Deep Learning Course

Assignment 5

MLP Project

In this assignment, you will be working with two different datasets. Your task is to implement Multilayer Perceptrons (MLPs) using PyTorch to solve a regression problem and a classification problem.

Dataset 1: Student Study Performance

The first dataset is the “Student Study Performance” dataset. This dataset contains information about students’ study habits and their performance in school. You can download the dataset from this link.

Your task is to predict students' performance based on their study habits. This is a regression problem.

Features

There are five different features in this dataset:

- **gender**: sex of students -> (Male/female)
- **race/ethnicity**: ethnicity of students -> (Group A, B, C, D, E)
- **parental level of education** : parents' final education ->(bachelor's degree,some college,master's degree,associate's degree,- high school)
- **lunch**: having lunch before the test (standard or free/reduced)
- **test preparation course**: complete or not complete before the test

There are three target variables your model should predict:

- **math score**
- **reading score**
- **writing score**

Tasks

Here are some steps to guide you:

1. **Data Preprocessing**: Load the dataset and perform any necessary preprocessing steps. This may include handling missing values, encoding categorical variables, etc.

2. **Model Building**: Implement an MLP using PyTorch. Your network should have at least one hidden layer. You can choose the number of neurons in the hidden layer(s).
3. **Training**: Train your model using an appropriate loss function for regression. Monitor the loss function to ensure your model is learning.
4. **Evaluation**: Evaluate your model’s performance on a separate test set.

Dataset 2: Obesity Levels

The second dataset is the “Obesity Levels” dataset. This dataset contains information about individuals’ eating habits and physical condition. You can download the dataset from this link.

Your task is to predict the level of obesity based on the given features. This is a classification problem.

Features

The dataset contains 16 features, and here are them:

- **Gender**: The gender of the individual.
- **Age**: The age of the individual.
- **Height**: The height of the individual in meters.
- **Weight**: The weight of the individual in kilograms.
- **Family History with Overweight**: Whether the individual has a family history of being overweight.
- **FAVC (Frequent Consumption of High-Caloric Food)**: Whether the individual frequently consumes high-caloric food.
- **FCVC (Frequency of Consumption of Vegetables)**: How frequently the individual consumes vegetables.
- **NCP (Number of Main Meals)**: The number of meals the individual has daily.
- **CAEC (Consumption of Food Between Meals)**: How often the individual eats between meals.
- **SMOKE**: Whether the individual smokes.
- **CH2O (Consumption of Water Daily)**: The amount of water the individual consumes daily.
- **SCC (Calories Consumption Monitoring)**: Whether the individual monitors their calorie consumption.
- **FAF (Physical Activity Frequency)**: The frequency of physical activity of the individual.
- **TUE (Time Using Technology Devices)**: The time the individual spends using technology devices.
- **CALC (Consumption of Alcohol)**: The frequency of alcohol consumption by the individual.
- **MTRANS (Transportation Used)**: The mode of transportation the individual uses.

The target variable is **NObeyesdad,** which represents the level of obesity in the individual. It takes seven values: Insufficient Weight, Normal Weight, Overweight Level I, Overweight Level II, Obesity Type I, Obesity Type II, and Obesity Type III.

Tasks

Here are some steps to guide you:

1. **Data Preprocessing**: Load the dataset and perform any necessary preprocessing steps. This may include handling missing values, encoding categorical variables, etc.
2. **Model Building**: Implement an MLP using PyTorch. Your network should have at least one hidden layer. You can choose the number of neurons in the hidden layer(s).
3. **Training**: Train your model using an appropriate loss function for classification. Monitor the loss function to ensure your model is learning.
4. **Evaluation**: Evaluate your model's performance on a separate test set. You can use accuracy, precision, recall, F1-score, etc., as your evaluation metrics.

Please submit your Python notebooks with models and a short report describing your approach and results.