

AIM-829 Natural Language Processing

Assignment-2

- 1) Implement a MLP from scratch (without any libraries) in Python to solve XOR Logic Gate.
 - 2) Implement a MLP (with libraries) for Sentiment Analysis, i.e., Classification task.
-

General Instructions

- There must be **ONLY ONE** submission made per group but all the group members should contribute towards completing the assignment.
 - Institute policies will apply in cases of **plagiarism**.
 - Create separate .ipynb or .py files for each part. The file name should follow the format: "A2_.ipynb/.py"
 - Create a single .ipynb file to generate the final outputs that are required for submittables. It should be named as "A2__infer.ipynb". Clearly indicate which cell corresponds to the output of which task/subtask. Outputs will be checked from this inference file only by TAs.
 - Carefully read the deliverables for all tasks. Along with the code files, submit all the other files mentioned for each task, strictly following the naming convention instructed.
 - **Only one** person has to submit the .zip file containing all the mentioned files and the report PDF. It will be named "A2_.zip". The person with the alphabetically smallest name should submit it.
 - You are required to submit your trained models. You must also retain all your checkpoints and load and run them during the **demo**.
 - Your report must include the details of each group member's contribution.
-

Dataset Details

- The dataset for task 1 can be downloaded from the given [link](#). Please go through the train.csv and test.csv files and download them separately.
 - The dataset for task 2 can be downloaded from the given [link](#). Please find the train.csv and test.csv files and download them separately.
-

Steps to Complete the Assignment

Step 1: Understand the Dataset

1. Load the training and testing datasets using Python libraries.
 2. Explore the structure of both datasets. Each row represents a word and its corresponding tag, grouped by sentences.
 3. Split the data into words and their respective tags for processing.
-

Step 2: Preprocessing

1. Extract unique words and tokens from the training dataset to build your vocabulary. Follow standard vectorization methods as shown in the course lectures.
-

Step 3: Define your Neural Network Architecture

1. Define several Neural Network Architectures with different hyperparameters and find the best network for your task. Explain in your words your choice of Network architecture and why it performs optimally on the given task.
 2. Choose a loss function for your task and explain in your words your choice of the function.
-

Step 4: Train and Validate Your Model

1. Define a suitable Optimization algorithm for your task. Explain in your own words your choice of Optimization to train the model.
 2. Train the Neural Network with the train.csv data
 3. Define appropriate training metrics.
 4. Plot the training statistics and visualise the process with seaborn or matplotlib.
-

Step 5: Evaluate Your Model

1. Evaluate the model defined with your choice of training metrics on test.csv
 2. Optionally, generate a confusion matrix to analyze the model's performance.
-

Rules and Guidelines

1. All submitted work must be your own. Plagiarism will result in a score of zero for the assignment.
 2. Code generated by AI tools is strictly prohibited.
 3. Submissions that are identified as AI-generated will receive a score of zero.
-

Submission Deadline

17/03/2025, EoD to be submitted in LMS.

Note: Late submissions will not be accepted. If you face any issues, contact TAs before the deadline.