CIS532/CIS831 DL - Assignment 1

Note

Please note that you are required to use PyTorch to solve the linear regression exercise in this assignment. Start by going through the PyTorch tutorial posted on Canvas. Subsequently, study the code examples for Linear Regression in PyTorch, which are also linked on Canvas (or other similar resources available online). They will be useful in writing your code and answering the questions in this assignment.

Linear Regression Exercise

In this exercise, you will train and evaluate a linear regression model using PyTorch. You will perform feature scaling. You will also analyze learning curves to gain insights into how the performance (i.e., mean square error) varies with the learning rate and the number of iterations used in the gradient descent algorithm.

The linear regression model will be trained and tested on data collected from homes in California and can potentially be used to make predictions about the monetary value of a home.

Dataset

The dataset you will use is available on Canvas (housing.csv). The csv file can be read as a DataFrame using `pandas.read_csv`. The dataset consists of $20,640$ house instances x, and each instance is represented using 9 numeric features, in addition to the target y, which represents the `median_house_value`.

Tasks/Questions

1. Load the housing data and randomly split it into training and test subsets (use 70% of the data for training and 30% for test) - we will not have a validation/development subset in this assignment. You can use the `train_test_split` in sklearn to split the data. Fixing the *random_state* will ensure you get the same splits every time you run your code.

2. Scale the features to ensure that their values are in similar ranges.

3. Create a linear regression model. The model should have one layer (the output layer) with one linear unit, and should take as input instances x representing houses.

4. Plot learning curves that show the variation of performance (i.e., mean squared error) with the number of iterations for both training and test sets for the 2 values of the learning rate (a smaller one and a larger one).

5. Compare the performance on the training data with performance on the test data. Do you see any major differences between the two learning rates you used? How many iterations were needed for convergence?

What to submit

Please submit a Jupiter Notebook containing your code and answers to the questions above. If you use Google Colaboratory, please export your code as a notebooks and upload the files on Canvas, so that we have a timestamp for your submission (links to the Colaboratory notebooks, in addition the Jupyter Notebook files, can also be useful when grading).