CS 422/622 Project 3: Linear Regression, Neural Networks

Due 11/19 @ 11:59PM

Logistics: This project includes two files, salary_data.csv and NNPyTorch.ipynb. The csv will be used for the first portion of this project, and the ipynb file will be used for the second portion. The expectation is to implement code for part 1 from scratch, and to answer the accompanying write-ups/discussions without the aid of ChatGPT. Part 2 does not require you to write code, only to try different parameters to tune the neural network. The written portion of the project should not use any generative AI, and is required to be done on your own. Due to only one dataset being provided, you can hardcode the name into your code for easier testing, but please do not provide it as an absolute data path (ex. /home/user/project/exe) as it will need to be changed when running on our end for grading. For part 1, please keep the solution in a single python file called lr.py. Part 2 should have you provide the ipynb file with your adjusted parameters. The write-up can be done in a single file. You are only allowed the numpy, pandas, and matplotlib libraries for this project. It is not required to do a train, test, split for the linear regression, but if you wish to implement it for more a more accurate methodology you can use the sklearn module. No extra credit will be provided for implementing this.

Part 2 of this project uses a ipynb which is the native file format for python notebook files. The expectation is to use this file with Google Colab to access the Google clusters where all the required libraries are already implemented. This service is free with a google account and internet access. You are free to download the required libraries on a local machine to practice PyTorch, but it is much easier through Google Colab. None of the code is loaded onto the GPU so an Nvidia GPU is not required if you wish to install on a local machine.

Deliverables: You should submit a single ZIP file, containing your project code (lr.py, NNPyTorch.ipynb), the data file (salary_.csv) and your writeup (PDF). Grad students are also expected to include their LaTeX source files (*.tex). Your zip file should be named lastname1_project3.zip. For example, a zip file for Sara Smith would be smith_project3.zip. Your code should run without errors on the ECC linux machines. If your code does not run for a particular problem, you will lose 50% on that problem.

1 Linear Regression (50 Points)

File name: **lr.py**

Code Implementation (30 Points)

```
class LinearRegression():
    def __init__(self, learning_rate, iterations):
        self.learning_rate = learning_rate
        self.iterations = iterations

    def fit(self, X, y):
        ...
        for i in range(self.iterations):
            self.update_weights()
        return self

    def update_weights(self):
        ...
        return self

    def predict(self, X):
        ...
        return predicted_value
```

Implement a linear regression algorithm using the class template provided above. The salary_data.csv is provided for testing. It is up to you to define how the algorithm performs the fit, update_weights, and predict functions. Use the Mean Squared Error formula for determining accuracy of the final model. Use gradient descent for updating the weights. If not using train, test, split, then simply use randomly sampled values from the whole dataset to report on the accuracy. If using train, test, split, then use the testing dataset. Make sure to indicate if you used the sklearn module in your report.

Write-Up (20 points)
Give a brief description of the algorithm. Explain what the algorithm's strengths and shortcomings are. How does gradient descent get expanded into use for neural networks?

2 Neural Networks (50 Points)

File name: **NNPyTorch.ipynb**

Code Implementation (20 Points)
A neural network implemented in pytorch is provided to you. There are two sections indicated in the code with comments asking you to provide hyperparameters in order to properly tune the neural network. Your goal is to modify these parameters with the intent of getting an accuracy above 93%. You may get above this, and multiple combinations of parameters will be able to achieve this result. Do not change any of the code outside of the two indicated sections. When you have settled on some parameters that achieve the goal, save the ipynb back to your project folder to upload with the rest of your files. I should be able to simply open your file, run it, and see the finalized results. You can change the loss function if you wish, but it is not required for this project.

Write-Up (30 points)
Describe the neural network algorithm provided in the code to the best of your ability. Indicate what your parameters were, and what they do in the final implementation. For example, describe what the learning rate does. For the optimizer functions, you do not have to write out all the math, but give a brief overview of how the two functions differ from one another in a general sense. Describe at a high level what cross entropy loss does.

Grading: Your code will be ran in the latest Ubuntu environment as provided. If the code fails to compile, it will be a 50% penalty. Your code will be ran as is with the salary_data.csv as provided. The functions provided here in the instructions are simply where the algorithm is intended to be implemented. You are free to implement whatever you need outside of the functions to get the code working.

Grad Students — LaTeX Files:
Grad students are expected to write their report in LaTeX and include the source .tex file with their pdf upload. If you are in need of a resource to write LaTeX files, overleaf.com is an excellent free resource available.