CSE 474/574: Introduction to Machine Learning (Fall 2021)

Checkpoint Date: September 26, Sunday, 11:59pm

Due Date: October 10, Sunday, 11:59pm

Sargur N. Srihari
University at Buffalo, The State University of New York
Buffalo, New York 14260
Contact: 716-645-6162 (O), srihari@buffalo.edu

September 6, 2021

1 Task

The task of this project is to perform classification using machine learning. It is for a two class problem. Your task is to classify whether a patient has diabetes(class 1) or not (class 0), based on the diagnostic measurements provided in the dataset, using logistic regression and neural network as the classifier. The dataset in use is the Pima Indians Diabetes Database(diabetes.csv). The code should be written in Python.

2 Dataset

Pima Indians Diabetes Database dataset will be used for training, and testing. The dataset contains medical data of female patients above the age of 21 and 768 instances with the diagnostic measurements of 8 features. The 8 features are as follows:

1	Glucose (Blood Glucose level)
2	Pregnancies (The number of pregnancies the patient has had)
3	Blood Pressure(mm Hg)
4	Skin Thickness(Triceps skin fold thickness (mm))
5	Insulin level
6	BMI (Body Mass Index : weight in kg/(height in m)^2)
7	Diabetes Pedigree Function
8	Age (In years)

3 Data processing

1. **Extract features values from the data:** Process the original CSV data files into a Numpy matrix or Pandas Dataframe.

2. **Data Partitioning:** Partition your data into training, validation and testing data. Randomly choose 60% of the data for training, 20% for validation and the rest for testing.

4 Part 1: Implement Logistic Regression[40 points]

Note : For Part 1, you are not allowed to use any Python libraries or built-in functions that directly perform regression. Use of ML libraries like sklearn will result in 0 points for the assignment.

1. **Train using Logistic Regression:** Use Gradient Descent for logistic regression to train the model using a group of hyper-parameters. Use the model to classify whether a patient has diabetes(class 1) or not (class 0).

2. **Test your machine learning model on the testing set:** After finishing all the above steps, fix your hyper-parameters and model parameter and test your model's performance on the testing set. This shows the effectiveness of your model's generalization power gained by learning.

5 Part 2: Implement Neural Networks[40 points]

Note : For Part 2, you are allowed to use any Python libraries or built-in functions.

1. **Train using Neural notworks:** Build a Neural Network with 1,2 or 3 hidden layers with different regularization methods(l2, l1), that takes the features as input and gives as output whether a person has diabetes or not.

2. **Test your machine learning model on the testing set:** After finishing all the above steps, fix your hyper-parameters(learning rate, lambda for penalty and number of neurons per layer) and model parameter and test your model's performance on the testing set. This shows the effectiveness of your model's generalization power gained by learning.

6 Part 3 Bonus Part: Implement different regularization methods for the Neural Networks[10 points]

Note : For Part 3, you are allowed to use any Python libraries or built-in functions.

1. Implement Neural Network with different regularization methods: Dropout, l1 or l2.

2. Compare the Regularization methods and explain it in your report.

7 Evaluation

1. Evaluate your solution on the test set using Accuracy.
2. Plot the graphs showing the train vs validation accuracy and train vs validation loss.

8 Deliverables

There are two deliverables: report and code. After finishing the project, you may be asked to demonstrate it to the TAs, particularly if your results and reasoning in your report are not clear enough.

1. **Report (20 points)**

   The report should be delivered as a separate pdf file, and it is recommended for you to use the NIPS template to structure your report. You may include comments in the Jupyter Notebook, however you will need to duplicate the results in the report. The report should describe your results, experimental setup and comparison between the results obtained from different setting of the algorithm and dataset.

2. **Code (80 points)**

   The code for your implementation should be in Python only. You can submit multiple files, but the name of the Main file should be main.ipynb or main.py. Please provide necessary comments in the code.

9 Checkpoint Submission[Due Date: 26th September]

You only need to submit part 1 for the checkpoint submission to receive feedback and it will not be graded. Your Python code and Report should be packed in a ZIP file named `UBIT name_assignment1_checkpoint.zip(e.g. soumyyak_assignment1_checkpoint.zip)` and upload it to UBlearns in the Assignments section.

10 Final Submission[Due Date: 10th October]

You need to submit both part 1 and part 2 for the Final submission. Your Python code and Report should be packed in a ZIP file named `UBIT name_assignment1_Final.zip(e.g. soumyyak_assignment1_Final.zip)` and upload it to UBlearns in the Assignments section.