INT 304: Pattern Recognition in Computer Vision
Final Assignment

Assessment Number	Final Coursework
Contribution to Overall Marks	70%
Start Date	21-April-2025
Submission Deadline	18-May-2025 (23:59, Beijing Time)

General Guidelines

- Your report should introduce each model, detailing hyper-parameters, structure, and algorithm interpretation. Include mathematical equations with explanations, clipped running images of the training process, and a concise analysis of results and observations, offering a complete understanding of each model's performance.
- The report should be written in English.
- Submit the report and the Python source code electronically into LearningMall. Report should be added the "SUBMISSION COVER SHEET" as the first page. All students must download their file and check that it is viewable after submission. Students themselves are responsible for submitting a functional and correct file for assessments.
- The **report in pdf format** and the **source code** of your implementation should be zipped into a single file. The naming of the report is as follows:

  INT304_StudentID_LastName_FirstName_FinalCW.zip,

  e.g. (INT304_123456789_Einstein_Albert_ FinalCW.zip)
- According to Xi'an Jiaotong-Liverpool University regulations, you will be penalised for late or non-submission. The use of Generative AI for content generation is not permitted on this coursework. You are reminded of the need to comply with Xi'an Jiaotong-Liverpool University's guidelines to Academic Integrity.
- 5% of the total marks available for the assessment shall be deducted from the assessment mark for each working day after the submission date, up to a maximum of five working days.

Assessment Objective

This final assessment aims at evaluating students' ability to exploit the pattern recognition knowledge, which is accumulated during lectures, and after-class study, to analyse, design, implement, develop, test, and document the pattern classification methods with Neural Networks (MLP) and Kernel Methods (Support Vector Machines) typically on the image data.

Feedback

The comments for this assignment will be available at the end of the exam week. The final grade will be available after the exam board at the end of the semester

Overall Description:

The assessment focuses on comparing Three classic classifiers, i.e. Neural Networks and Kernel Support Vector Machine (RBF and Polynomial kernels), on the image classification dataset. You are required to design and optimise the models to achieve accurate classification performance on the specified dataset.

The assessment includes the programming codes and the report. **The programming codes should be implemented correctly, run successfully, and yield sensible results.** Moreover, the code quality will also be considered, such as efficiency, comments, and robustness.
Provide a comprehensive report summarizing your findings. The report should include: Description of data preprocessing steps, description and implementation details of the RBM, MLP and custom classifiers. You are also required to follow the general guidelines and answer all the questions in the report clearly.

Dataset (LFW):

Labeled Faces in the Wild (LFW) is a database of face photographs designed for studying the problem of unconstrained face recognition. The data set contains more than 13,000 images of faces collected from the web. Each face has been labeled with the name of the person pictured. 1680 of the people pictured have two or more distinct photos in the data set. The example data is illustrated in Fig. 1. *Source: https://vis-www.cs.umass.edu/lfw/*

Download the Labeled Faces in the Wild Home dataset from module web-page on LearningMall.
Or you can use the following python codes for convenience:

```
from sklearn.datasets import fetch_lfw_people
lfw_people = fetch_lfw_people(min_faces_per_person = , resize = )
```

Fig. 1. Example of LFW dataset.

Hints: Load and preprocess the LFW dataset to make it suitable for training with RBM. Normalize or standardize the data if necessary.

Tasks: (70 marks in total)

Implementing and comparing **three** classifiers on LFW Dataset. You are required to design and optimise the models to achieve accurate classification performance.

1 Implement Multilayer Perceptron (MLP) on LFW Dataset (35 marks)

An artificial neural network is composed of many artificial neurons that are linked together according to a specific network architecture. The model transforms the inputs into meaningful outputs by iterating a combination of linear and nonlinear functions.

In the 1st layer, weightless connections pass inputs to the hidden layer. The input feature vector is augmented with the 1 since

$$w^T x + w_0 = [w^T \;\; w_0] \begin{bmatrix} x \\ 1 \end{bmatrix}, \tag{1.1}$$

The label l_n of the n-th example is converted to a L dimensional vector n as follows (one-hot labels)

$$t_{nk} = \begin{cases} +1, & k = l, \\ 0, & k \neq l, \end{cases} \tag{1.2}$$

(a) Initialize all weight w_{ij} of MLP network, for example $w_{ij} \in \left[\!\left[-\sqrt{\frac{6}{D+1+L}}, \sqrt{\frac{6}{D+1+L}} \right]\!\right]$, where D and L is the number of the input nodes and the output nodes, respectively (allow different initialization methods).

(b) Choose randomly an input vector x to network and forward propagate through the network

$$a_j = \sum_{i=1}^{d} w_{ij}^{(1)} x_i \,, \qquad z_j = g(a_j), \qquad y_k = \sum_{j=0}^{M} w_{kj}^{(2)} z_j \,, \tag{1.3}$$

where $g(\cdot)$ denote an Activation Function (e.g., $g(a_j) = \frac{1}{1+e^{-a_j}}$, allow different activation functions).

(c) The error rate is $E = \frac{1}{2}\sum_{l=1}^{L}(y_l - t_l)^2$ for the example x.

(d) Evaluate the δ_k for all output units $\delta_k = y_k - t_k$.

(e) Backpropagate the δ's to obtain δ_k for each hidden unit in the network

$$\delta_j = g(a_j)' \sum_k w_{kj}\delta_k = \left(1 - z_j^2\right) \sum_k w_{kj}\delta_k . \quad (1.4)$$

(f) The derivative with respect to the first-layer and the second-layer weights are given by

$$\frac{\partial E_n}{\partial w_{ji}^{(1)}} = \delta_j x_i , \quad \frac{\partial E_n}{\partial w_{kj}^{(2)}} = \delta_k z_j . \quad (1.5)$$

Algorithm 1 Stochastic Backpropagation Algorithm

```
1: Initialize w, η
2: for t = 1 to T do
3:   Shuffle the training data set randomly.
4:   for n = 1 to N do
5:     Choose the pattern x_n
6:     Forward the input x_n through the network
7:     Backward the gradient from the output layer through network to obtain ∂E_n/∂w_ji^(1) and ∂E_n/∂w_kj^(2)
8:     Update the weights of the network
```

$$w_j k = w_{jk} - \eta \frac{\partial E_n}{\partial w_{kj}^{(2)}}, \quad w_{ij} = w_{ij} - \eta \frac{\partial E_n}{\partial w_{ji}^{(1)}}$$

```
9:   end for
10: end for
11: return w
```

(g) The framework of MLP algorithm is as follows, where $\eta = 0.001$ and $M = 20$ is the number of hidden nodes. Note that M, η, T are the hyperparameters of the network. Try to tune these hyperparameters to obtain a good performance.

(h) The algorithm may be terminated by setting the total iteration T except that setting the threshold θ of the gradient referred in the lecture slide.

(i) In the test stage, the test example x is forwarded into the network to obtain the output $y_{L\times 1}$ and then assigned to the label with the maximum output value.

Download the refernce python code from module web-page on LearningMall: Final Coursework-->Reference Code-->INT304_SLP_and_MLP.ipynb

(1) **Adjust and modify the codes to implement a Multilayer Perceptron (MLP) classifier to perform the classification task on Facial Data.** This includes process the data, parameter initialization, implementation of the forward and backward propagation *(7 points).*

(2) **Train and tune the MLP:** Design a training strategy that includes the design of the loss function, regularization term, and other components. Additionally, optimize the learning rate alongside other hyper-parameters for effective tuning and performance enhancement *(12 points).*

(3) **Results evaluation and visualization:** focusing on performance metrics and graphical representations to assess the model's effectiveness and insights. Provide an analysis of the results, highlighting the model's strengths, weaknesses, and potential areas for enhancement *(16 points).*

2 Implement Kernel Support Vector Machines (SVM) on LFW Dataset (35 marks)

A Support Vector Machine constructs a hyperplane or set of hyperplanes in an infinite-dimensional space, which can be used for classification, regression, or other tasks.

Support Vector Machine implements a very simple idea – mapping pattern vectors to a high-dimensional feature space where a 'best' separating hyperplane (the maximal margin hyperplane) is constructed.

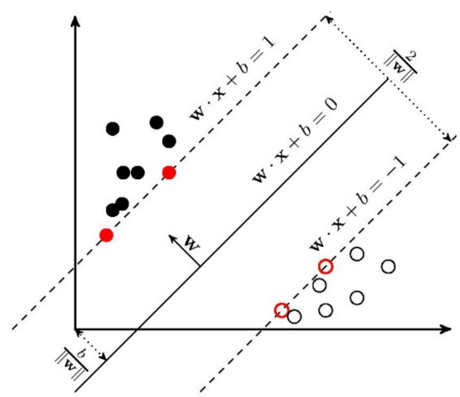

Fig 2: SVM demonstration for binary classification, showing the decision boundary and margin-maximizing support vectors.

For binary classification problem, our goal is now to maximize the margin while softly penalizing points that lie on the wrong side of the margin boundary (as shown in Fig. 2). We therefore minimize

$$\frac{1}{2}\|w\|^2 + C\sum_{n=1}^{N}\xi_n , \qquad (2.1)$$

where the parameter $C > 0$ controls the trade-off between the slack variable penalty and the margin. Because any point that is misclassified has $\xi_n > 1$, it follows that $\sum_n \xi_n$ is an upper bound on the number of misclassified points.

(a) We now wish to minimize (2.1) subject to the constraints

$y_n f(x_n) \geq 1 - \xi_n,\ \xi_n \geq 0$ together. The corresponding Lagrangian is given by

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 + C\sum_{n=1}^{N}\xi_n - \sum_{n=1}^{N}\alpha\{y_n f(x_n) - 1 + \xi_n\} - \sum_{n=1}^{N}\mu_n\xi_n, \quad (2.2)$$

where $\alpha_n \geq 0$ and $\mu_n \geq 0$ are Lagrange multipliers.

(b) We now optimize out w, b, and $\{\xi_n\}$ and obtain the dual Lagrangian in the form as follows:

max. $\tilde{L}(\alpha) = \sum_{n=1}^{N}\alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m k(x_n, x_m)$

s.t. $0 \leq \alpha_n \leq C,$

$\sum_{n=1}^{N}\alpha_n y_n = 0\ .$ (2.3)

(C) Design kernel functions:

(i) Define the RBF kernel function k by:

$$k(x_1, x_2) = \exp(-\|x_1 - x_2\|^2/2\sigma^2)$$

(ii) Define the Polynomial kernel function K by:

$$k(x_1, x_2) = (x_1^T x_2 + 1)^d.$$

Training the models and get the decision planes.

(d) For testing, the new data z is classified as class 1 if $f \geq 0$, and as class 2 if $f < 0$:

$$w = \sum_{m=1}^{S}\alpha_m y_m \phi(x_m),$$

$$f = \langle w, \phi(z) \rangle + b = \sum_{m=1}^{S} \alpha_m y_m k(x_m, z) + b,$$

$$b = 1 - \frac{1}{|n: 0 < \alpha_i < C\,|} \sum_{n:0<\alpha_n<C} \sum\nolimits_{m=1}^{S} \alpha_m y_m k(x_m, x_n).$$

Download the refernce python code from module web-page on LearningMall: Final Coursework-->Reference Code-->INT304_Support_Vector_Machine.ipynb

(1) Adjust and modify the codes to implement an SVM with Polynomial Kernel and RBF Kernel on LFW Dataset. Show the training process and classification results *(7 points).*

(2) Explain the key differences between the polynomial kernel and the RBF kernel in the context of SVM. **Discuss the principles** of those kernel functions. What are the **challenges** in choosing a kernel for a given dataset? *(5 points)*

(3) Describe your training strategy for the kernel SVMs, including the design of the loss function and regularization term. Explain why these choices are effective. Additionally, **describe a strategy for tuning the hyperparameters** for each kernel. Document the process and your findings. (*7 points)*

(4) Results evaluation and visualization: Compare the performance of SVMs with polynomial and RBF kernels. Analyze the results and present them graphically. Create visualizations to show decision boundaries or support vectors for a simplified version of the LFW dataset (considering only two classes or using a reduced feature set for visualization purposes) *(16 points).*

Analysis and Comparison (20 marks)

Analyse and compare the advantages and disadvantages of these Three models in classification from different aspects, such as effectiveness, efficiency, complexity. Your analysis and conclusion should be well justified theoretically and/or empirically.

Report writing (10 marks)

1) Is the report well presented (including writing style, grammar and use of figures)?
2) Is the report clearly structured?
3) Is there evidence of using relevant literature and other resources with explicit references?

Report Requirements:

1) The report should be formatted using a 11pt Arial font, single-column, 1.15 Spacing (Microsoft Word) or written with the latex type setting language. (The Latex Template can be downloaded from module web-page on LearningMall).
2) The report must **not** be longer than 20 pages (15 pages as a minimum).
3) Please attach the useful code for each step as an Appendix. Appendix does not count in the total pages of the report, but only allows to put screenshots of the useful code.
4) Reports submitted without accompanying source code (in the submission package) will receive a zero mark.

Programming Requirements:

1) Students should use Python to implement the classification tasks.
2) Highly integrated libraries (e.g. pytorch, tensorflow) for deep learning are **forbidden**, but it can be used for reading and splitting data.
3) Third-party libraries (ML libraries, e.g. sklearn, LIBSVM, SVMlight) are allowed to use. Highly integrated functions from third-party libraries which directly achieving the required programming processes (such as forward and backward propagation) are forbidden.
4) Relative paths must be used for the paths of reading and writing files.

Marking Scheme:

- 70%-100% Essentially complete the tasks with sufficient and correct analysis.
- 60%-69% Shows understanding but contains a small number of errors or gaps.
- 40%-59% Clear evidence of a serious attempt at the work, showing some understanding, but with important gaps.
- 20%-39% Scrappy work, bare evidence of understanding or significant work omitted.
- <20% No understanding or little real attempt made.