Deep Learning HW1

Logistic Regression using Feed Forward Networks

In this assignment, we will explore logistic regression through the lens of deep learning, using a feedforward neural network to implement binary and multiple class classification. Logistic regression, though traditionally a simple linear model, can be extended and enhanced by leveraging the power of neural networks. By mapping input features through multiple layers of neurons and applying non-linear activation functions, feedforward networks enable more complex decision boundaries.

This assignment is divided into two parts. In the first part, you will implement logistic regression using a feedforward neural network for binary classification, where the goal is to predict one of two possible outcomes. In the second part, we will extend this to multi-class classification, allowing the network to predict from more than two classes. Your objective is to experiment with different configurations of the network to improve accuracy, aiming to outperform traditional machine learning algorithms like support vector machines (SVM) and decision trees.

Part 1: (Binary Classification)

Predicting Yearly Income Using a Feed Forward Neural Network:

You are tasked with building a feed-forward neural network to predict whether an individual's yearly income exceeds $50,000 based on a dataset containing personal attributes such as age, gender, nationality, and more.

The data containes 15 features (with the target value), you are given 2 **csv** files for training and testing.
It is **forbidden** to use the test data for training, only for validation, i.e the model shouldn't see the test data during the training process.

First, load the dataset onto your google drive, and mount your drive on the notebook.

Print a couple of samples from the data to study the structure.

Print the distribution of yearly income for the individuals and the percentage of them more / less than 50k/year.

Preprocess the data as you see fit and make it ready for training.

Define your model using **only** pytorch libraries and train it.

Plot the loss, training accuracy and validation accuracy as a function to the number of epochs.

Your job is to try and break the current benchmarks, report your model's number of parameters, layer numbermbers and hyper parameters.

Current benchmarks: Logistic Regression: 79.7% — SVM: 79.8%

You may use any regularization techniques from what we learned in the class and you should choose an optimizer and a loss function suited for this problem.

Hint: recall which loss functions are used for binary and non binary classification.

Part 2: (Multi Class Classification)

Predicting Clothes Categories Using a Feed Forward Neural Network:

In this part we will use the FashionMNIST dataset which is a 70,000 grayscale images of 28x28 pixels dataset, representing 10 categories of clothing items, you can download it using the torch libraries.

Your task is to define 2 feed forward networks:

Network 1: 2 layer network to classify the first 4 categories of the dataset, the network parameters should not pass 50k.

Network 2: 4 layer network to classify all of the categories, the network parameters should not pass 60k

Plot the loss, training accuracy, and validation accuracy as a function of the number of epochs for both networks. Analyze the results, comparing the performance of the two networks. In your explanation, discuss why the first network performs better or worse than the second in terms of accuracy.

You may use any regularization techniques we talked about in class and choose a suitable optimizer and loss function for this particular problem.

After getting your results, write a short summary of the final used network including the hyper parameters and architecture, explain your solution.

Submission:-

Your submission should include a notebook with all code cells already run, good documentation and plots, add text cells above code cells to explain everything you did in that section, if you experimented with different hyper parameters, add those results as well and explain why you got better results with your recent hyper parameters, if not, then explain how different hyper parameters would yield worse results than the current ones.

The name of the submission file should be HW1_ID1_ID2.ipynb where ID1 and ID2 are the IDs of the submitters.

Make sure your notebooks are tidy and easily readable, with the names of the students at the head of the notebook.

Print each model you define and use, with its number of parameters.

Only the pytorch library can be used as a deep learning tool to work with the models, you can’t use keras or tensorflow under any circumstances, you may use other libraries for manipulating the dataset and preprocessing as long as it doesn’t affect the model definition within pytorch.