

Lab 02: Softmax Regression Project

1 General Regulations

- The assignment must be completed in registered groups.
- Any member who does not participate will receive no points for this assignment.
- The work must comply with the project requirements.
- All reference materials (if any) must be fully cited in the “References” section of the report.
- Support tools such as ChatGPT, GitHub Copilot, etc., may only be used as references; the content must be reviewed and adjusted to fit the given problem. If it is found that AI tools were used excessively or that the content generated is inappropriate/incorrect, up to 50% of the score may be deducted (depending on the severity).
- Submissions that are identical, plagiarized, or copied from existing works will result in a **zero grade for the entire course**.

2 Project Description

In this project, you will work on a **Classification** problem using the **MNIST** dataset to classify handwritten digits. The main objective is to implement and train a **Softmax Regression** model from scratch, analyze its performance, and build a simple application that can recognize handwritten digits based on your trained model.

This lab emphasizes understanding the mathematical foundation and practical implementation of Softmax Regression. You will also explore different ways of designing feature vectors to improve model performance.

3 Project Requirements

3.1 Model Implementation

- You must **manually implement** the **Softmax Regression** model using only **NumPy** and basic image processing libraries such as **OpenCV** or **PIL**.

- You are **not allowed** to use high-level machine learning frameworks such as TensorFlow, PyTorch, scikit-learn, or similar.
- Clearly derive the mathematical formulation of the Softmax Regression model, including:
 - Model hypothesis
 - Softmax function
 - Cross-entropy loss function
 - Gradient descent update rules
- You must design and compare at least **three different feature vector designs** to improve the model's performance. Examples include:
 - Normalized pixel intensity features
 - Edge or contour-based features using image filters
 - Dimensionality reduction or handcrafted features (e.g., PCA, block averaging, etc.)
- For each feature design, you must:
 - Clearly explain the underlying mechanism and how it transforms the original image into feature vectors.
 - Include **illustrations or diagrams** (e.g., visual examples, block diagrams, or transformation flowcharts) to demonstrate how the features are extracted and represented.
 - Discuss how the feature representation is expected to improve classification accuracy or robustness.

3.2 Model Evaluation and Analysis

- Evaluate your model using appropriate classification metrics such as:
 - Accuracy
 - Precision, Recall, and F1-score
 - Confusion matrix visualization
- Compare the performance of the different feature vector designs.
- Explain and interpret the obtained results in detail, including how the metrics reflect the model's behavior and performance across different feature representations.
- Discuss what your results reveal about the strengths and weaknesses of Softmax Regression on handwritten digit recognition. Your discussion must be strictly based on the observed model outcomes and data patterns — you **must not assume or use any prior domain knowledge or external information** beyond what is derived from your experimental results.

3.3 Application Development

- Develop a simple **handwritten digit recognition application** that allows users to:
 - Draw or upload an image of a handwritten digit.
 - Obtain a prediction from your trained Softmax Regression model.
 - Visualize the prediction probabilities or confidence levels for each digit class.
- The application can be a desktop app, web app, or notebook-based interface.

4 Submission Format

- Each group must designate one representative to submit the assignment.
- The submission must be a compressed file named **Group [Group-ID].zip**, which includes:
 - A folder containing the source code or Github link.
 - An environment file (e.g., **requirements.txt** or **environment.yml**) listing all the libraries and dependencies required to run the project.
 - A video that demo our application.
 - Your final report (PDF format, 10-20 pages recommended) should include:
 - * Dataset description and preprocessing
 - * Mathematical formulation and model implementation
 - * Feature design experiments
 - * Evaluation and analysis of results
 - * Application description and demo
 - * Conclusions and insights
 - * References
- If the dataset is too large, you may upload it to an external server such as Google Drive and submit the public link (which must remain accessible for at least two years). The code and report files must still be submitted on Moodle.
- If the dataset or source code link is inaccessible, 50% of the total score will be deducted.

5 Contact Information

For any questions or issues during the project, please send an email to vntan.work@gmail.com