Generative AI tools cannot be used in this assessment task

*In this assessment, you must **not** use generative artificial intelligence (AI) to generate any materials or content in relation to the assessment task.*

FIT3081 Image Processing - Take Home Assessment (S1, 2023)

(Due 11.55 p.m. on 9th Jun 2023)

[Weight: 30 marks]

Overview

You are given an image dataset with 15 class that **contains 5 Numerals and 10 Alphabets**. Each class will have 10 Images. Hence, the **total number of images is 150**. With the given dataset, you need to use **80% for training and the remaining 20% for testing**. An example of a Table below shows the naming of the images for training and testing.

We used a random number generator to assign 150 images for each dataset. There are 80 students in this course, and each will have a dataset that is different from the others. Each will be given a random seed number to be used in initializing the weights during the training phase of the neural network.

Please note: The neural network that will be used in this take-home assessment is the same as in assignment 3.

Question Layout

There are **three questions** to be answered in this take-home assessment.

Question 1 requires you to use Sobel Magnitude followed by threshold. There are 5 experiments to be answered in this question. You need to fill up the given Tables for all 5 experiments.

Question 2 requires you to propose an algorithm that performs better than the accuracies obtained in Question 1. Here too, you have to fill up the Table and besides that, the description of the proposed method is required.

Question 3 is about error weight correction.

Marks allocation

Those who have been very involved in assignment 3, will realize several factors affect the testing accuracy. In our case, we are using a small model and hence there is a chance that in some cases, the training data set and the type of inputs used, which is fixed for **experiments 1 to 5**, may not give higher accuracies when compared to other training datasets.

We used several of the datasets (9 of them) and found that the accuracies for experiment 1 varied from 53% to 70% for 1000 epochs. One epoch means that the learning algorithm will work through the entire training dataset. In our case, one epoch will take 120 iterations to go through the entire training set.

We have indicated an expected accuracy for the second and third experiments by comparing them with experiment 1.

Table: Layout of the naming of the images used in training and testing

Images chosen for experiments	Training Images								Testing Images	
A	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
D	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
F	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
G	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
H	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10
K	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10
M	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
O	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
Q	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
S	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
1	1a	1b	1c	1d	1e	1f	1g	1h	1i	1j
2	2a	2b	2c	2d	2e	2f	2g	2h	2i	2j
4	4a	4b	4c	4d	4e	4f	4g	4h	4i	4j
7	7a	7b	7c	7d	7e	7f	7g	7h	7i	7j
9	9a	9b	9c	9d	9e	9f	9g	9h	9i	9j

QUESTION 1 (Total 18 Marks):

There are five experiments to be carried out. Follow the instructions given in each experiment. You need to write a Python program to implement the instructions given in each experiment. Train the neural network with the training images and test the images with the remaining 20% of testing images.

Experiment 1 (6 Marks): Expected accuracy (varies from 53% onwards)

You are required to **apply the Sobel operator** to the image dataset given and **followed by a threshold number to binarize the images** for Training and Testing. **NO LIBRARY FUNCTION using SOBEL is allowed**. Those who did use it will have 1 point deducted.

Fill up the Table below after you have completed this experiment. (2 Marks)

Items	**Your Answer**
Seed number used to initialize the random weights and bias	91
What is the threshold number used to binarize the images?	255
Number of Inputs	784
Number of Hidden Neurons	25
Global Error	2.5228096927343713
Number of Epoch / Iteration	1000
Testing Accuracy	20/30 = 0.6666
List the images correctly classified with their output values	1i (0.93), 4i (0.98), 7i (0.12), D9 (0.89), J9 (0.79), L9 (0.56), T9 (0.91), U9 (0.87), 1j (0.96), 4j (0.98), 7j (1.0), D10 (0.68), G10 (0.04), J10 (0.9), L10 (1.0), P10 (0.87), S10 (0.98), T10 (0.47), U10 (1.0), X10 (0.96)

Instead of using the Magnitude, the Orientation angle is used where they are grouped into 8 different bins (45 degrees each). Will there be a change in the accuracy obtained? State the reason(s) for your answer. (2 marks)

Yes. There are severable reasons for this.

Using orientation angles instead of magnitude introduces rotation variance into the neural network. In the case where the image is rotated, the neural network is sensitive to the changes in the direction of the edges, and may lead to a decrease in its accuracy.

The use of orientation angle instead of magnitude also introduces directional bias into the neural network, which means that the network might become more sensitive to edges or features oriented along certain direction. In the case where orientations fall in between the bins, it might have a negative impact upon the accuracy.

It was found, that one of the reasons for lower accuracy was that some of the testing images were shifted form of the training images. Suggest a solution that can deal with this problem that still uses the binarized values obtained from the Sobel algorithm. Your solution uses the binarized values of the image to produce a new set of outputs that are used for training and testing of the neural network. (2 marks)

Generating multiple shifted versions of the binarized images by applying various horizontal and vertical shifts to the original binarized images. For each shifted binarized image, create a corresponding output label that represents the correct class for that image. Combining the shifted dataset and the original dataset, these can be fed into the neural network to create a more robust, shift-invariant network.

Experiment 2 (4 Marks): Expected accuracy (improved accuracy than experiment 1)

In this experiment, you need to **reorganize the number of inputs to 49**. Suggest a method that can reduce the initial 784 inputs to 49 inputs obtained using the algorithm mentioned in experiment 1. Train and test the neural network using these 49 inputs. Note that **your accuracy obtained should be better than the accuracy obtained in Experiment 1.**

Fill up the Table below after you have completed this experiment. (2 Marks)

Items	Your Answer
Seed number used to initialize the random weights and bias	91
Number of Inputs	**49**
What is the threshold number used to binarize the images?	255
Number of Hidden Neurons	50
Global Error	0.5250643206094294
Number of Epoch / Iteration	1000

Testing Accuracy	21/30 = 0.7
List the images correctly classified with their output values	1i (0.18), 4i (0.99), 8i (0.1), G9 (0.91), J9 (1.0), L9 (1.0), T9 (0.98), U9 (0.74), X9 (0.19), 1j (0.99), 4j (0.98), 7j (0.85), D10 (1.0), G10 (0.65), L10 (0.99), P10 (0.94), S10 (1.0), T10 (0.93), U10 (0.94), X10 (0.78), Y10 (0.97)

Discuss your algorithm used in this experiment and compare the accuracy between this experiment with experiment 1. What could be the reason(s) for improved accuracy? (2 marks)

The algorithm I used works by resizing the original image. It scans the 784-pixel image with a 2x2 window and performs max pooling of the pixels within that window. This results in a 7x7 = 49-pixel image containing the local maximum values of each of the 16-pixel windows. There are a few reasons for improved accuracy.

Firstly, reducing the number of inputs is effectively reducing the number of parameters passed through the neural network. This can help to alleviate overfitting, which occurs when the model is too complex and learns irrelevant features in the training data.

Pooling operations aggregate information from local neighbourhoods in the input image, which increases robustness to local variations. The network is less affected by fluctuations in the images, resulting in improved accuracy.

Experiment 3 (4 Marks): Expected accuracy (same or slightly lower than experiment 1)

In this experiment, the **number of inputs is further reduced to 16**. Suggest a method to do that. Again, the accuracy is dependent on the type of dataset used. The accuracy obtained is about the **same or slightly lower than the accuracy obtained in Experiment 1**.

Fill up the Table below after you have completed this experiment. (2 Marks).

Items	**Your Answer**
Seed number used to initialize the random weights and bias	91
Number of Inputs	**16**
Threshold number used to binarize the images.	255
Number of Hidden Neurons	100

Global Error	17.719096595805244
Number of Epoch / Iteration	1000
Testing Accuracy	12/30 = 0.4
List the images correctly classified with their output values	4i (0.87), G9 (0.83), L9 (0.98), T9 (0.97), 1j (0.97), 4j (0.63), D10 (0.93), J10 (0.16), L10 (0.98), P10 (0.57), U10 (0.95), Y10 (0.46)

Discuss your algorithm used in this experiment and compare the accuracy between this experiment with experiment 1. (2 marks)

The algorithm I used works by resizing the original image. It scans the 784-pixel image with a 7x7 window and performs max pooling of the pixels within that window. This results in a 4x4 =16-pixel image containing the local maximum values of each of the 49-pixel windows. There are a few reasons for reduced accuracy.

Firstly, reducing the input dimensions from 784 to 16 drastically removes information from the original image, causing a significant loss of information. Over-compressing the input data can greatly hinder the network's ability to accurately understand the input data.

With only 16 inputs, the neural network has limited capacity to capture and represent the complexity and diversity of features present in the original images. It may struggle to differentiate between patterns or objects, resulting in a decreased ability to classify accurately.

Experiment 4 (2 Marks):

The testing images are **added with Gaussian Noise** where the **standard derivation is 1.0** and they are **tested using the weights obtained in Experiment 1.** Write a program to add Gaussian noise to the testing images.

Fill up the Table below after you have completed this experiment. (1 Mark)

Items	**Your Answer**
Testing Accuracy	11/30 = 0.3664
List the images correctly classified with their output values	1i (0.59), 4i (0.24), G9 (0.39), L9 (0.28), P9 (0.79), T9 (0.96), 1j (0.36), 4j (0.58), 7j (0.64), P10 (0.02), T10 (0.98)

Are there any differences between the results obtained in Experiment 1 and the results obtained after adding Gaussian noise to the testing images? State the reason(s), if any for the differences in the testing accuracies between them. (1 Mark)

Yes. There are several possible reasons for this.

Gaussian noise introduced certain levels of distortion in the image, such as blurring or smudging important features that the neural network relies on for classification.

The addition of noise can introduce randomness and uncertainty into the input data, which poses more challenge for the network to make confident predictions.

The neural network is not trained with noisy images, therefore it lacks the ability to handle and adapt to noisy inputs.

Experiment 5 (2 Marks):

In experiment 4, the testing images were added with Gaussian noise. In this experiment**, propose a method to remove the Gaussian noise**. Write a program to do that. To know the effectiveness of your proposed method, the testing images with the Gaussian noise removed are tested using the weights obtained in experiment 1.

Fill up the Table below after you have completed this experiment. (1 Mark)

Items	**Your Answer**
Testing Accuracy	11/30 = 0.3664
List the images correctly classified with their output values	1i (0.67), 4i (1.0), G9 (0.99), L9 (1.0), T9 (1.0), U9 (0.99), 1j (0.92), 7j (0.94), G10 (0.86), L10 (0.99), Y10 (0.35)

Are there any differences between the results obtained in Experiment 1 and the results obtained after implementing the proposed method to remove the Gaussian noise? State the reason(s), if any for the differences in the testing accuracies. (1 Mark)

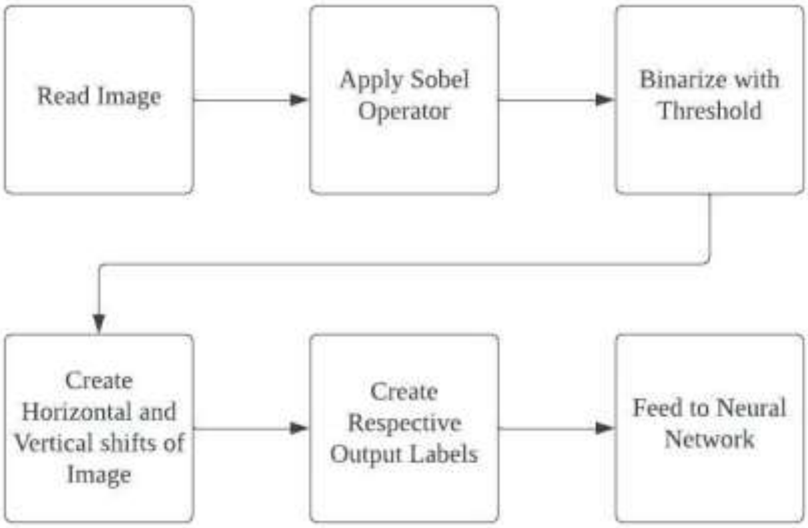
Yes. After implementing the proposed method (mean filtering) to remove the Gaussian noise, the results show that the neural network has lower accuracy compared to the results obtained in Experiment 1. There are a few reasons contributing to this difference in testing accuracy.

The process of noise removal involves blurring operations, which can inadvertently remove or distort important details and features in the image, causing information loss.

It is highly challenging to remove all traces of Gaussian noise from an image, therefore the noise removal process may not completely eliminate all the noise. The residual noise that remains in the image can still interfere with the neural network's ability to make accurate predictions.

QUESTION 2 (Total 10 Marks):

a) The testing accuracies obtained using the Sobel method followed by a threshold in the earlier experiments can be improved. **Propose an algorithm that can improve their testing accuracies.** Together with a **flowchart**, describe the details of the proposed method. (3 Marks)



The proposed algorithm adds on to the original Sobel method whereby I generate shifted versions of the input image to the neural network. This is to introduce new variants of image to increase the network's shift-variance, or put simply, the network is able to recognize the same character even if it Is shifted to a different place.

The first section is the original Sobel method followed by binarizing with a threshold. Then, the algorithm takes in the binarized image, and creates one horizontal shifted version and one vertical shifted version of the image. The corresponding labels of this image is also created and saved.

Afterwards, the images are added to the training dataset and used to train the neural network.

b) **Write a program to implement the proposed algorithm** that performs better than the accuracies obtained in experiments 1, 2, and 3. (4 Marks)

Fill up the Table (2 Marks)

Items	Your Answer
Seed number used to initialize the random weights and bias	91
Number of Inputs	784
Threshold number used to binarize the images (Optional)	255
Number of Hidden Neurons	50
Global Error	1.5087157401697164
Number of Epoch / Iteration	1000
Testing Accuracy	25/30 = 0.8334
List the images correctly classified with their output values	1i (0.03), 2i (0.22), 4i (1.0), 7i (0.74), D9 (0.58), G9 (0.01), J9 (0.93), L9 (0.97), S9 (0.66), T9 (0.89), U9 (0.92), X9 (0.65), Y9, (0.62), 1j (0.99), 2j (0.91), 4j (0.99), 7j (1.0), D10 (0.74), J10 (0.67), L10 (0.89), P10 (0.99), S10 (0.88), T10 (0.83), U10 (0.4), X10 (0.98)

Compare the proposed method's accuracy with the highest accuracy obtained from experiments 1 or 2 or 3. State the reason(s) for a better performance achieved using the proposed method. (1 Mark)

The method's accuracy is found to be 0.8334 compared to the highest accuracy obtained from experiment 2, which is 0.7.

This may be due to the new neural network's resistance to shifts in the image. In experiments 1, 2 and 3, the dataset used to train the images did not have shifted versions of the images, therefore the neural network is less robust and may be unable to accurately learn the features. When the image is slightly shifted, the network misinterprets it as a completely different character. However for the proposed method, the neural network works on a dataset where there are horizontal and vertical shifts of the same image, allowing it to learn the underlying patterns in the character image.

QUESTION 3 (Total 2 Marks):

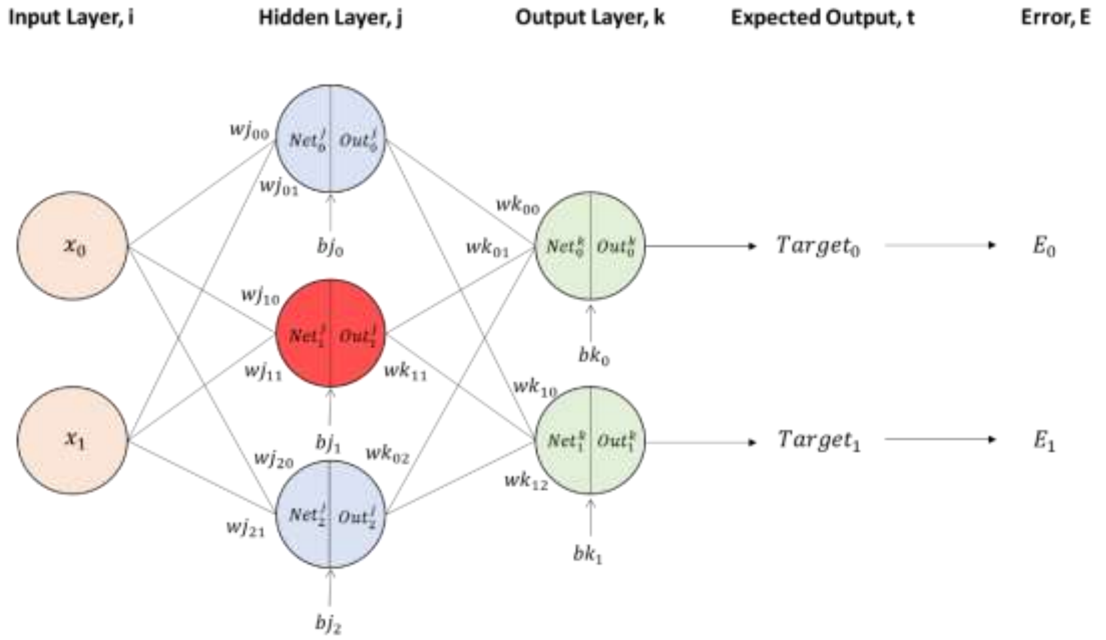
This is regarding error correction of the weights in the output layer and hidden layer. We have developed the error correction equations for both cases where all the neurons in the 3-layered neural network used the sigmoid function as the activation function. The equations used for error weight correction are:

$$\partial wk_{mn} = (Out_m^k \ - Target_m) \ * Out_m^k \ (1 - Out_m^k) \ Out_n^j \tag{1}$$

$$\partial wj_{mn} = x_n \ * Out_m^j \left(1 - Out_m^j\right) * \ \sum_{l=0}^{n} \delta_l^k \ * wk_{lm} \tag{2}$$

$$\boldsymbol{where} \ \ \delta_l^k \ = \left(Out_l^k \ - Target_l\right) * Out_l^k \left(1 - Out_l^k\right)$$

Given below is a 3-layered neural network with their labelled weights. All the neurons used the sigmoid function except one neuron drawn in red (Hidden neuron 1) in the hidden layer. For this neuron, we use the **hyperbolic tangent as an activation function**. Can we still use the same equations 1 and 2 during the error correction of weights?



Answer Yes or No with a reason for your choice for each weight given below:

Weights	Yes/No	Your reason for Yes/No
wk_{00}	Yes	Both neurons connected to this weight use the same activation function.
wk_{01}	No	Out^j_1 is required to calculate the error correction, but it is computed using a different activation function.

wj_{00}	**Yes**	Both neurons connected to this weight use the same activation function.
wj_{10}	**No**	Out^{j}_{1} is required to calculate the error correction, but it is computed using a different activation function.

====== END OF QUESTIONS ======

Submission Details

Submit the following Files. Please follow a naming convention that is easy for us to test them. For example, in Experiment 1, please use yourName_P1 and yourName_Exp1_Weights for the program and Experiment 1 weights. Follow the same naming convention for the remaining experiments.

For question 2, please use yourName_Q2 and yourName_Q2_Weights for the program and the weights obtained using the proposed method.

1) Program file for experiment 1. (yourName_P1)
2) Weights obtained in the training of neural network for experiment 1. (yourName_Exp1_Weights)
3) Program file for experiment 2. (yourName_P2)
4) Weights obtained in the training of neural network for experiment 2. (yourName_Exp2_Weights)
5) Program file for experiment 3. (yourName_P3)
6) Weights obtained in the training of neural network for experiment 3. (yourName_Exp3_Weights)
7) Program file for experiment 4 (add Gaussian noise). (yourName_P4)
8) Program file for experiment 5 (removal of Gaussian noise). (yourName_P5)
9) Program file for the proposed method in Question 2. (yourName_Q2)
10) Weights obtained in the training of neural networks for the proposed method. (yourName_Q2_Weights)