Handin 1

Søren Meldgaard 201303712, Malthe Bisbo 201303718

September 25, 2017

1 PART I: LOGISTIC REGRESSION

1.1 CODE

1.1.1 SUMMARY

We achieve and test and validation accuracy of 0.893. The regularization used was very small, which either suggest that perhaps our model is too simple since overfitting is not even possible.

1.1.2 RESULTS AND PLOTS

results_2 _vs _7.csv:

Alg.	Time	Ein	Etest
Batch Gradient Descent	0.22021690499968827	0.9826589595375722	0.9844961240310077
Mini Batch Gradient Descent	1.0230391009999948	0.9980732177263969	0.9864341085271318
Scipy Optimize	1.7462638160004644	0.9980732177263969	0.9844961240310077

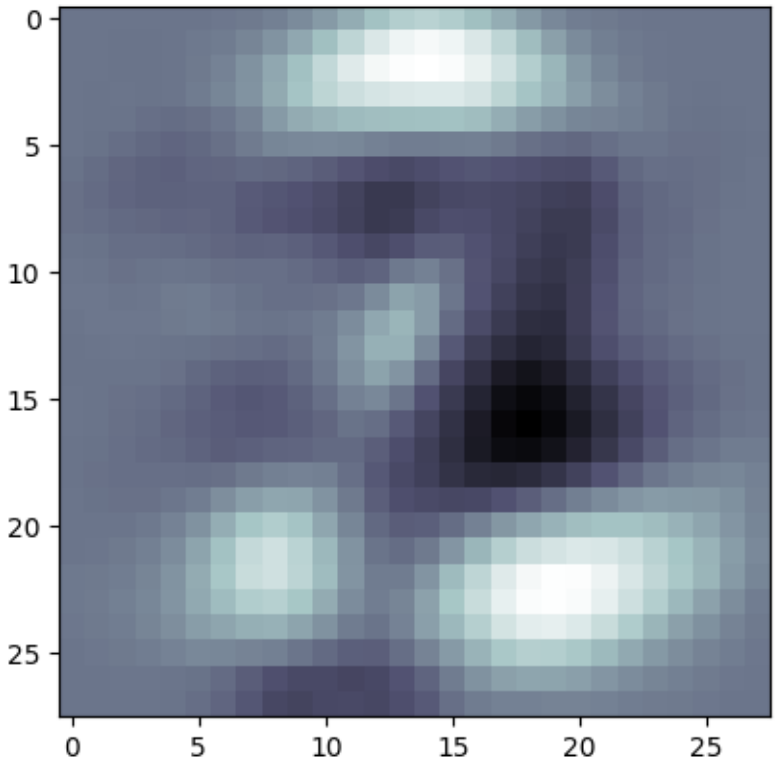
pairwise_scores_train.csv

0	1	2	3	4
0.0	0.0	0.0	0.0	0.0
0.9990366088631984	0.0	0.0	0.0	0.0
0.9990366088631984	0.9894026974951831	0.0	0.0	0.0
1.0	0.9942196531791907	0.9927745664739884	0.0	0.0
1.0	0.9951830443159922	0.9985549132947977	0.9995183044315993	0.0
0.9971098265895953	0.9961464354527938	0.9995183044315993	0.9922928709055877	0.998073217726
0.9956647398843931	0.9985549132947977	1.0	1.0	1.0
0.9995183044315993	0.9985549132947977	0.9951830443159922	0.9975915221579962	0.993737957610
0.9975915221579962	0.9812138728323699	0.9927745664739884	0.9879576107899807	0.999036608863
0.9975915221579962	0.9797687861271677	0.9966281310211946	0.9918111753371869	0.945568400770

pairwise_scores_test.csv

0	1	2	3	4
0.0	0.0	0.0	0.0	0.0
0.998062015503876	0.0	0.0	0.0	0.0
0.998062015503876	0.9767441860465116	0.0	0.0	0.0
0.9961240310077519	0.9864341085271318	0.9883720930232558	0.0	0.0
0.998062015503876	0.9844961240310077	0.9941860465116279	0.998062015503876	0.0
0.9903100775193798	0.9883720930232558	0.9825581395348837	0.9670542635658915	0.988372093023
0.9844961240310077	0.9903100775193798	0.9844961240310077	0.9903100775193798	0.986434108527
0.998062015503876	0.9883720930232558	0.9786821705426356	0.9825581395348837	0.992248062015
0.9883720930232558	0.9728682170542635	0.9612403100775194	0.9825581395348837	0.992248062015
0.9941860465116279	0.9554263565891473	0.9922480620155039	0.9709302325581395	0.932170542635

logistic_regression _two _class:



logistic_regression_classification_report:

```
Classification Report
             precision    recall  f1-score   support

          0       0.98      0.97      0.97       258
          1       0.89      0.89      0.89       258
          2       0.93      0.88      0.90       258
          3       0.97      0.84      0.90       258
          4       0.97      0.87      0.92       258
          5       0.90      0.86      0.88       258
          6       0.92      0.97      0.94       258
          7       0.90      0.96      0.93       258
          8       0.79      0.90      0.84       258
          9       0.79      0.86      0.83       258

avg / total       0.90      0.90      0.90      2580
```

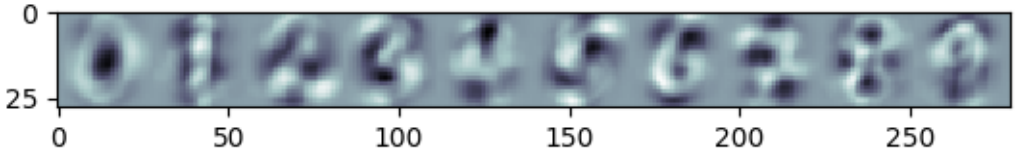
logistic_regression _confusion _matrix:

0	1	2	3	4	5	6	7	8	9
252	0	1	1	0	0	2	0	2	0
0	235	4	0	1	1	0	2	7	8
0	13	231	1	0	0	1	4	7	1
0	3	3	236	0	2	1	6	4	3
0	2	0	0	231	3	1	5	3	13
3	1	2	12	1	210	4	2	20	3
3	2	1	0	2	3	243	0	4	0
0	6	2	0	0	1	0	242	4	3
4	5	10	5	0	1	0	2	230	1
0	14	0	11	9	0	0	6	8	210

logistic_regression _accuracy:

```
train_accuracy,test_accuracy
20.9265895953757225,0.8992248062015504
```

logistic_regression _validation _scores:



reg. params	val. score
1e-15	0.8713872832369942
1e-14	0.8868015414258189
1e-13	0.8834296724470135
1e-12	0.8921001926782274
1e-11	0.869942196531792
1e-10	0.8930635838150289
1e-09	0.8853564547206165
1e-08	0.8877649325626205
1e-07	0.8872832369942196
1e-06	0.8834296724470135
1e-05	0.8680154142581888
0.001	0.8892100192678227
0.01	0.8810211946050096
0.1	0.7456647398843931
1.0	0.5496146435452793
10.0	0.09778420038535646

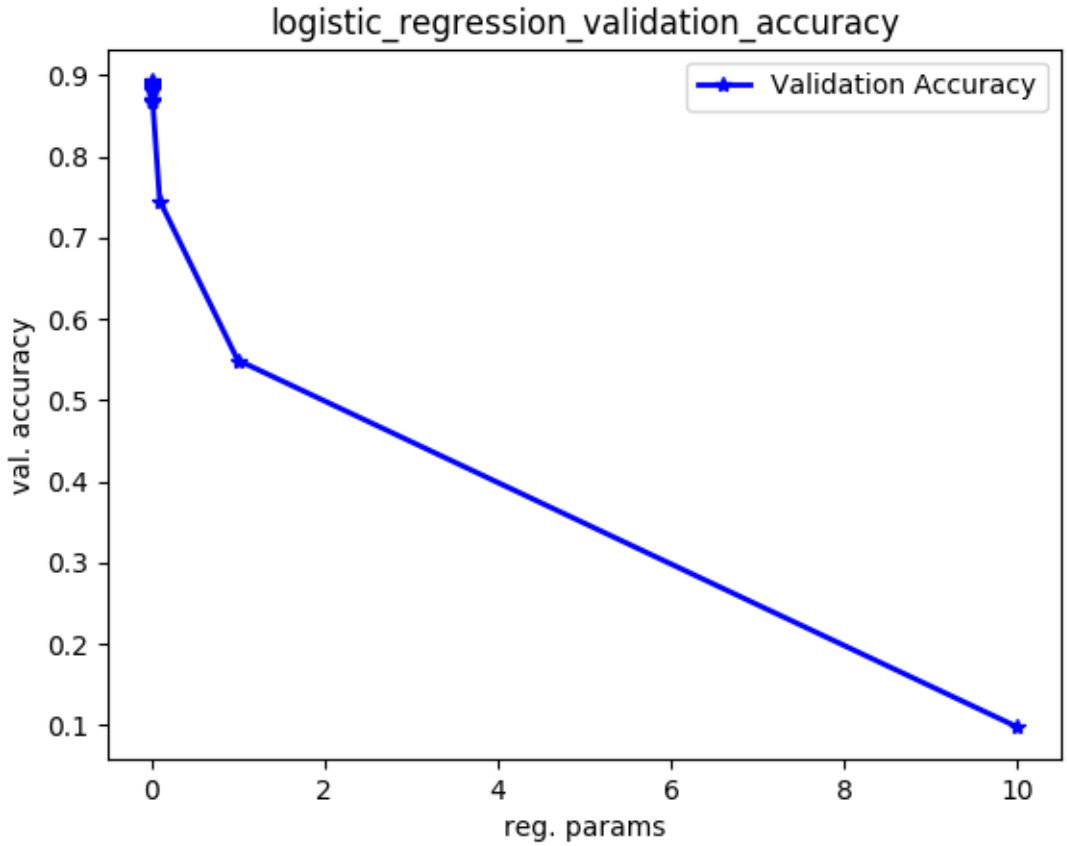
logistic_regression _best _results:

reg. value	validation accuracy	test accuracy
1e-10	0.8930635838150289	0.8930232558139535

1.1.3 Weight Parameter

The 2 vs 7 parameters weighs heigh when the two is present and seven is not. For instance just above the horizontal top line in seven several people probably makes a curly line for the two instead. Also in the right bottom the two is present but seven is not.

It looks like the actual numbers. It makes sense that the algorithms has large weights at the positions of the pixels of the numbers it tries to model. Then the dot product will be large and it will likely be classified as the correct digit.



1.1.4 Best parameters

The best regularization parameter is 1e-10. Validation accuracy is 0.893 and test accuracy is 0.893.

1.2 Theory

1.2.1 Question 1

The running time of logitstic regression is:
Preprocessing: $\mathcal{O}(n)$
Logcost: $\mathcal{O}(nd)$ (assuming naive matrix multiplication)
Gradient: $\mathcal{O}(nd)$ (assuming naive matrix multiplication)

The running time of gradient descent is
Total: $\mathcal{O}(epochs * logcost) = \mathcal{O}(epochs * n * d)$
The running time of mini gradient descent is
Total: $\mathcal{O}(epochs * n/batchsize * batchsize * d) = \mathcal{O}(epochs * nd)$

1.2.2 QUESTION 2

The running itme of logistic regression for two classes are:
$\mathcal{O}(epochs * n * d)$
Since we just have to do one mini gradient descent. For K classes we have to do K gradient descents whichs gives a running time of
$\mathcal{O}(epochs * n * d * K)$

1.2.3 QUESTION 3

The algorithm only looks at individual pixels so as long as the same permutation is done to all the images there should be no difference.

2 PART II: SOFTMAX REGRESSION

2.1 CODE

2.1.1 SUMMARY

We achieved a validation accuracy of 0.903 and test accuracy of 0.776 using regularization parameter of 1e-10. The generalization seems to be very bad for some unknown reason.

2.1.2 RESULTS AND PLOTS

softmax_classification _report

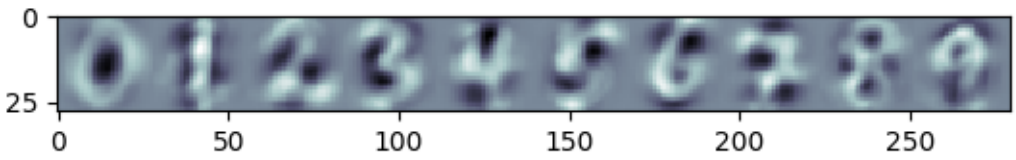
```
Classification Report
             precision    recall  f1-score   support

          0       0.94      0.96      0.95       258
          1       0.89      0.81      0.85       258
          2       0.92      0.82      0.87       258
          3       0.85      0.93      0.89       258
          4       0.79      0.91      0.85       258
          5       0.94      0.80      0.87       258
          6       0.92      0.90      0.91       258
          7       0.83      0.95      0.88       258
          8       0.80      0.86      0.83       258
          9       0.85      0.74      0.79       258

avg / total       0.87      0.87      0.87      2580
```

softmax_confusion _matrix:

0	1	2	3	4	5	6	7	8	9
246	0	1	0	1	0	2	0	7	1
0	223	4	2	4	1	0	1	5	18
0	17	196	9	0	0	3	9	22	2
0	4	0	240	0	2	1	3	6	2
1	3	0	0	219	3	3	2	2	25
4	1	0	11	1	206	3	1	28	3
7	3	0	0	10	3	228	0	7	0
0	3	4	6	0	1	0	225	6	13
3	5	1	3	1	0	0	1	238	6
0	14	0	11	5	1	0	1	9	217

softmax_accuracy:

```
train_accuracy,test_accuracy
0.8713872832369942,0.8674418604651163
```



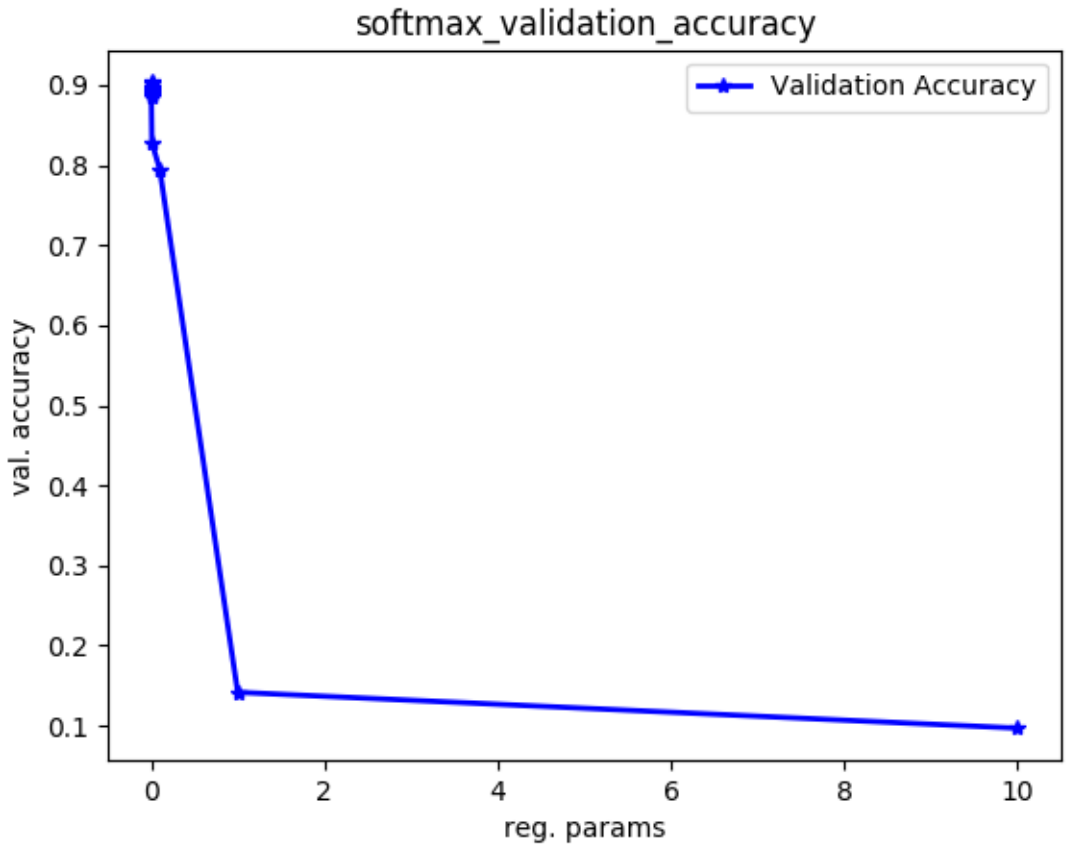
softmax_validation _scores:

reg. params	val. score
1e-15	0.9022157996146436
1e-14	0.8945086705202312
1e-13	0.8959537572254336
1e-12	0.8868015414258189
1e-11	0.8896917148362236
1e-10	0.9036608863198459
1e-09	0.8858381502890174
1e-08	0.8916184971098265
1e-07	0.8892100192678227
1e-06	0.8940269749518305
1e-05	0.8896917148362236
0.001	0.896917148362235
0.01	0.8280346820809249
0.1	0.7947976878612717
1.0	0.1416184971098266
10.0	0.0968208092485549

softmax_best _results:

reg. value	validation accuracy	test accuracy
1e-10	0.9036608863198459	0.7763565891472868

‘



2.1.3 Weight parameters

Looks identical to the parameters for the logistic regression.

2.1.4 Best Parameters

Best regularization is 1e-10 which gives an validation accuracy of 0.903 and test accuracy of 0.776.

2.2 THEORY

2.2.1 QUESTION 1

The running time of softmax is:
Softcost: $\mathcal{O}(ndK)$ (assuming naive matrix multiplication)
Gradient: $\mathcal{O}(ndK)$ (assuming naive matrix multiplication)

The running time of mini gradient descent is
Total: $\mathcal{O}(epochs * n/batchsize * batchsize * d * K) = \mathcal{O}(epochs * ndK)$
Softmax is done by running mini gradient descent one time which makes the above time complexity the time complexity of softmax.

3 PART III: SOFTMAX VS LOGISTIC REGRESSION

The logistic regression model turned out to have a better generalization to the test set, but achieved a slightly larger in-sample error. We expect the two methods to be comparable in performance since both have (d + 1) x K parameters.