CS 451/551 - HW3

May 14, 2022

1 Introduction

In this homework, you will use machine learning approach to recognize hand written digits. You will implement **at least three** supervised learning techniques to classify hand written digits in the provided dataset. The dataset has 70,000 images and 10 classes. Besides, you are expected to tune hyper-parameters of each algorithm. Thus, you can not only compare the classifiers but also examine the effects of hyper-parameters on the classification results. You should report the classification results with the classification task metrics (e.g. accuracy, precision, recall, F1, etc.). The deadline of this homework is **28 May 2022**.

2 Dataset

For this task, you will use MNIST (”Modified National Institute of Standards and Technology”) dataset which is a ”hello world” dataset of computer vision. The dataset has 70,000 instances with 10 classes which are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Each instance is 28x28x1 gray-scale image with corresponding label. Figure 1 display some examples of images.

”DataLoader.py” Python file is provided to fetch the dataset and to split training and test sets. You will call *”get_data”* method in the Python file to get training and test data. Besides, the method also displays some details of the dataset. You will have 56,000 training and 14,000 test instances. You have to use only training set to train classifiers. However, you have to report the classification results for both training and test sets. Figrue 2 demonstrates the class distribution of both training and test sets.

2.1 Pre-Processing

”get_data” method provides images as flatted vector (784x1). Each pixel is an integer number in range [0-255]. However, the machine learning algorithms perform better on normalized data. Pre-processing of the data is up to you; you are free to choose which pre-processing technique to use (e.g. MinMaxScaler,

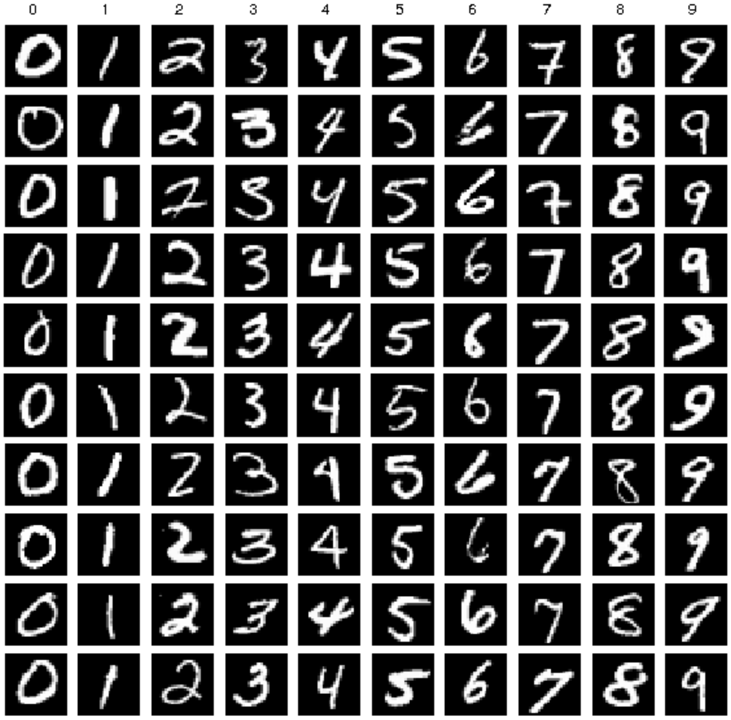


Figure 1: MNIST image examples.

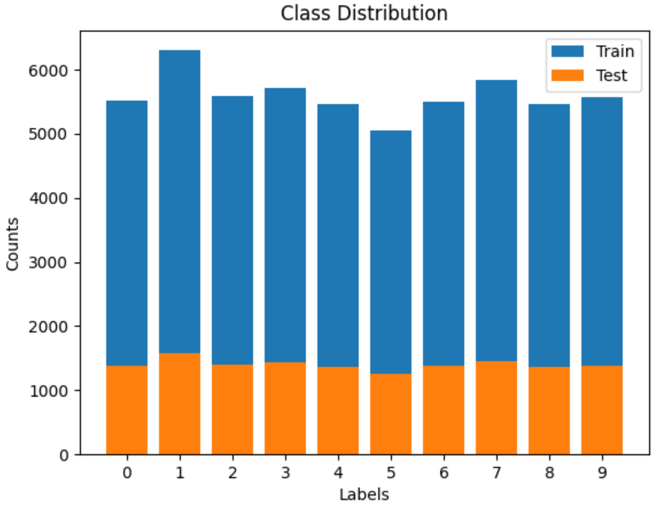


Figure 2: Class distribution of both training and test sets.

StandartScaler, PCA, LDA, etc.) The pre-processing technique you use, may yield different results depending on the data set or the task. Besides, you should also explain which pre-processing technique you chose and how you decided on it.

Hint: You can try different techniques and choose the best performing one among them. You can also points the performance results in your report.

3 Implementation

In this homework, you will use Python programming language with version 3.8 or 3.9. The allowed Python libraries are listed below:

- NumPy
- Pandas
- Scikit-Learn
- Matplotlib
- SeaBorn

Using any library except the listed ones will be penalized. Also, *"DataLoader.py"* Python file is provided to get training and test sets. However, any change is not allowed in the provided file. You will implement **at least three** machine learning algorithms which you learned in this course (e.g. KNN, Decision Tree, etc.). You are free to choose which pre-proccessing approach you will apply. You will use the classification metrics such as Accuracy, Recall, Precision, F1 and Confusion Matrix for classification results.

3.1 Hyper-Parameters

Each algorithm has its own hyper-parameters that having considerably effect on the performance. Therefore, determining the proper values of each hyper-parameters has crucial importance in machine learning. However, the effect of the hyper-parameter values on the performance differs depending on the data set or the task. Tuning the hyper-parameters is one of the expectation of the assignment. The effect of various values and combinations of hyper-parameters on performance should be examined, their performance should be compared and reported in detail.

4 Requirements

Requirements of this homework are listed below:

1. You have to implement **at least three** machine learning algorithms (you learned in this course) to classify hand written digits.

2. Accuracy, Recall, Precision and F1 scores must be reported for both training and test sets for each algorithm. You can examine the examples Table 1 and Table 2.

3. Confusion Matrix must be demonstrated for both training and test sets for each algorithm. Figure 3 and 4 are examples of confusion matrix.

4. Each classifier has various hyper-parameters. You are expected to tune these hyper-parameters. Besides, you will report the effect of hyper-parameters on the results.

5. You should also explain the results. For example, if you detect over-fitting problem, you should explain how you detected and why it over-fitted.

6. Pre-processing is up to you. You are free to choose which technique to use. You should explain how you prepare the data and why you decided to use this technique.

7. There is no minimum accuracy score limit or minimum F1 score limit. However, higher scores are desired, and they will be taken into account during grading.

8. *"DataLoader.py"* file is provided to get data. You are not allowed to change anything inside the file. Any change will be penalized.

9. Python programming language must be used with version 3.8 or 3.9. Only NumPy, Pandas, Scikit-Learn, Matplotlib and Seaborn Python libraries are allowed. Using other libraries will be penalized.

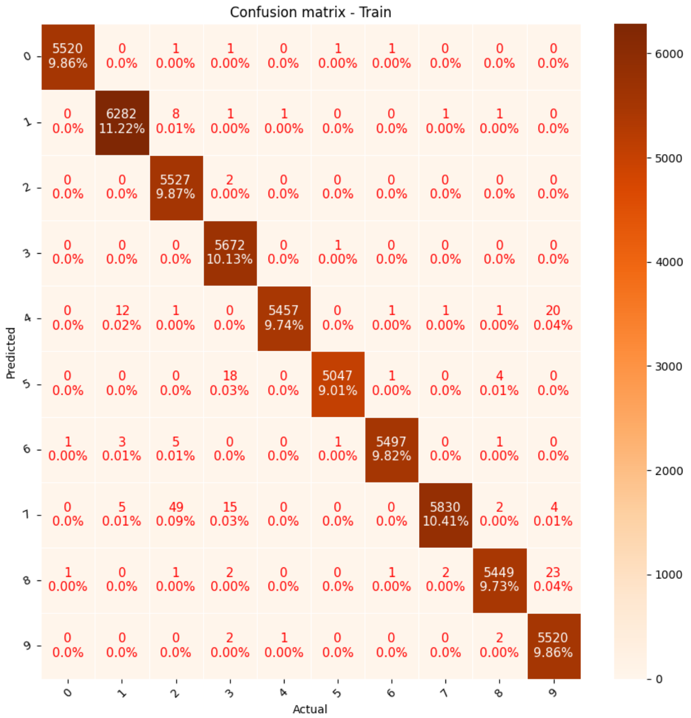
10. You will submit your homework as *zip* format. File name format is **NAME_SURNAME_STUDENTID_hw3.zip**. The zip file must contain your report as PDF file and your implementation as **.py* file format.

11. Your report must be detailed and well organized. Also, your report must be clear and in English.

12. Grading:

    - Classification results and explanations: **45pt** (15x3)
    - Hyper-parameter analysis: **45pt** (15x3)
    - Test accuracy: **10pt**

13. The deadline for this homework is **28 May 2022**.

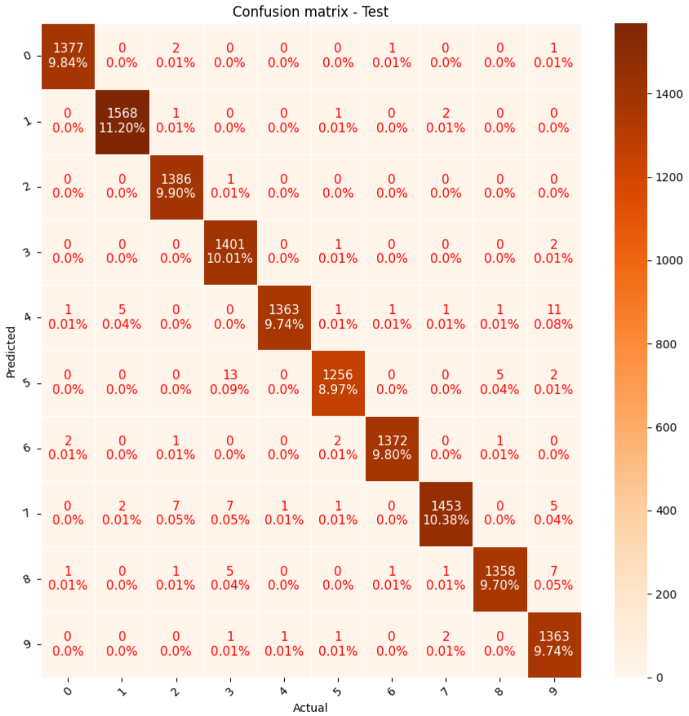


	0	1	2	3	4	5	6	7	8	9
0	5520 9.86%	0 0.0%	1 0.00%	1 0.00%	0 0.0%	1 0.00%	1 0.00%	0 0.0%	0 0.0%	0 0.0%
1	0 0.0%	6282 11.22%	8 0.01%	1 0.00%	1 0.00%	0 0.0%	0 0.0%	1 0.00%	1 0.00%	0 0.0%
2	0 0.0%	0 0.0%	5527 9.87%	2 0.00%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%
3	0 0.0%	0 0.0%	0 0.0%	5672 10.13%	0 0.0%	1 0.00%	0 0.0%	0 0.0%	0 0.0%	0 0.0%
4	0 0.0%	12 0.02%	1 0.00%	0 0.0%	5457 9.74%	0 0.0%	1 0.00%	1 0.00%	1 0.00%	20 0.04%
5	0 0.0%	0 0.0%	0 0.0%	18 0.03%	0 0.0%	5047 9.01%	1 0.00%	0 0.0%	4 0.01%	0 0.0%
6	1 0.00%	3 0.01%	5 0.01%	0 0.0%	0 0.0%	1 0.00%	5497 9.82%	0 0.0%	1 0.00%	0 0.0%
7	0 0.0%	5 0.01%	49 0.09%	15 0.03%	0 0.0%	0 0.0%	0 0.0%	5830 10.41%	2 0.00%	4 0.01%
8	1 0.00%	0 0.0%	1 0.00%	2 0.00%	0 0.0%	0 0.0%	1 0.00%	2 0.00%	5449 9.73%	23 0.04%
9	0 0.0%	0 0.0%	0 0.0%	2 0.00%	1 0.00%	0 0.0%	0 0.0%	0 0.0%	2 0.00%	5520 9.86%

Figure 3: Example Confusion Matrix of training set.

Label	Precision	Recall	F1	Accuracy
0	1.00	1.00	1.00	99.96%
1	1.00	1.00	1.00	99.68%
2	1.00	0.99	0.99	98.84%
3	1.00	0.99	1.00	99.28%
4	0.99	1.00	1.00	99.96%
5	1.00	1.00	1.00	99.94%
6	1.00	1.00	1.00	99.93%
7	0.99	1.00	0.99	99.93%
8	1.00	1.00	1.00	99.80%
9	1.00	0.99	1.00	99.16%
Average	**1.00**	**1.00**	**1.00**	**99.65%**
Weighted Average	**1.00**	**1.00**	**1.00**	**99.64%**

Table 1: Training Set Results



	0	1	2	3	4	5	6	7	8	9
0	1377 9.84%	0 0.0%	2 0.01%	0 0.0%	0 0.0%	0 0.0%	1 0.01%	0 0.0%	0 0.0%	1 0.01%
1	0 0.0%	1568 11.20%	1 0.01%	0 0.0%	0 0.0%	1 0.01%	0 0.0%	2 0.01%	0 0.0%	0 0.0%
2	0 0.0%	0 0.0%	1386 9.90%	1 0.01%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%
3	0 0.0%	0 0.0%	0 0.0%	1401 10.01%	0 0.0%	1 0.01%	0 0.0%	0 0.0%	0 0.0%	2 0.01%
4	1 0.01%	5 0.04%	0 0.0%	0 0.0%	1363 9.74%	1 0.01%	1 0.01%	1 0.01%	1 0.01%	11 0.08%
5	0 0.0%	0 0.0%	0 0.0%	13 0.09%	0 0.0%	1256 8.97%	0 0.0%	0 0.0%	5 0.04%	2 0.01%
6	2 0.01%	0 0.0%	1 0.01%	0 0.0%	0 0.0%	2 0.01%	1372 9.80%	0 0.0%	1 0.01%	0 0.0%
7	0 0.0%	2 0.01%	7 0.05%	7 0.05%	1 0.01%	1 0.01%	0 0.0%	1453 10.38%	0 0.0%	5 0.04%
8	1 0.01%	0 0.0%	1 0.01%	5 0.04%	0 0.0%	0 0.0%	1 0.01%	1 0.01%	1358 9.70%	7 0.05%
9	0 0.0%	0 0.0%	0 0.0%	1 0.01%	1 0.01%	1 0.01%	0 0.0%	2 0.01%	0 0.0%	1363 9.74%

Figure 4: Example Confusion Matrix of test set.

Label	**Precision**	**Recall**	**F1**	**Accuracy**
0	1.00	1.00	1.00	99.71%
1	1.00	1.00	1.00	99.56%
2	1.00	0.99	1.00	99.14%
3	1.00	0.98	0.99	98.11%
4	0.99	1.00	0.99	99.85%
5	0.98	0.99	0.99	99.45%
6	1.00	1.00	1.00	99.78%
7	0.98	1.00	0.99	99.59%
8	0.99	1.00	0.99	99.49%
9	1.00	0.98	0.99	97.99%
Average	**0.99**	**0.99**	**0.99**	**99.27%**
Weighted Average	**0.99**	**0.99**	**0.99**	**99.26%**

Table 2: Test Set Results