CS422/622- HW 4

Neural Network Implementation

In HW4, you'll build a Neural Network (NN) model to strengthen your practical understanding of machine learning. This assignment will guide you through essential steps such as data preprocessing, model training, parameter tuning, and performance evaluation. Follow the structured guidelines below to ensure a successful implementation.

1. Dataset Selection

- Use a dataset from either HW2 (Iris dataset) or HW3 (Auto MPG) for consistency and comparison purposes.

2. Data Preprocessing

- Apply the same preprocessing techniques used in previous assignments to prepare your data for training. This may include normalization, handling missing values, or feature selection.

3. Model Training

- Construct a neural network architecture suitable for your dataset and problem type (e.g., classification or regression).
- Experiment with different hyperparameters to optimize model performance, including the learning rate, batch size, number of epochs, and layer structure.

4. Testing and Model Evaluation

- Choose an evaluation metric appropriate for your task:
    - For classification: accuracy, precision, recall, f1-score, etc.
    - For regression: RMSE, MAE, R-squared, etc.
- Report the evaluation results and analyze the model's performance.

Report

- **Architecture Details:** Provide a detailed description of your NN architecture, including the type and number of layers, number of nodes per layer, activation functions, loss function, and optimizer used.
- **Comparative Analysis**: Compare the performance and efficiency of your neural network with a previous model (e.g., a logistic regression or decision tree) used in HW2 or HW3. Discuss differences in results, model complexity, and computational requirements.
- **Results and Visualization**: Include code snippets, execution results, and screenshots to clearly demonstrate the process and final outputs of your NN model.

- **Insights and Learning**: Reflect on what you learned through this assignment, including any new insights into neural networks, challenges faced in data preprocessing, model training, or parameter tuning, and potential improvements or different approaches you would consider.
- **Additional Observations**: Share any other observations or interesting findings, such as unexpected results or patterns in the data.
- **There are no restrictions on using external libraries,** but make sure to document the libraries you used and their purposes in the report.

Additional requirements for graduate students in CS 622

Graduate students are expected to:

- Achieve a better performance than the model implemented for HW2 or HW3.

Submission instructions:

You must submit the followings to UNLV WebCampus:

1. A report file
2. Source code file(s)
   - Must be well organized (function name, indentation, …)
   - **You need to upload the python text file (*.py.txt). Simply add ".txt" to the py extension. Don't upload jupyter notebook files**

You must submit the files SEPERATELY. DO NOT compress into a ZIP file. If you fail to provide all required information or files, you may be given zero score without grading.

Once you submit, Webcampus will perform similarity check for your submission and show you the result. Your similarity score must be lower than 50% unless something essential is described in the report. Otherwise, (the score -50%) will be deducted. Detecting any attempts to bypass the similarity check may result in receiving zero points.