Assignment 2

UMC 203: Artificial Intelligence and Machine Learning

March 2025

No copying is allowed. **Thorough plagiarism check will be run.** Refer: CSA misconduct policy You are given three questions, each of which requires Python programming. You may use jupyter notebook to write your Python programs, although not necessary. **If you are using a notebook, please convert it to a Python file before you submit it.** We expect you to implement your own programs. Code generated by non-humans will be non-acceptable ;). Shenanigans will beget Shenanigans.

SUBMISSION INSTRUCTIONS

1. You should submit **four files** (NOT a zip file) with the following naming convention.
   - ▷ `AIML_2025_A2_LastFiveDigitsOfSRNumber.pdf` → Answers to all the problems.
   - ▷ `AIML_2025_A2_LastFiveDigitsOfSRNumber.py` → Code for the all the problems.
   - ▷ `w_ols_LastFiveDigitsOfSRNumber.csv`, `w_rr_LastFiveDigitsOfSRNumber.csv` → Solution for Question 3.1.

   For example, if the last five digits of your SR Number is 20000, then you should submit four files: `AIML_2025_A2_20000.pdf, AIML_2025_A2_20000.py, w_ols_20000.csv, w_rr_20000.csv`.
2. **Any deviation from the above rule will incur serious penalty!**
3. For the coding questions, you are asked to report some values, e.g., the number of iterations. These values should be reported in the `.pdf` file you submit.
4. At the top of the `.pdf` file you submit, write your name and SR Number.
5. Reports must be typed neatly in LaTeX
6. You will be using the `torch`, `numpy`, `pandas` and `sklearn` libraries for this assignment.

Additional Instructions

- You are encouraged to use the GPU-laden machines in the UGC Lab.
- Details for using the oracles are present along each question.
- We recommend caching oracle outputs locally after the first call to speed up your programs.
- You will need to be connected to the IISc VPN to be able to use the oracle.
- Please install the `requests` library in your environment using `pip install requests` for the oracle to work correctly.
- **THE ORACLE SERVER WILL BE CLOSED 24 HOURS BEFORE THE SUBMISSION DEADLINE!**

1 Support Vector Machine and Perceptron (10 marks)

For this question, you are given with CIFAR-100 dataset. You will validate its separability by testing the convergence of the perceptron, before isolating (if they exist) sources of non-separability through the support vector machine.

Tasks

1. **Perceptron**: Run the perceptron algorithm on your data. Report whether it converges, or appears not to. If it doesn't seem to converge, make certain that you are reasonably sure.
2. **Linear SVM**: Construct a *slack* support vector machine with the linear kernel. Solve both the primal and dual versions of the SVM quadratic programs **using cvxopt**. Use the SVM's solution to isolate the sources of non-separability.
3. **Kernelized SVM**: Repeat the previous construction, but with the Gaussian kernel this time. Choose your hyperparameters such that non-separability is no longer an issue, and your decision boundary is consistent with the training data's labels.
4. **Perceptron, again**: Retrain the perceptron, after removing the sources of non-separability isolated by the linear SVM. Verify that it converges.

Deliverables

1. A plot between misclassification rate and number of iterations for the perceptron algorithm as defined in Task 1. **(2 marks)**
2. Which, between the primal and dual, is solved faster for Task 2. Report the times taken for running both, and justify any patterns you see. **(2 marks)**
3. The images that cause non-separability. **(2 marks)**
4. The final misclassification rate for the kernelized SVM for Task 3. **(2 marks)**
5. A plot between misclassification rate and iterations for the perceptron for Task 4. **(2 marks)**

Details of the Oracle functions:

- `q1_get_cifar100_train_test(srn)`: Takes a 5 digit integer, which is your SR number as input. **Please use your SR number**. The output is a 2-tuple: `(train_data,test_data)` where each data is a list of `['features','labels']` ,i.e `for x in train_data, x[0]=features & x[1]=labels`. The features and labels are numpy array of shape `(N,432)` and `(N,)` respectively.

2 Logistic Regression, MLP, CNN & PCA (10 marks)

For this question, you are given with a subset (of size 10 x 1,000 specific to your 5 digit serial number) of MNIST-JPG dataset consisting of 28x28 grayscale images (total-70,000) with ".jpg" extension. You will design and analyse the classification (10 classes) models using neural networks along with some classical techniques and submit a comparison report.

Tasks

1. **Multi Layer Perceptron**: Construct and train an MLP that takes flattened image as input (for example if image size is 28x28 then input dimension of mlp should be 784) and gives class probabilities as output. **(1 marks)**
2. **Convolution Neural Network**: Construct and train a CNN that takes direct image as input and gives class probabilities as output. **(1 marks)**
3. **Pricipal Component Analysis**: Extract image features using PCA (which is a famous dimensionality reduction technique). **(1 marks)**

4. **MLP with PCA**: Repeat Task-1, but now use handcrafted features extracted using PCA as input and class probabilities as output. **(1 marks)**

5. **Logistic Regression with PCA**: Train a Logistic Regression model for multi class classification and also train 10 binary classifiers for each class by one vs rest approach using PCA features. **(1 marks)**

Deliverables

1. Reconstruct an image of your choice using principal components (1,2,3,....) and conclude the results. **(1 marks)**

2. Print confusion matrix for all the multi class classification models and compare them using following metrics : Accuracy, precision, recall and F1 score for each class and compare the results. **(2 marks)**

3. Compute average AUC score using ROC curves for each class obtained from 10 binary classifiers (logistic regression) trained in Task-5. **(2 marks)**

Details of the Oracle functions:

- `q2_get_mnist_jpg_subset(srn))`: Takes a 5 digit integer, which is your SR number as input. **Please use your SR number**. The output is a folder consisting of 10 sub folders named as 0,1,2,...,9 (representing class), each consists of 1000 images in ".jpg" format. By providing your serial number to the oracle you can get a sample dataset of size 10,000 (10 x 1,000) specific to you, organised as expected by PyTorch's ImageFolder() data loader.

3 Regression (10 marks)

3.1 Linear Regression

In this section, you will solve a regression problem using two approaches:

- **Ordinary Least Squares (OLS):**
  Minimize the objective function:

$$J(w) = \frac{1}{2n}\sum_{i=1}^{m}(y^{(i)} - w^T x^{(i)})^2$$
$$= \frac{1}{2n}||Y - Xw||^2, \quad (X \in \mathbb{R}^{n\times d}, Y \in \mathbb{R}^n)$$

  where $x^{(i)} \in \mathbb{R}^d$ are your features and $y^{(i)} \in \mathbb{R}$ are your corresponding labels.

- **Ridge Regression (RR):**
  Minimize the objective function:

$$J(w) = \frac{1}{2n}||Y - Xw||^2 + \frac{\lambda}{2}||w||^2, \quad (X \in \mathbb{R}^{n\times d}, Y \in \mathbb{R}^n)$$

  where $x^{(i)} \in \mathbb{R}^d$ are your features and $y^{(i)} \in \mathbb{R}$ are your corresponding labels.

Tasks

1. Query the oracle to obtain $\mathcal{D}_1^{\text{train}}$, $\mathcal{D}_2^{\text{train}}$, $\mathcal{D}_1^{\text{test}}$, and $\mathcal{D}_2^{\text{test}}$.

2. Solve the linear regression and ridge regression with $\lambda = 1$ on $\mathcal{D}_1^{\text{train}}$ and $\mathcal{D}_2^{\text{train}}$ to obtain $w_1^{\text{ols}}, w_1^{\text{rr}}$, w_2^{ols} and w_2^{rr}. That is, w_1^{ols} means optimal weight vector corresponding to ordinary least square using $\mathcal{D}_1^{\text{train}}$ and w_1^{rr} means optimal weight vector corresponding to ridge regression using $\mathcal{D}_2^{\text{train}}$.

3. Calculate MSE for $w_1^{\text{ols}}, w_1^{\text{rr}}$ using $\mathcal{D}_1^{\text{train}}$ and $w_2^{\text{ols}}, w_2^{\text{rr}}$ using $\mathcal{D}_2^{\text{train}}$.

Deliverables:

1. Can you do OLS if $\mathbf{X}$ does not have full column rank? **(0.5 marks)**
2. Report MSE on $\mathcal{D}_1^{\text{train}}$. Report w_1^{ols} and w_1^{rr} in the pdf. **(1 mark)**
3. Report MSE on $\mathcal{D}_2^{\text{train}}$. Attach w_2^{ols} and w_2^{rr} as csv files named `w_ols_[five-digit-srnumber].csv` and `w_rr_[five-digit-srnumber].csv` **(1 mark)**
   - To save a numpy array `w` to a csv, use `numpy.savetxt("<filename>.csv", w, delimiter = ",")`. For example, if your SR Number is 2000, use `numpy.savetxt("w_ols_20000.csv", w_ols, delimiter = ",")`.

Details of the Oracle functions:

- `q3_linear_1(srn)`: Takes a 5 digit integer, which is your SR number as input. **Please use your SR number**. The output is a 4-tuple: `(X_train,y_train,X_test,y_test)`. The array `X_train` is of size $(N \times 5)$ and `y` is of shape $N \times 1$.
- `q3_linear_2(srn)`: Takes a 5 digit integer, which is your SR number as input. **Please use your SR number**. The output is a 4-tuple: `(X_train,y_train,X_test,y_test)`. The array `X` is of shape $(N \times 100)$ and `y` is of shape $N \times 1$.

3.2 Support Vector Regression

In this problem, you will apply Support Vector Regression (SVR) to predict stock prices using historical stock market data.

1. Query the oracle to find the stock assigned to you.
2. Download the Stock-net dataset. Find the appropriate CSV file in `stocknet-dataset/price/raw/`.
3. Extract the closing prices and normalize them using the formula $x' = \frac{x-\mu}{\sigma}$. You may use scikit learn's `StandardScaler`. You now have a $N \times 1$ vector $\mathbf{d}$.
4. Choose the time window t. Create a $(N-t) \times t$ matrix where the i^{th} row contains the closing prices of days i through $(i+t-1)$. This is your data $\mathbf{X}$.
5. Obtain the labels $\mathbf{y}$ by removing the first t elements of $\mathbf{d}$.
6. You will use the first half of the dataset to fit the data and the second half of the dataset as a test set.

Let $\mathbf{X}_i$ represent row i of $\mathbf{X}$. You will use SVR to fit a function $f(x) : \mathbb{R}^t \to \mathbb{R}$ such that $f(\mathbf{X}_i) = y_i$.
In **SVR**, instead of minimizing the mean squared error, we use an $\epsilon-$insensitive loss function to ignore small deviations and focus on significant errors:

$$E_\epsilon(f(x) - y) = \begin{cases} 0 & \text{if } |f(x) - y| < \epsilon, \\ |f(x) - y| - \epsilon & \text{otherwise.} \end{cases}$$

We therefore minimize a regularized error function given by,

$$\min_{w,b} \quad C \sum_{n=1}^{N} E_\epsilon(f(x_n) - y_n) + \frac{1}{2}||w||^2$$

where $f(x) = w^T \phi(x) + b$

and $\phi(x)$ represents a fixed feature space transformation

Tasks

Train the following SVRs using the train set for $t \in \{7, 30, 90\}$:

1. Solve the dual of the slack linear support vector regression using `cvxopt`.
2. Solve the dual of the kernelized support vector regression using the RBF kernel for $\gamma = [1, 0.1, 0.01, 0.001]$ using `cvxopt`.

Deliverables

For each SVR trained, plot a graph on the test set containing the following: **(0.5 x 15 = 7.5 marks)**

1. Predicted closing price value.
2. Actual closing price value.
3. Average price on the previous t days.

Details of the Oracle functions:

- `q3_stocknet(srn)`: Takes a 5 digit integer, which is your SR number as input. **Please use your SR number**. The output is the stock ticker symbol of the stock you are supposed to use.