4.3593333071132845

● Logistic Regression

For logistic regression, we work with probabilities of occurrence of certain outputs over some others. For this we based our model on the sigmoid function which outputs values between 0 and 1, where positive arguments give probability above 0.5 and vice-versa for negative arguments. The model also carries an essence of one versus all classification where for every training example, the labels were converted to a binary form in the form of an array of length 26 for each training example, and eventually a matrix of shape 26 by number of training examples. The model was trained using the logistic cost function and the updation process was done with use of gradient descent. Using the optimized value of weights, predictions were made on the testing data and accuracy was calculated by first converting the prediction to the binary form and then to an array of size corresponding to the number of testing examples. A simple operation then gives the required accuracy of the model.

Accuracy = 75.567%

● K – Nearest Neighbour (KNN)

For this algorithm we had to work with distances between a training example and a testing example. I used the logic of repeating the testing example into a matrix spanning all the training examples for simulation calculation of distance to reduce the computational time. Using the acquired distances, we choose the 'k' number of neighbours which have the least distance and the training labels that occur the maximum number of times, is the label given to the testing example, using these labels, we calculate the accuracy of the model. For accuracy calculation, only 1000 testing examples were taken as computational time was becoming too large.

Accuracy = 79.1%

● K – Means Clustering

This model groups the given data into clusters based on the similarity between them. I used the logic of repeating the testing examples and finding the difference between the testing example and coordinates of the various clusters, and further used it to calculate the distance of the testing example and cluster centers. Then take the minimum of the distance and assign the appropriate cluster. Using the examples in the clusters, the new centre is found (similar to calculation of centre of mass), then using the new centres, we iterate over the same and finalise the clusters each example belongs to.

● Neural Networks

Neural Network is the heart of deep learning and tries to imitate the functionality of the human brain. Neural networks are powered by the various layers that form it, forward propagation and back propagation. The implementation in neural networks required us to create weights and biases which depended on the number of hidden layers. Each layer has nodes which have their own values which depend on the weights, biases and values stored in the nodes of the previous layers, and this constitutes the forward propagation. The power of a neural network lies in its back propagation which involves calculation of cost and gradient of various weights and biases, these weights and biases are then updated and optimized to fit our model. The code is also written in such a way that the user can input the number of hidden layers they require as well as the number of nodes they need in each hidden layer. After the final iteration, the weights and biases are then used to forward propagate once on the testing data and then calculation of accuracy is done using the logic of the one versus all classifier (similar to that in logistic regression)

Accuracy on training data = 53.567% (Two layer, 60 nodes(hidden))

Accuracy on testing data = 52.074% (Two layer, 60 nodes (hidden))

Technology Stack

- Python
- NumPy (Python library)
- Pandas (Python library)
- Matplotlib (Python library)
- Jupyter Notebook
- Google Colaboratory

About Me

- **Personal Details**

Name	Aryan Rajani
Branch	Mathematics and Computing
College	IIT (ISM) Dhanbad
Year	First Year
Contact	+91 7620597828
E – Mail Address	aryanrajani2003@gmail.com 21je0172@iitism.ac.in
LinkedIn	https://www.linkedin.com/in/aryan-rajani-33363b227/