Generative AI, Assignment # 1

Department of Computer Science
National University of Computer and Emerging Sciences,
Islamabad, Pakistan

Due Date: September 21, 2025

Instructions

- Each student must submit the following three files packaged into a single ZIP file and named as **ROLLNO_NAME.ZIP**:
  - A Jupyter Notebook (`.ipynb`) or Python script (`.py`) with the complete implementation.
  - A PDF report written in LaTeX using Overleaf, following the Springer's LNCS paper format:
    Springer LNCS Template on Overleaf.
  - A plain text file (`.txt`) containing all the GPT prompts used for each question.
- Ensure that the code is well-structured with proper comments for each function. Include all necessary dependencies to ensure the code runs without errors.

1 Question 1: Implementing a CNN for CIFAR-10

1.1 Objective

The objective of this task is to build a Convolutional Neural Network (CNN) for image classification using the CIFAR-10 dataset. Students will explore CNN architecture, feature extraction, and model evaluation by implementing and training a CNN model using deep learning frameworks.

1.2 Dataset

For this task, use the CIFAR-10 dataset available at:
Hugging Face - CIFAR-10 Dataset

CIFAR-10 consists of 60,000 images (32x32 pixels, RGB, 10 classes), making it a suitable dataset for CNN training.

1.3 Instructions

1. **Dataset Preparation: Load the CIFAR-10 dataset from Hugging Face and Preprocess the images**
2. **Build a Convolutional Neural Network using a deep learning framework and train the model. Plot training error against each epoch**
3. **Evaluate and Compare Model Performance of the model on the test data. Using confusion matrix show the correct and miss classification then in the format of a table show Accuracy, precision, recall and f-measure**
4. **Extract and visualize feature maps from different convolution layers to analyze the impact of feature extraction. Based on the results, discuss what the convolution layers are doing**
5. **Run the experiments once again to better understand the effect of different hyperparameters on CNN performance, conduct an ablation study by modifying the following four hyperparameters and observing their impact on accuracy. Get the best possible set and then using the optimal values re-run the CNN model and then compare the performnce of a model with and without hyperparmater computation**

    - **Learning Rate**: Experiment with at least three different learning rates (e.g., 0.001, 0.01, 0.1) and analyze how they affect model convergence and accuracy.
    - **Batch Size**: Train the model with different batch sizes (e.g., 16, 32, 64) and compare how it influences training time and performance.
    - **Number of Convolutional Filters**: Vary the number of filters in the convolutional layers (e.g., 16, 32, 64) and observe the effect on feature extraction and accuracy.
    - **Number of Layers**: Modify the number of convolutional layers in the model (e.g., 3, 5, 7 layers) and compare how deeper models perform compared to shallower ones.

Compute the above metrics and present the results in a tabular format:

Table 1: Performance Metrics Comparison of CNN Models

Model	Accuracy	Precision	Recall	F1-Score
RNN	[Value]	[Value]	[Value]	[Value]

2 Question 2: Implementing a RNN for Next-Word Prediction

2.1 Objective

The goal of this task is to train a Recurrent Neural Network (RNN) on the Shakespeare text dataset from Hugging Face for next-word prediction. Instead of using pre-trained embeddings such as Word2Vec or GloVe, students will train their own word embeddings using an Embedding Layer in TensorFlow or PyTorch.

2.2 Dataset

Use the publicly available Shakespeare text dataset from Hugging Face: `https://huggingface.co/datasets/karpathy/tiny_shakespeare`

2.3 Tasks

1. Load and preprocess the dataset.
2. Implement a suitable RNN model (LSTM or GRU may be used).
3. Train the model and monitor performance. Plot the training and validation curves.
4. Generate text predictions: provide a seed phrase (e.g., "To be or not to"). The model should iteratively generate at least 10 words to form a complete sentence.
5. Evaluate model performance using appropriate metrics (e.g., perplexity, accuracy).
6. Perform an ablation study by modifying one component of the model (e.g., hidden size, number of layers, dropout) and compare results.

3 Question 3: Implementing PixelCNN, Row LSTM, and Diagonal BiLSTM (PixelRNN)

3.1 Objective

The goal of this task is to implement and compare three generative models for image data based on the Pixel Recurrent Neural Networks (PixelRNN) paper

by van den Oord et al. (2016). Students are required to carefully read the paper and reproduce the three main architectures proposed: PixelCNN, Row LSTM, and Diagonal BiLSTM.

3.2 Reference Paper

`https://arxiv.org/abs/1601.06759`

3.3 Dataset

Use the CIFAR-10 dataset for training and evaluation. Dataset link: `https://www.cs.toronto.edu/~kriz/cifar.html`

3.4 Tasks

1. Read and understand the PixelRNN paper, focusing on Sections 2 and 3 where the model architectures are described.
2. Implement the following three models:
   - PixelCNN using masked convolutions (mask type A for the first layer; type B for later layers).
   - Row LSTM: a recurrent layer scanning rows with input-to-state and state-to-state convolutions as described in the paper.
   - Diagonal BiLSTM: implement skewing of the input map, apply bidirectional LSTMs along diagonals, then unskew the output and combine directions properly.
3. Train all three models on the CIFAR-10 dataset using a discrete softmax over pixel values as the output distribution.
4. Monitor and plot training and validation performance using negative log-likelihood (bits/dimension) as the evaluation metric.
5. Compare the models using the evaluation metrics reported in the original paper (e.g., negative log-likelihood) or other suitable metrics for generative image modeling.
6. Compare the models using the evaluation metrics reported in the original paper (e.g., negative log-likelihood).
7. (Bonus) In addition, you may use other suitable metrics for generative image modeling, such as Inception Score (IS), Fréchet Inception Distance (FID), or qualitative visual inspection of generated samples.