ECE5831 Final Project – Summer 2024

Deadline: August 22nd, 2024, by midnight

Instructor: Alireza Mohammadi

Remarks:

(*) Please submit your typed final project in pdf format by sending it to the email address amohmmad@umich.edu. In the very first page of this pdf file, please include the name of all the team members.
(*) A help file associated with the final project is posted to Files section of Canvas. You can download it along with this pdf file from the Modules section.

(*) Usage of LLMs (e.g., ChatGPT) for this final project is permitted if you clearly state how and where you have utilized it in your final project.

(*) You should either use scikit-learn (https://scikit-learn.org/stable/) and your own written routines for implementation of the ML algorithms in this final project. In case you use a package like PyTorch, you must provide a clear explanation of why you have not been able to use scikit-learn. A very good YouTube channel for implementation of ML algorithms from scratch is Patrick Loeber's channel: https://www.youtube.com/playlist?list=PLqnslRFeH2Upcrywf-u2etjdxxkL8nl7E. You can use the codes in this YouTube channel for from-scratch implementations if you clearly state that you are using the codes in this channel. Finally, please note that we have a rich set of Python codes that we went through together during our lectures in the Files section of Canvas. You are encouraged to go through them and utilize them within your project if needed.

(*) You are encouraged to use the available academic resources online to answer some of the questions. For instance, you can find the paper "Balakrishnan, P.V., Cooper, M.C., Jacob, V.S. and Lewis, P.A., 1994. A study of the classification capabilities of neural networks using unsupervised learning: A comparison with K-means clustering. *Psychometrika*, *59*, pp.509-525." by using the search the search query "k-means clustering versus neural networks" in Google Scholar (https://scholar.google.com/).

Objective:
The primary goal of this project is to apply various pattern recognition techniques and neural network models to accurately classify the species of iris flowers using the Iris dataset. You will explore and implement three different methods: K-means clustering, logistic regression, and neural network-based classification. Both from-scratch implementations and library-based (scikit-learn) solutions will be utilized where applicable. There is a huge educational value associated with this project and thorough implementation of the codes by your own self will ensure a very good learning outcome for ECE5831.

Dataset:
The dataset in use, the Iris dataset, comprises 150 samples from three species of Iris: Iris setosa,

Iris virginica, and Iris versicolor. Each sample has four features measured: sepal length, sepal width, petal length, and petal width. For further information, check the Wikipedia entry: https://en.wikipedia.org/wiki/Iris_flower_data_set.

Tasks:

Part 1: Data Exploration and Preprocessing

- **Data Visualization:** Visualize the distribution of the four features for each iris species using histograms and scatter plots to understand data characteristics.
- **Data Preprocessing:** Standardize or normalize the data to prepare for subsequent analysis.
- **Getting a Head Start:** Check the help file.

Part 2: K-means Clustering

- **Implementation:** Develop the K-means clustering algorithm from scratch.
- **Application:** Use your K-means algorithm to cluster the Iris dataset into three clusters.
- **Implementation with scikit-Learn:** Repeat the above step using SciKit-Learn and compare the results with the previous step.
- **Analysis:**
    - Determine the optimal number of clusters by plotting the elbow curve.
    - Compare the clusters formed with the actual species classification using a confusion matrix.
- **Mathematical Foundations:** Explain the update rules for the centroids (geometric centers/clusters) and discuss the criteria for algorithm convergence.

Part 3: Logistic Regression-Based Classification

- **Implementation:** Code logistic regression from scratch to handle multi-class classification in both from-scratch and scikit-Learn implementations.
- **Model Training:** Train your model on the standardized dataset.
- **Evaluation:**
    - Assess the model using metrics such as accuracy, precision, and recall.
    - Employ cross-validation techniques to validate the results. (Note: You might need to do an online search about cross-validation techniques and pick one for achieving the tasks of this part of the project).
- **Mathematical Foundations:** Derive the cost function for logistic regression and illustrate the use of gradient descent for optimization.

Part 4: Neural Network-Based Classification

- **Network Design:** Design a neural network architecture that is effective for classifying the iris species.
- **Implementation:** Implement the neural network using scikit-Learn (https://scikit-learn.org/stable/modules/neural_networks_supervised.html).

- **Training and Testing:**
  - Conduct training sessions and evaluate the model's performance on a held-out test set.
  - Examine the learning curves to gain insights into the training dynamics.
- **Discussion:** Evaluate the impact of various activation functions (e.g., ReLU versus Logistic/Sigmoid), the number of layers, and the neurons per layer on model performance.

Comparative Analysis:

- **Performance Comparison:** After implementing the models, conduct a comparative analysis to evaluate their performance in terms of accuracy and computational efficiency. (Note: You might need to
- **Technique Suitability:** Discuss which methods are best suited for the task based on the results obtained, considering factors such as dataset size, feature characteristics, and model complexity. (Note: It might be the case that all these learning algorithms perform the same for the Iris dataset given its small size.)

Deliverables:

- **Report:** A comprehensive report containing:
  - An introduction to the methods used.
  - A detailed methodology for each part including code snippets and mathematical equation explanations.
  - Results (graphs and tables) with a discussion interpreting the findings from each method.

Assessment Criteria:

- **Code Correctness and Completeness:** How well the code meets the project specifications and its functionality.
- **Mathematical Understanding:** Depth of understanding shown through mathematical derivations and explanations.
- **Report Quality:** Clarity, thoroughness, and professionalism of the written report.