**ADL PS1 (Feature extraction via dimensionality reduction using variants of autoencoders)**

This Project is about feature extraction via dimensionality reduction using variants of autoencoders.  Use the CIFAR10 dataset provided in Keras, after conversion to gray-level images! Use randomly selected 70% of the dataset as training set and remaining 30% as the test set.

Task 1: Perform standard PCA with 70% of the training dataset and identify the eigenvectors associated with top eigenvalues with 95% total energy. With these, train a logistic regression classifier to classify the images into 10 classes. Draw the ROC curve for the test dataset. Repeat the same with randomized PCA and compare.

Task 2: Train a single layer autoencoder with linear activation function and appropriately mean and variance normalized input with constraint that encoder weight matrix and decoder weight matrix are transpose w,r,t, each other. Compare the eigenvectors obtained in step 1 with those obtained using the autoencoders. Explain your observations.

Task 3: Train an appropriate deep convolutional autoencoder with same dimension of latent space. Calculate the reconstruction error fand compare that with a single hidden layer autoencoder (with sigmoid activation at the autoencoder and linear at the decoder) for the test dataset. What will be the reconstruction error if the hidden nodes are distributed equally (approximately) among 3 hidden layers in a new 3 hidden layer autoencoder with sigmoid activation at the autoencoder and linear at the decoder final layer?

Task 4. Train a deep convolutional autoencoder with MNIST dataset and using extracted features train a MLP classifier with 7 outputs (7 segment LED display) that are representative of 10 digits

**ADL-2 (Implementation of Wesserstein GAN, SNGAN with CIFAR10)**

Task 1: Implement and train a conditional Wesserstein GAN with CIFAR10 dataset. Generate and display 10 new images belonging to 'automobile' class. Report IS and FID scores.

Task 2: Implement and train a SNGAN with CIFAR10 dataset. Generate and display 10 new images. Report IS and FID scores.

Task 3: Implement and train a variant of SAGAN **without** spectral normalization and TTUB  with CIFAR10 dataset. Generate 10 new images. Report IS and FID scores.

Task 4: Implement and train a complete SAGAN with CIFAR10 dataset. Generate and display 10 new images. Report IS and FID scores.

**GNN-1 (Amazon product co-purchasing network)**

The ogbn-products dataset is an undirected and unweighted graph, representing an Amazon product co-purchasing network [1]. Nodes represent products sold in Amazon, and edges between two products indicate that the products are purchased together. We follow [2] to process node features and target categories. Specifically, node features are generated by extracting bag-of-words features from the product descriptions followed by a Principal Component Analysis to reduce the dimension to 100.

GNN-2

1. Problem Statement:
   1. Predict the label of graph based on model designed as per below details.
   2. Generate Graph embedding using Research Paper Anonymous Walk Embeddings Sergey Ivanov12 Evgeny Burnaev1 URL: [Anonymous Walk Embeddings Sergey Ivanov12 Evgeny Burnaev1 URLLinks to an external site.](https://arxiv.org/abs/1805.11921)
   3. Use Suitable neural network to predict the label.
   4. Optimize entire model pipeline for prediction.
   5. Dataset :ogbg-molhiv from <https://ogb.stanford.edu/docs/graphprop/#ogbg-mol>

**CAI (Retrieval-Augmented Generation (RAG) model)**

Develop a **Retrieval-Augmented Generation (RAG)** model to answer financial questions based on company financial statements (last two years).

**IMPORTANT**

* **Use Only Open-Source Embedding Models**
* **Use a Small Open-Source Language Model (SLM) for Response Generation** (No proprietary APIs)
* **Implement One Guardrail** (Either Input-side or Output-side)
* **Develop an Application Interface** (Web-based, CLI, or GUI)
* Each Group Implements a Specific Advanced RAG Technique based on the group number

**Advanced RAG Techniques**

1. Multi-Stage Retrieval
2. Chunk Merging & Adaptive Retrieval
3. Re-Ranking with Cross-Encoders
4. Hybrid Search (Sparse + Dense Retrieval)
5. Memory-Augmented Retrieval

**Tasks**

| **Component** | **Details** |
| --- | --- |
| 1. Data Collection & Preprocessing | Download the last two years of financials (Use any one group member's company earning statements, if nothing is available use any company's data freely available). Clean and structure the data for retrieval. |
| 2. Basic RAG Implementation | Implement a simple RAG model: - Convert financial documents into text chunks. - Embed using a pre-trained model - Store and retrieve using a basic vector database |
| 3. Advanced RAG Implementation | Improve retrieval by: - Using BM25 for keyword-based search alongside embeddings. - Testing different chunk sizes & retrieval methods for better accuracy. - Implementing re-ranking. |
| 4. UI Development (e.g., Streamlit) | Build an interactive UI: - Accept user queries. - Display answer & confidence score. - Ensure clear formatting & responsiveness. |
| 5. Guard Rail Implementation | Implement one guardrail: - Input-Side: Validate and filter user queries to prevent irrelevant/harmful inputs. - Output-Side: Filter responses to remove hallucinated or misleading answers. |
| 6. Testing & Validation | Ask 3 test questions: - A relevant financial question (high-confidence). - A relevant financial question (low-confidence). - An irrelevant question (e.g., "What is the capital of France?") to check system robustness. |

DNN-1

* **Implement the Lab code:**Execute the code in Lab sheets as uploaded in the module section. The implementation should follow the instructions provided in the lab sheets.
* **Code Submission:** Upload the Python .ipynb file. Download the .ipynb file as a PDF, ensuring all outputs are clearly displayed. ZIP files are not accepted.
* **File Naming Convention:** DNN\_assignment\_2b\_group##. (Fllow the same Template as given in "DNN\_Assignment2a\_Template.ipynb" and change based on the networks )
* **Plagiarism & Late Submissions:** Any plagiarism will result in zero marks. Late submissions incur a penalty of (-2) marks.

**Additional Instructions:**

* Data need not be uploaded with the submission.
* Submit the updated Jupyter Notebook with outputs + the final .ipynb notebook file converted as PDF, with proper formatting and alignment.
* Incomplete output, misalignment, or lack of comments may result in mark deductions.
* If the given template is not followed, ZERO marks will be awarded.
* Journals can be chosen without any restrictions on impact factors or other indices.
* If the dataset URL is not provided in the research papers, utilize datasets from publicly accessible resources.

NLP1

**Word Embeddings and text Classification**

 The goal of this assignment is to train a simple feed foreword neural network for text classification and get a trained word embedding vector from the network.

**DRL-1**

Multi-Arm Bandit Problem **(5 Marks)**: Imagine an innovative digital advertising agency, AdMasters Inc., that specializes in maximizing click-through rates (CTR) for their clients' advertisements. One of their clients has identified four key tunable elements in their ads: Age, City, Gender, and Mobile Operating System (OS). These elements significantly influence user engagement and conversion rates.

DRL-2 ENERGY Consumption optimization

Assignment 2: Energy Consumption Optimization (15Marks)  
Instructions:  
● Read the assignment proposal carefully.  
● If any of the requirements are missed in the inal code submission, the respective  
marks will be deducted.  
● It is mandatory to submit the assignment in the PDF format only consisting of all  
the outcomes with each and every iteration. Any other format will not be  
accepted.  
● Add comments and description to every function you are creating or operation  
you are performing. If not found, then 1 mark will be deducted. There are many  
assignments that need to be evaluated. By providing the comments and  
description it will help the evaluator to understand your code quickly and clearly.  
● Maintain the same naming conventions for the PDF iles to be submitted as that  
of ipynb iles.  
● Submit 2 different PDFs. One forActor-Critic and One for DQN & DDQN.  
● Late submissions will lead to a deduction of 1 Mark.  
Problem Statement: The objective of the problem is to implement an Actor-Critic  
reinforcement learning algorithm to optimize energy consumption in a building. The  
agent should learn to adjust the temperature settings dynamically to minimize energy  
usage while maintaining comfortable indoor conditions.  
Dataset: https://archive.ics.uci.edu/dataset/374/appliances+energy+prediction  
This dataset contains energy consumption data for a residential building, along  
with various environmental and operational factors.  
Data Dictionary:  
o Appliances: Energy use in Wh  
o lights: Energy use of light ixtures in the house in Wh  
o T1 - T9: Temperatures in various rooms and outside  
o RH\_1 - RH\_9: Humidity measurements in various rooms and outside  
o Visibility: Visibility in km  
o Tdewpoint: Dew point temperature  
o Press\_mm\_hg: Pressure in mm Hg  
o Windspeed: Wind speed in m/s  
State Space:  
The state space consists of various features from the dataset that impact energy  
consumption and comfort levels.  
• Current Temperature (T1 to T9): Temperatures in various rooms and  
outside.  
• Current Humidity (RH\_1 to RH\_9): Humidity measurements in different

ML

Dream Housing Finance company deals in all home loans. They have a presence across all  
urban, semi-urban and rural areas. The customer first applies for a home loan, after that  
company validates the customer's eligibility for a loan. The company wants to automate the  
loan eligibility process (real-time) based on customer detail provided while filling out the online  
application form. These details are Gender, Marital Status, Education, Number of Dependents,  
Income, Loan Amount, Credit History and others.