BITS F464 - Machine Learning

Assignment-2A: Logistic Regression

Deadline: ~~17/04/2021, 23:59~~ 19/04/2021, 1300hrs

1 General Instructions

1. This assignment is a coding project and is expected to done in groups. Each group can contain at most **three** members. Make sure that all members in the group are registered to this course and please try to maintain the same group for all the assignments.
2. This assignment is expected to be done in Python using standard libraries like NumPy, Matplotlib and Pandas. You can use Jupyter Notebook and any other python in-built data structure or library. No other ML libraries like scikit/sklearn, TensorFlow, Torch etc. should be used.
3. Refrain from directly copying codes/snippets from other groups or the internet as all codes will be put through a plagiarism check.
4. All deliverable items (`.py` files, `.ipynb` files, report, images) should be put together in a single `.zip` file. Rename this file as `LR_`⟨`id-of-first-member`⟩ (preferably ID number of the student submitting) before submission.
5. Submit the zip file on CMS/GForms on or before the aforementioned deadline. Please note that this is a hard deadline and no extensions or exemptions will be given. The demos for this assignment will be held on a later date which shall be conveyed to you by the IC. All group members are expected to be present during the demo.
6. The link to the dataset for this assignment can be found here.

2 Problem Statement

1. In this assignment, you will implement logistic regression from scratch for binary classification. To train the model, you will use Gradient Descent and Stocastic Gradient Descent with appropriate learning rates. Plot the loss and accuracy for your model every 50 iterations to visualize the training better. You are expected to create 10 independent random 70:30 splits on the given data, train the model and report the average loss and accuracy over all those 10 splits.
2. Try to vectorize your code as much as possible to make your computations faster and efficient. Do not hard code any parts of the implementation unless it is absolutely necessary.

3 What needs to be documented

1. A very brief description of your model and its implementation.
2. The most important feature in the dataset.
3. The final train and test metrics (loss, accuracy, recall, precision and fscore) achieved by your model with GD and SGD
4. Plots of accuracy for three different learning rates using GD and SGD, i.e. three plots for GD with with varying learning rate (say η_1, η_2, η_3) and three plots for SGD with the same set of learning rate on any one data split.

BITS F464 - Machine Learning

Assignment-2B: Artificial Neural Networks

Deadline: ~~17/04/2021, 23:59~~ 19/04/2021, 1300hrs

1 General Instructions

1. This assignment is a coding project and is expected to done in groups. Each group can contain at most **three** members. Make sure that all members in the group are registered to this course and please try to maintain the same group for all the assignments.

2. This assignment is expected to be done in Python using standard libraries like NumPy, Matplotlib and Pandas. You can use Jupyter Notebook and any other python in-built data structure or library. No other ML libraries like scikit/sklearn, TensorFlow, Torch etc. should be used.

3. Refrain from directly copying codes/snippets from other groups or the internet as all codes will be put through a plagiarism check.

4. All deliverable items (`.py` files, `.ipynb` files, report, images) should be put together in a single `.zip` file. Rename this file as `NN_`⟨`id-of-first-member`⟩ (preferably ID number of the student submitting) before submission.

5. Submit the zip file on CMS/GForms on or before the aforementioned deadline. Please note that this is a hard deadline and no extensions or exemptions will be given. The demos for this assignment will be held on a later date which shall be conveyed to you by the IC. All group members are expected to be present during the demo.

6. The link to the dataset for this assignment can be found here.

2 Problem Statement

1. In this assignment, you will have to a implement a simple artificial neural network with at most two hidden layers from scratch that can perform multi-class (1-out-of-K) classification. Choose appropriate non-linear activation (such as sigmoid, tanh, ReLU, LeakyReLU, etc) functions for the hidden and output layers and try to achieve as good accuracy as possible.

2. Use Stocastic Gradient Descent or Mini-Batch Gradient Descent (with appropriate batch size) to train the model. As usual, you can make a random 70:30 split on the given data and use it for training and testing respectively. Plot the loss and accuracy of your model (advisedly in two separate plots, i.e. one for accuracy and one for loss) after every 50 iterations to visualize the training better.

3. Try to vectorize your code as much as possible to make your computations faster and efficient. Do not hard code any parts of the implementation unless it is absolutely necessary.

3 What needs to be documented

1. A very brief description of your model and its implementation.
2. A brief description of your chosen hyper-parameters for the model such as number of hidden layers, number of units per layer, activation functions for each layer and learning rate.
3. The final train and test metrics (loss and accuracy) achieved by your model for ANN with one hidden layer and two hidden layers.
4. Plots of accuracy for three different learning rates for ANN with one hidden layer and two hidden layers.

BITS F464 - Machine Learning

Assignment-2C: Comprehensive Comparison

Deadline: 17/04/2021, 23:59

1 General Instructions

1. This assignment is a coding project and is expected to done in groups. Each group can contain at most **three** members. Make sure that all members in the group are registered to this course and please try to maintain the same group for all the assignments.

2. This assignment is expected to be done in Python using standard libraries like Sklearn, NumPy, Matplotlib and Pandas. You can use Jupyter Notebook and any other python in-built data structure or library. No other ML libraries like TensorFlow, Torch etc. should be used.

3. Refrain from directly copying codes/snippets from other groups or the internet as all codes will be put through a plagiarism check.

4. All deliverable items (`.py` files, `.ipynb` files, report, images) should be put together in a single `.zip` file. Rename this file as `COMP_`⟨`id-of-first-member`⟩ (preferably ID number of the student submitting) before submission.

5. Submit the zip file on CMS/GForms on or before the aforementioned deadline. Please note that this is a hard deadline and no extensions or exemptions will be given. The demos for this assignment will be held on a later date which shall be conveyed to you by the IC. All group members are expected to be present during the demo.

6. The link to the dataset for this assignment can be found here.

2 Problem Statement

1. As for this assignment, you need to do a comparative study and analysis of the ML models you have studied till now, i.e. Fisher Linear Discriminant, Linear Perceptron, Naive Bayes, Logistic Regression, Artificial Neural Networks and Support Vector Machines. Note, only for this assignment you can use the Sklearn library to directly import models/methods and use them.

2. For all the models, use 7-fold cross validation and generate a box-plot over the test set accuracy over each fold. Visualize the all box-plots in a single image. That is, the image must contain six box-plots, one for each model with the box-plot denoting the variation of test set accuracy over each fold.

3. Try to vectorize your code as much as possible to make your computations faster and efficient. Do not hard code any parts of the implementation unless it is absolutely necessary.

3 What needs to be documented

1. A comparative analysis of the models and their accuracies (train and test).
2. The model that performed best and one the that performed worst. (Do mention reasons why that certain model may have given the best or worst results.)
3. The image containing box-plots for each model.
