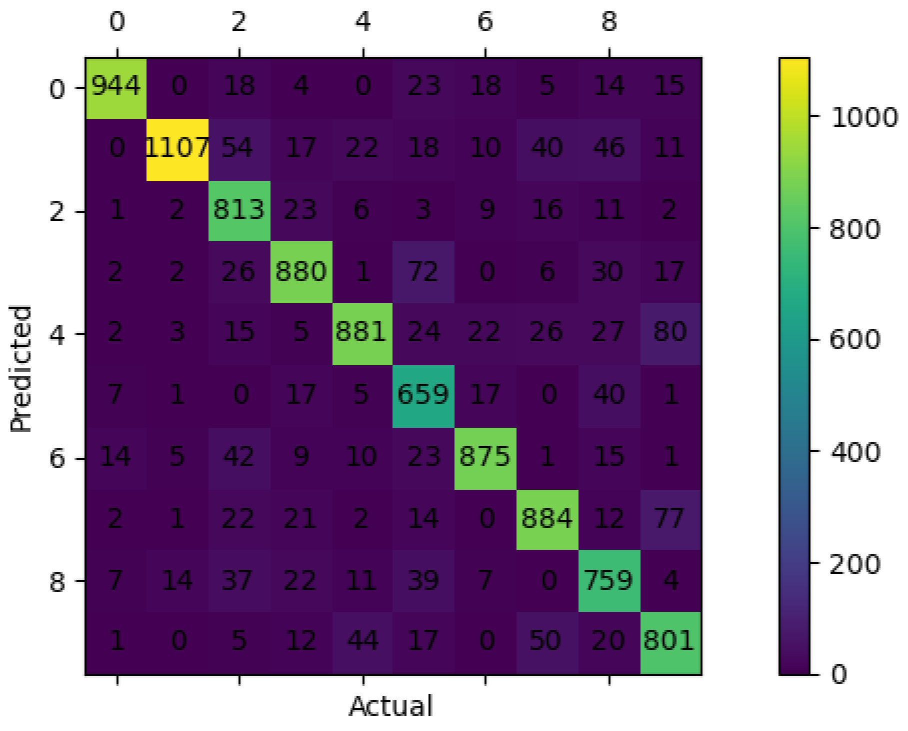
May Zhang
A17452487

Evaluation of multiclass classifiers

I. One-versus-all multiclass classifier

The error rate of the one-versus-all multiclass classifier is 13.97%, which is moderately low. As shown in figure 1, most of the data lie in the diagonal boxes, suggesting that most of the test data are predicted correctly with a moderately low false positive and negative rate. With an error rate between the range of 10-20%, the one-versus-all classifier generalizes moderately well on the test data.

In order to determine the easiness of the digits for recognition, the recall rate for each digit is calculated. This tells us given the digit, what is the percentage that the digit is predicted correctly. We will set the threshold for a decent recall rate at 0.80, and a good recall rate at 0.95. Based on this standard, from the recall rate given in table 1, digits 0 and 1 have a good recall rate, suggesting they are easy to recognize, and digits 2, 5, 8, and 9 have a recall rate below the threshold, suggesting they are hard to recognize.



944	0	18	4	0	23	18	5	14	15
0	1107	54	17	22	18	10	40	46	11
1	2	813	23	6	3	9	16	11	2
2	2	26	880	1	72	0	6	30	17
2	3	15	5	881	24	22	26	27	80
7	1	0	17	5	659	17	0	40	1
14	5	42	9	10	23	875	1	15	1
2	1	22	21	2	14	0	884	12	77
7	14	37	22	11	39	7	0	759	4
1	0	5	12	44	17	0	50	20	801

Figure1. Confusion matrix of predicted value v. actual value for the performance of one-versus-all multiclass classifier on the test data.

```
           0      1      2      3      4      5      6     7      8      9
Digit
Rate   0.963  0.975  0.788  0.871  0.897  0.739  0.913  0.86  0.779  0.794
```

Table 1. Table showing the recall rate for each digit predicted by one-versus-all classifier.

II. One-versus-one multiclass classifier

The error rate of the one-versus-one multiclass classifier is 7.03%, which is low. As shown in figure 2, most of the data lie in the diagonal boxes, suggesting that most of the test data are predicted correctly with a low false positive and negative rate. With an error rate below 10%, the one-versus-one classifier generalizes well on the test data.

Based on the recall rate shown in table 2, digits 0 and 1 have a good recall rate, suggesting they are easy to recognize. Digits 5 and 8 have a comparatively low recall rate compared to other digits, but all the digits from 0 to 9 still have a recall rate above the threshold, suggesting that the one-versus-one classifier is able to recognize all the digits at a decent recall rate.

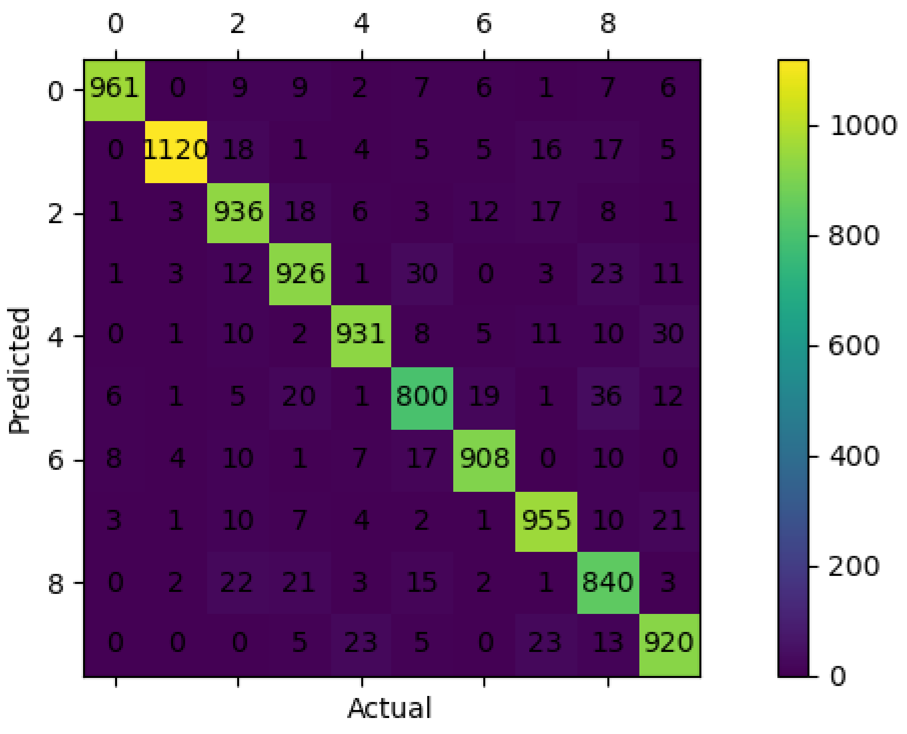


Figure2. Confusion matrix of predicted value v. actual value for the performance of one-versus-one multiclass classifier on the test data.

```
          0      1      2      3      4      5      6      7      8      9
Digit
Rate  0.981  0.987  0.907  0.917  0.948  0.897  0.948  0.929  0.862  0.912
```

Table 2. Table showing recall rate for each digit predicted by one-versus-one classifier.

The reason that 0 and 1 have a higher recall rate compared to other digits in both classifiers could be there is more training data for these two digits such that the classifiers better predict the test data. For the digits that are harder to recognize, it might result from the lower number of training data in the database, or the digit might look similar to another digit such that the classifier would sometimes misclassify it.

Overall, the one-versus-one multiclass classifier performs slightly better than the one-versus-all multiclass classifier, but it also takes more time for it to calculate the parameters and give the predictions. Both multiclass classifiers use the similar mechanisms built from the binary classifier that calculate the least-square solution of the data. Since the one-versus-one multiclass classifier is an $O(n^2)$ implementation while the one-versus-all multiclass classifier is an $O(n)$ implementation, it takes more time for the one-versus-one multiclass classifier to run.

Evaluation of different feature mappings

In this part, I transform the training data into 1000 feature vectors by first putting them into a randomized matrix and then using the non-linear mappings. The prediction of the test data also requires the transformation of the feature vectors first. The one-versus-all classifier is selected to make the predictions based on two reasons: 1) this classifier has a comparatively higher error rate so it would be easier to see if the feature mapping would make any improvement. 2) since it already takes a long time to transform the data into feature vectors, it would be helpful to use the classifier that requires less time to run.

I. Identity function

For the identity function, only a randomized matrix and biased vector is used to transform the data. The classifier has an error rate of 14.23% and 13.97% on the prediction of training data and test data respectively. The error rate for the test data is the same as the error rate with the one-versus-all classifier itself. This is because the superposition of the linear mappings is still linear so the parameters calculated would not change. The error rate for the training data is slightly higher than the error rate for the test data. This is unexpected but it could be the reason that the data size for the test data is smaller and the data are slightly easier to recognize in general. But this information suggests that at least the classifier is not over-fitting and generalizes well on the test data.

II. Sigmoid function

For the sigmoid function, the classifier has an error rate of 8.19% and 8.29% on the prediction of training data and test data respectively. This feature mapping takes more time to run, but the error rate of the test data is also lower than the error rate without feature mapping (13.97%). Both the error rate for training and test data are low, and the error rate of the test data is similar to the one of training data. Hence the feature mapping generalizes well on the test data, and it also helps the classifier to better predict the digits.

III. Sinusoidal function

For the sinusoidal function, the classifier has an error rate of 82.28% and 90.16% on the prediction of training data and test data respectively. This suggests that the feature mapping is not working with a high error rate, and it’s bad at generalization with a large gap between the error rate on training and test data. This is reasonable since the sinusoidal function is not monotone and does not help to disperse the data at different values. It also takes a long time to run this feature mapping. Hence the feature mapping with sinusoidal function cannot be used for classifying the digits.

IV. Rectified Linear Unit (ReLU) function

For the ReLU function, the classifier has an error rate of 5.42% and 5.79% on the prediction of training data and test data respectively. The error rate on the test data is significantly lower than the classification without the feature mapping (13.97%), and the error rates are consistent in both the prediction of training and test data. This feature mapping also takes shorter time to run compared to the sigmoid function mapping but it performs better indeed.

Overall, the feature mapping with identity function performs equally well compared to the prediction made by the one-versus-all multiclass classifier without data transformation, the feature mappings with sigmoid function and the ReLU function perform better, and the sinusoidal function doesn’t work well. The feature mapping with identity function, sigmoid and ReLU function all perform well on the prediction of training data and generalize well on the test data. The ReLU function works the best with the lowest error rate, well generalization with consistent error rate, and relatively fast to run.

Evaluation on the number of features L

The number of features L is varied at the level of L = 250, 500, 1000, 1500, and 2000 respectively, and the error rate of the prediction made by the one-versus-all multiclass classifier using the feature mapping with the ReLU function is calculated. The error rates for the test data are 11.60%, 8.09%, 5.42%, 4.30%, and 3.51% respectively with L = 250, 500, 1000, 1500, and 2000. The error rate for the training data and the test data always have similar error rates with different numbers of features in this feature mapping.

As shown in figure 3, the error rate decreases with the increase in the number of features, but the rate of decrease slows down as the number of features increase. The pattern seems to follow an exponential decay as the fitting curve with the logarithmic function seems to well fit the scatter plot. This is reasonable since more features would help the classifier to learn about the data better. The cost of increasing the features is that more time is needed to calculate the parameters and give the predictions. But as the features increased to a certain range (maybe around 2000 features), little improvement could be made just by increasing the features and the error rate does not lower by a lot. So it might not be worth it to increase the features and thus the running time to trade for a lower error rate.

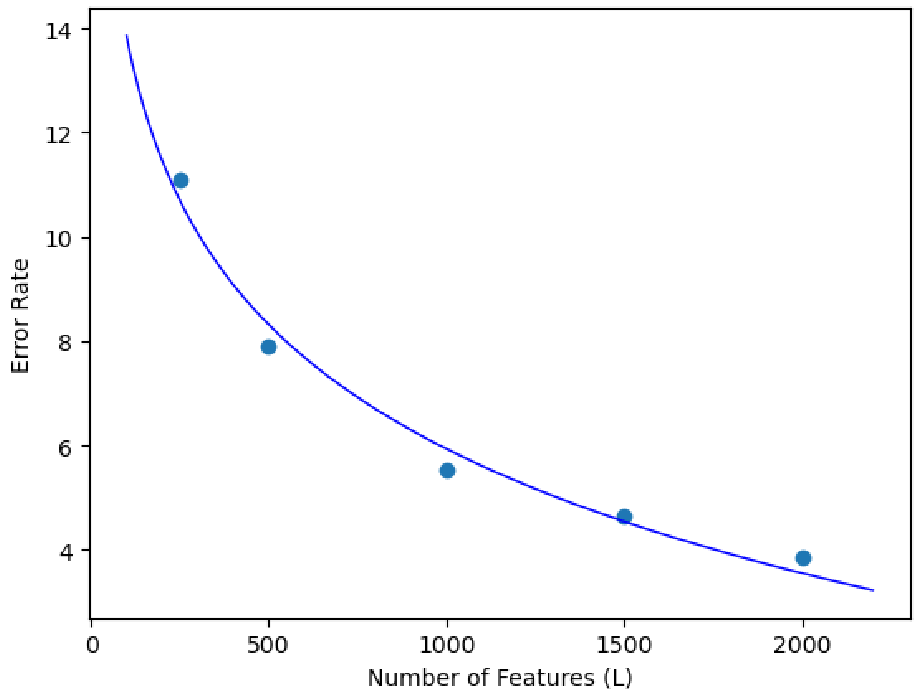


Figure3. The scatter plot of the error rate v. the number of features L with an exponential fitting curve,