Homework

For Practical work 2

Project 1: Predicting Housing Prices using Linear Regression in PyTorch

Objective:

Build a linear regression model in PyTorch to predict housing prices based on features from a given dataset. This project will help students understand the workflow of data preprocessing, model training, evaluation, and interpretation of results.

Dataset:

Use a housing prices dataset, such as the Boston Housing Prices dataset or Kaggle's House Prices dataset. If you're using a dataset with complex features, students can focus on a subset of relevant features for simplicity.

Steps:

1. Data Loading and Preprocessing

Load the dataset and explore its structure.

Perform basic preprocessing: handle missing values, normalize numerical features, and encode categorical features (if applicable).

Split the data into training and testing sets.

2. Define the Linear Regression Model in PyTorch

Set up a simple linear regression model with PyTorch, using torch.nn.Linear for defining the layer.

3. Training the Model

Define the loss function (Mean Squared Error Loss) and an optimizer (e.g., Stochastic Gradient Descent).

Implement a training loop that:

Feeds inputs into the model,

Calculates loss and performs backpropagation, and Updates weights iteratively over epochs.

4. Evaluating the Model

Calculate metrics such as Mean Absolute Error or Root Mean Squared Error on both the training and test sets to evaluate model performance.

Visualize predictions versus actual prices for a subset of data points.

5. Interpret Results and Experiment

Ask students to experiment with different learning rates, epochs, and features to observe how each factor influences the results.

Encourage them to analyse the limitations of linear regression for this task and suggest potential improvements or alternative models.

6. Reporting

Write a report summarizing their process, findings, and reflections on model performance.

Project 2: Predicting Crop Yield using Machine Learning in PyTorch

Objective:

Build a machine learning model to predict crop yields based on various environmental and soil-related features. This project will introduce students to data preprocessing, feature selection, model building, and evaluation in the context of agricultural data.

Dataset:

Use an accessible crop yield dataset, such as FAO crop production data, Indian crop yield data (available on platforms like Kaggle), or a similar dataset containing information about crop yield, weather conditions, soil properties, and potentially other factors like fertilizer use.

Steps:

1. Data Loading and Exploration

Load the dataset and examine the features (e.g., temperature, rainfall, soil type, fertilizer quantity).

Explore and clean the data: handle missing values, identify outliers, and perform any necessary transformations.

Split the data into training and testing sets.

2. Define the Linear Regression Model in PyTorch

Set up a simple linear regression model with PyTorch, using torch.nn.Linear for defining the layer.

4. Training the Model

Set up a training pipeline with Mean Squared Error or Mean Absolute Error as the loss function, and an optimizer like Adam or Stochastic Gradient Descent.

Implement a training loop with training-validation split, saving model weights periodically, and logging training metrics.

5. Evaluation and Interpretation

Calculate performance metrics, such as Root Mean Squared Error, Mean Absolute Error, or R-Squared, on a hold-out test set.

Plot predictions versus actual yield values to visualize performance and highlight areas for improvement.

7. Reporting and Insights

Write a report discussing the results, including which features were most impactful, the model’s limitations, and how predictions could aid farmers or policymakers.