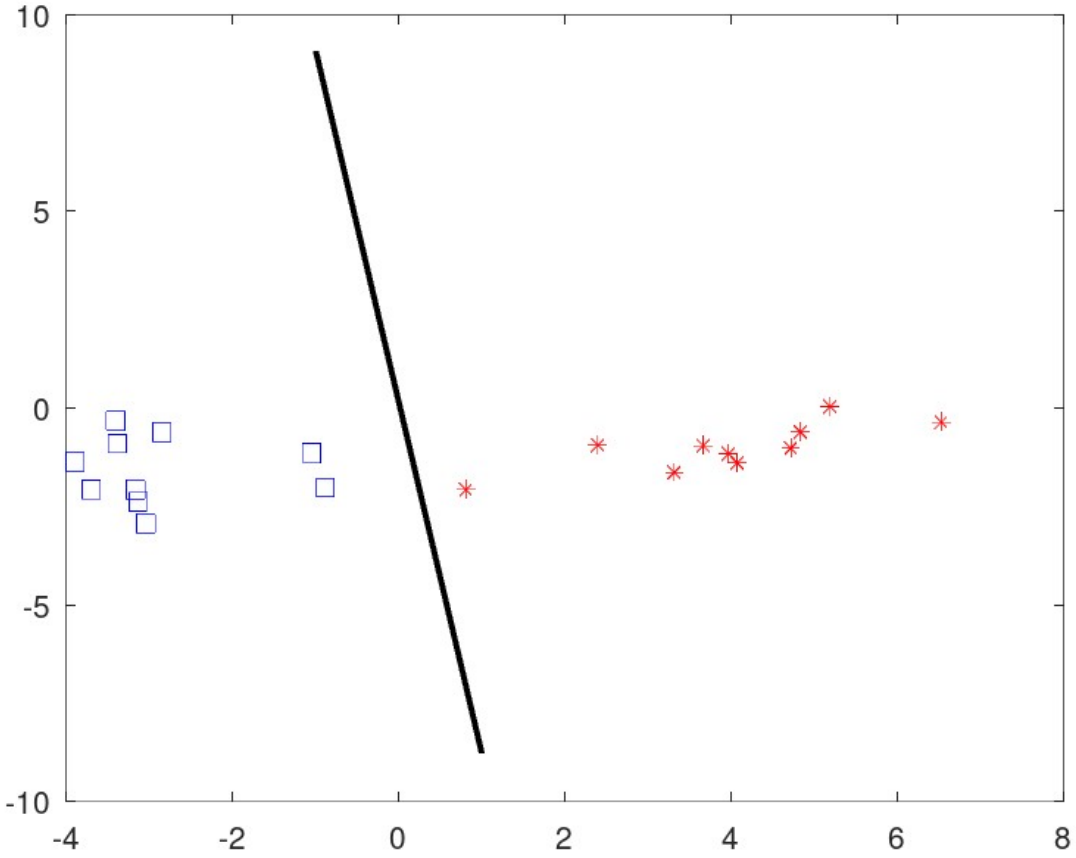
Recognition of handwritten digits with linear classifiers

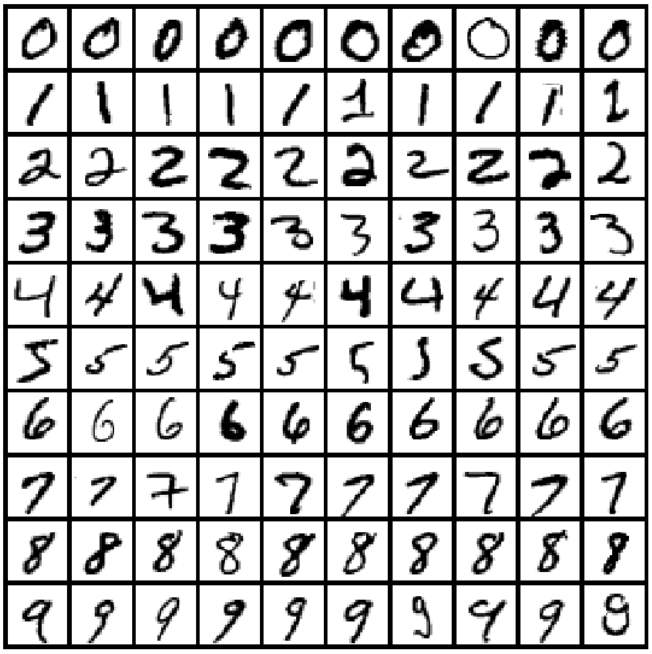
By Rebecca UFITAMAHORO

Perceptron

We begin our project by ensuring that the perceptron function is working correctly. We visualize the data and confirm that the perceptron function is performing as expected.



We continue by testing the performance of perceptron on our test data, which we have visualised below:



We have two tests for this, using a simpler and harder case to see how well it functions in both, our test cases are:

• Comparison between 1 and 0 (Easy)

Comparison between 4 and 9 (Difficult)

EASY CASE

- Positive accuracy: 0.9961
- Negative accuracy: 0.9984

DIFFICULT CASE

- Positive accuracy: 0.9579
- Negative accuracy: 0.9565

The results show that the easy case achieves an accuracy of over 99%, while the difficult case has an accuracy just under 96%. Although the accuracy is not perfect in the difficult case, it is still sufficient for our intended use.

OvO Analysis

The error rates of individual classifiers in the canonical One-vs-One (OvO) approach are analyzed for both the original and extended datasets. The digit classification outcomes for both cases are presented.

Testing OvO Classification returns us this confusion matrix

944	0	4	2	0	6	6	2	1	0	15
0	1106	3	3	0	1	1	1	10	0	10
2	2	944	11	5	3	7	8	13	0	37
0	0	9	909	1	30	0	7	9	4	41
1	0	5	1	909	0	6	7	5	27	21
9	2	6	36	3	780	7	1	14	5	29
8	3	13	1	6	15	887	1	3	0	21
0	4	19	7	6	0	0	935	5	21	31
4	4	3	27	4	22	6	4	840	7	53
3	5	3	6	38	3	1	29	7	877	37

From the result, It seems like the perceptron has the biggest issue differentiating 3 & 5, 3 & 8, 4 & 9, and interestingly enough 7 & 9. With the overall statistic being

- 91.4% Proper Classification
- 5.8% Error Rate
- 2.8% Reject Rate

OvO Classification

First Class	Second Class	Error
0	1	0.001421239637
0	2	0.011278511910
0	3	0.007881201261
0	4	0.003824904377
0	5	0.016748942172
0	6	0.009289755933
0	7	0.003774204135
0	8	0.009002887719
0	9	0.005475067385
1	2	0.010314960630
1	3	0.011885341412
1	4	0.003655435474
1	5	0.006084025323
1	6	0.002764612954
1	7	0.006611824402
1	8	0.018184705789
1	9	0.005752107793
2	3	0.031681694102
2	4	0.018474576271
2	5	0.027067404869
2	6	0.023829572247
2	7	0.015544465352
2	8	0.034126513676
2	9	0.015453094818
3	4	0.005094796626
3	5	0.051159972299
3	6	0.005975599635
3	7	0.014198128429

3	8	0.043481889501
3	9	0.020529801325
4	5	0.011187072716
4	6	0.010629251701
4	7	0.012389526720
4	8	0.008552125203
4	9	0.040878636248
5	6	0.025928212364
5	7	0.008129385590
5	8	0.045599716111
5	9	0.017854001759
6	7	0.002872855618
6	8	0.012235534030
6	9	0.003792028314
7	8	0.011142291185
7	9	0.050433928279
8	9	0.022372881356

Expanding our dataset, we get far better results. With improvement of accuracy all across the board

964	1	1	0	0	1	4	1	2	0
0	1119	2	1	1	0	3	1	2	0
5	2	987	6	2	0	5	5	8	1
0	0	3	976	0	8	0	6	6	6
0	0	3	0	951	0	4	0	0	10
2	0	2	9	1	850	3	0	5	2
7	4	2	0	6	5	926	0	1	0
0	3	10	3	2	0	0	979	3	7
1	0	3	10	2	4	0	4	929	3
3	4	1	4	13	1	1	6	1	962

- 96.4% Proper Classification
- 2.4% Error Rate
- 1.2% Reject Rate

First Class	Second Class	Error
0	1	0.0009474930912
0	2	0.0037033919704
0	3	0.0017421602787
0	4	0.0014449638759
0	5	0.0018511988717
0	6	0.0050671395997
0	7	0.0012307187397
0	8	0.0020383896722
0	9	0.0027796495957
1	2	0.0037007874016
1	3	0.0027965509205
1	4	0.0038143674507
1	5	0.0009865987010
1	6	0.0029225908373
1	7	0.0046129007458
1	8	0.0028587310411
1	9	0.0025214719092
2	3	0.0027297543221
2	4	0.0016101694915
2	5	0.0010545742157
2	6	0.0010104412260
2	7	0.0041724617524
2	8	0.0028791599627
2	9	0.0025195263291
3	4	0.0005846487931
3	5	0.0041551246537
3	6	0.0014109054693
3	7	0.0031461761859
3	8	0.0060090135203
3	9	0.0037251655629
4	5	0.0006215040398
4	6	0.0022959183673
4	7	0.0036342611712
4	8	0.0017959462927
4	9	0.0102620642863
5	6	0.0029985007496
5	7	0.0008557247989
5	8	0.0015968772179
5	9	0.0028144239226
6	7	0.0004924895346
6	8	0.0009346588495
6	9	0.0005056037752
7	8	0.0028887421591
7	9	0.0057311282135
8	9	0.0027118644068

OvO Classification - Expanded Datase

OvR Analysis

Error rates of individual basic classifiers in OVR solution (for original & extended data set). Digit classification results for both cases.

We adapt unamvoting.m and voting.m to add group voting functionality. Running it, we get this confusion matrix

863	0	0	0	1	2	1	1	1	0	111
0	990	2	2	0	1	1	1	10	0	128

1	3	756	6	4	1	5	6	23	2	225
2	0	5	777	0	16	2	1	11	5	191
1	0	0	1	777	1	2	1	11	10	178
5	1	3	28	6	539	8	2	15	6	279
8	1	2	1	4	9	803	1	5	0	124
3	2	19	0	2	0	1	799	2	7	193
5	4	2	14	5	11	3	2	659	1	268
0	6	1	6	20	6	0	11	14	599	346

Whilst it is more computationally efficient when compared to OvO, the accuracy has been significantly lowered to
• 75.6% Proper Classification
• 4% Error Rate
• 20.4% Reject Rate

Class	Error
0	0.010950000000
1	0.009866666667
2	0.027233333333
3	0.031816666667
4	0.025233333333
5	0.046916666667
6	0.016816666667
7	0.020133333333
8	0.052183333333
9	0.047233333333

OvR Classification

The reason of the high rejection amount is due to us rejecting draws.
Expanding our dataset and running the computation again significantly improved our results,

with nearly a 20% increase in accuracy.

949	0	0	0	0	0	1	1	0	1	28
0	1093	2	1	0	0	1	0	1	0	37
5	0	963	2	1	0	1	4	2	0	54
0	0	2	939	0	3	0	3	3	1	59
0	0	3	0	917	0	1	0	1	4	56
3	0	0	3	1	822	3	0	1	2	57
2	3	0	0	3	3	908	0	1	0	38
0	4	8	1	0	0	0	944	1	8	62
0	0	1	3	2	3	1	2	883	3	76
2	3	0	5	6	0	2	1	0	940	50

• 93.6% Proper Classification
• 1.2% Error Rate
• 5.2% Reject Rate

0	0.003216666667
1	0.003833333333
2	0.006266666667
3	0.008533333333
4	0.005766666667
5	0.007366666667
6	0.004050000000
7	0.007116666667
8	0.010950000000
9	0.011016666667

OvR Classification - Expanded

Balanced OvR Analysis

This analysis examines the error rates of individual classifiers in the One-vs-Rest (OvR) solution, which was trained on a balanced positive/negative class dataset, for both the original and extended datasets. The results of digit classification for both cases are presented.

Due to the inherent imbalance between the positive and negative classes, it could potentially impact the classifier's accuracy. To address this, we explored reducing the quantity of negative classifications by varying percentages in order to assess any improvements in accuracy or overall performance. Specifically, we tested two configurations: reducing the negative class by 20% and 80%, representing a slight and extreme reduction in data, respectively. Additionally, we evaluated these configurations with the expanded dataset, and the results are shown below.

class	OvR 20% Error	OvR 80% Error	OvR 20% Expanded Error	OvR 80% Expanded Error
0	0.02000836170	0.01207611613	0.003941945888	0.004045702204
1	0.01776066215	0.01069865456	0.005287964134	0.004437510131
2	0.06338322104	0.03108039435	0.005962673663	0.006687671511
3	0.07445738956	0.03721308145	0.009698976876	0.009993906155
4	0.04407003238	0.02826812007	0.007015229644	0.007036524852
5	0.14307569916	0.05461619948	0.007771862187	0.007802314211
6	0.03208076946	0.01884644317	0.004480554394	0.003537520076
7	0.03719809602	0.02302210853	0.009049773756	0.006679253710
8	0.10979922086	0.05647293285	0.011027869344	0.009334201001
9	0.08064900978	0.05155205009	0.011095204009	0.011668326794

OvR Classification Error (Reduced)

	Proper Classification	Error	**Reject Rate**
OvR 20%	51.4%	3.4%	45.2%
OvR 80%	73.5%	3.8%	22.7%
OvR 20% Expanded	90.3%	1%	8.7%
OvR 80% Expanded	93.8%	1.3%	5%

OvR Classification Details(Reduced)

Analysis of Data Reduction and Accuracy Impact

Upon reviewing the data and comparing it to the previous section, we can see that the difference between reducing the negative classification data to 80% and not reducing it is minimal, likely within the margin of error. However, drastically reducing the negative data to just 20% could significantly affect our accuracy, especially without expanding the dataset. This leads to an approximate 20% decrease in performance. Interestingly, when we expand the dataset, the accuracy gap narrows, with the result only about 4% worse, at least in this specific case.

Alternative Implementation

An alternative approach to enhancing classification quality involves using a dual-layer method that incorporates both OvR (One-vs-Rest) and OvO (One-vs-One) classification techniques. Although this method would demand more resources and inherit the limitations of both OvO and OvR, it could potentially be implemented in a novel way, starting with OvR and then refining the results with OvO. While we will not test the full performance of this approach, we can assess the accuracy of the model using a basic implementation of the algorithm. The results for both the standard and expanded datasets are provided using our custom voting.m implementation.

Normal Dataset:

950	0	3	1	0	6	8	2	1	1	8
0	1104	3	2	0	2	1	1	11	0	11
3	3	946	6	5	3	7	10	18	0	31
2	0	8	920	0	25	0	10	11	3	31
1	0	5	1	912	0	5	6	4	27	21
9	0	6	41	4	766	8	0	14	7	37
9	2	13	1	7	16	883	2	2	0	23
3	1	18	5	5	0	0	936	5	21	34
5	3	3	23	4	22	7	6	858	2	41
3	5	0	7	33	6	0	29	12	887	27

• 91.6% Proper Classification
• 5.7% Error Rate
• 2.7% Reject Rate

Expanded Dataset:

964	1	1	0	0	0	4	1	0	0	9
0	1119	2	1	0	0	3	1	1	0	8
5	1	991	5	2	0	3	7	7	0	11
0	0	1	971	0	8	0	6	6	4	14
4	1	2	0	946	0	4	1	1	11	12
3	0	0	10	1	858	2	1	4	3	10
5	2	2	0	6	5	927	0	1	0	10
0	6	8	2	2	0	0	978	4	13	15
4	1	3	9	2	3	2	3	929	1	17
2	4	1	4	9	3	1	9	1	965	10

• 96.5% Proper Classification
• 2.4% Error Rate
7• 1.1% Reject Rate

We can observe that the performance is only marginally better than OvO classification. However, considering the efficiency, I believe it is not worth using unless additional resources are available and the small increase in accuracy is deemed essential. This highlights why OvO and OvR are the industry standards, as they balance accuracy and speed with relatively straightforward implementations.