New Uzbekistan University
2023/24 Term 1
Assigned Date: 20/12/2023
Due Date: 02/01/2024, 23:59

Assignment III
Introduction to Machine Learning

Due Date

Late assignments will not be accepted and will receive ZERO mark.

Objectives

The objectives of this assignment are as follows.

- You will practice with machine learning techniques including Decision Trees, Feedforward Neural Networks, and Convolutional Neural Networks, using Python, scikit-learn and PyTorch.
- You will enhance skills in benchmarking by running the prediction and training in different split, with potentially several runs.
- You will also implement various machine learning models for tasks like satellite image pixel classification as well as fashion images classification, and critically evaluate their performance through accuracy metrics and other evaluation strategies.
- You can use the scripts given in utils.py. It is advised to explore the file first before getting started on the problems.

1 Satellite Images Pixel Classification [2 Points]

In this task, you will the satellite dataset. The full name of this dataset is Statlog (Landsat Satellite) Data Set, and it contains data for classification of pixels in satellite images.

1. Read data in Python. Then explore it; display datatypes, and check for missing values. Do we need to use scaling?
2. Build a basic decision tree classifier and fit it to the training data. Evaluate the model using accuracy, precision, recall and f1 score.
3. Define a set of parameters to tune, you can use the scikit-learn documentation of the DecisionTreeClassifier class. Explain briefly what do they stand for.
4. Run run a grid search algorithm to find the best combination of the hyper-parameters. What was the best result? Which values we need for the hyper-parameters?
5. Choose the best three combinations and run each 10 times. Average over the accuracies over the runs. Did the best score change?

2 Feedforward Neural Networks [2 Points]

You are given a python file utils.py with a function ‘check_fashion_mnist_dataset_exists’. You use this function to download and use the fashion MNIST dataset. It contains 60000 images (separate 10000 images for testing) of resolution 28x28 with 10 classes. Write an FNN to classify the images into ten classes.

1. Read data in Python. Then explore it; display some images (you can use functions in utils.py), display class names.
2. Write a two layer FNN (that extends nn.Module) to classify the images. How many parameters does it have?
3. Create a model, print it and display output probabilities for a random image from the training set.
4. Run the training by choosing the appropriate values for criterion, batch size, epochs and learning rate.
5. Make a single prediction on the trained model, display a random image from the test set and display returned probabilities for the image.
6. Report the accuracy, precision, recall, and the f1 score for the trained model on the test set.

3 Convolutional Neural Networks [2 Points]

Given the dataset and the model you trained in the previous task, you are going to make a CNN model that solves the same problem and compare its performance agains the FNN model.

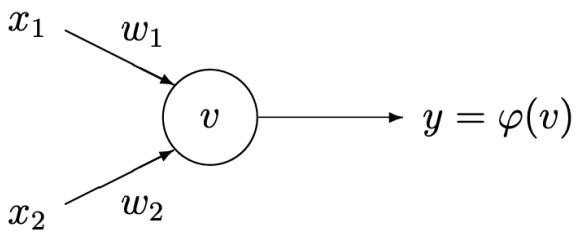
1. Create a class that (that extends nn.Module) that uses 3 convolutional blocks together with some linear layers. What is the input shape? How many layers does it have? How many parameters does it have?
2. Run the training by choosing the appropriate values for criterion, batch size, epochs and learning rate.
3. Make a single prediction on the trained model, display a random image from the test set and display returned probabilities for the image.
4. Report the accuracy, precision, recall, and the f1 score for the trained model on the test set.
5. Is this model performing better than the FNN model you created in the previous step? Why?

4 Shallow FNNs [2 Points]

Logical operators (i.e. NOT, AND, OR, XOR, etc) are the building blocks of any computational device. Logical functions return only two possible values, true or false, based on the truth or false values of their arguments. For example, operator AND returns true only when all its arguments are true, otherwise (if any of the arguments is false) it returns false. If we denote truth by 1 and false by 0, then logical function AND can be represented by the following table:

x_1	x_2	x_1 AND x_2
0	0	0
0	1	0
1	0	0
1	1	1

This function can be implemented by a single-unit with two inputs:



if the weights are w_1 = 1 and w_2 = 1 and the activation function is:

$$\phi(v) = \begin{cases} 1 & \text{if } v > 2 \\ 0 & \text{otherwise} \end{cases}$$

Note that the threshold value is 2 (v>2).

a. Test how the neural AND function works.

b. Suggest how to change either the weights or the threshold level of this single-unit in order to implement the logical OR function (true when at least one of the arguments is true):

x_1	x_2	x_1 OR x_2
0	0	0
0	1	1
1	0	1
1	1	1

c. The XOR function (exclusive or) returns true only when one of the arguments is true and another is false. Otherwise, it returns always false. This can be represented by the following table:

x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	1
1	1	0

Do you think it is possible to implement this function using a single unit? A network of several units?

Deliverables

You are required to submit your solutions as a single ipynb file. Please, put your name and university email as the first line in the notebook.

Notes

- Cheating is a serious academic offense and will be strictly treated for all parties involved. So delivering nothing is always better than delivering a copy.
- Organize your notebook appropriately. Divide it into sections and cells with clear titles for each task and subtask.
- Make your code clean with the appropriate naming conventions. So maybe you want to take a look at some references: Link 1 and Link 2.
- Make sure your plots are descriptive and self-contained.