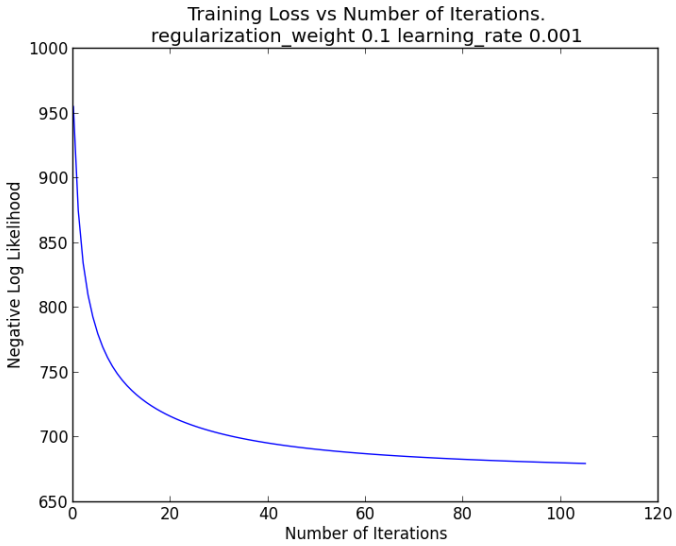
CS189 - HW4
Jeffrey Tsui
Justin Nguyen

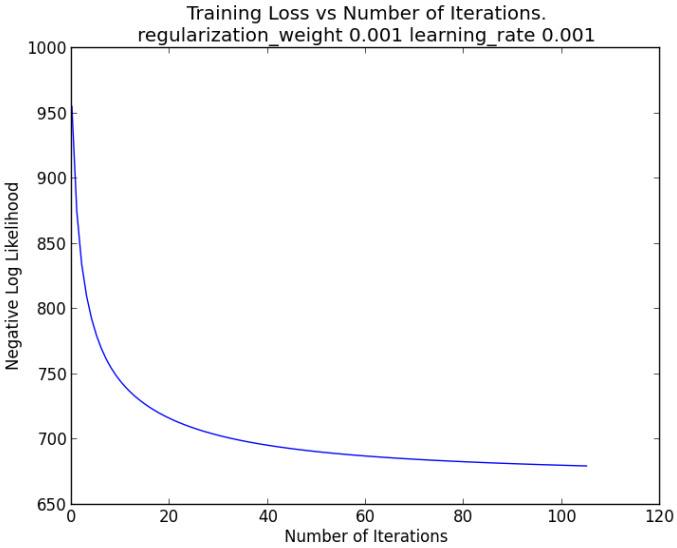
1.

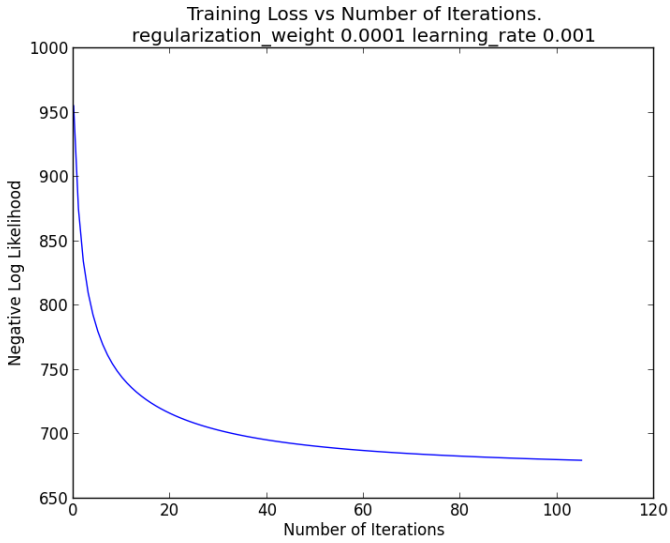
i) Standardized columns

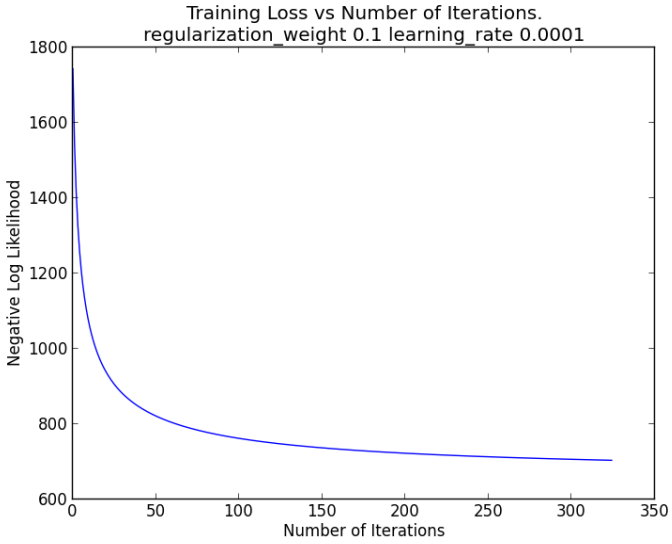
threshold = 0.1

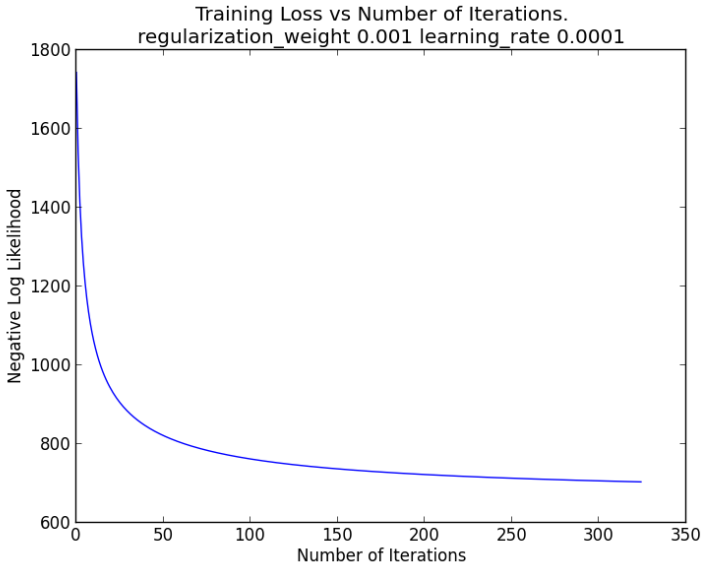
regularization weight	learning rate	training error	test error
0.1	0.001	0.0799347471452	0.0911458333333
0.1	0.0001	0.078967934323	0.0846322917867
0.001	0.001	0.0802610114192	0.0930989583333
0.001	0.0001	0.078955954323	0.0827832926667
0.0001	0.001	0.0802610114192	0.0930989583333
0.0001	0.0001	0.078955954323	0.0826822916667

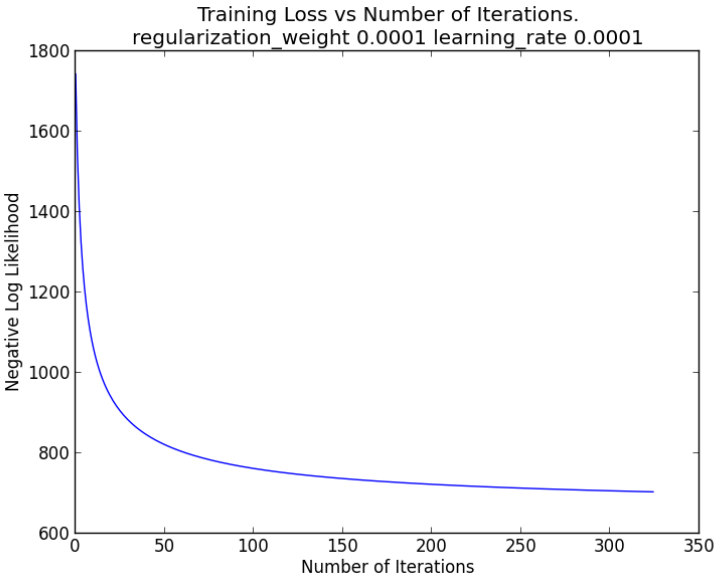

Training Loss vs Number of Iterations.
regularization_weight 0.1 learning_rate 0.001
Negative Log Likelihood
Number of Iterations
1000
950
900
850
800
750
700
650
0
20
40
60
80
100
120


Training Loss vs Number of Iterations.
regularization_weight 0.001 learning_rate 0.001
Negative Log Likelihood
Number of Iterations
1000
950
900
850
800
750
700
650
0
20
40
60
80
100
120


Training Loss vs Number of Iterations.
regularization_weight 0.0001 learning_rate 0.001
Negative Log Likelihood
Number of Iterations
1000
950
900
850
800
750
700
650
0
20
40
60
80
100
120



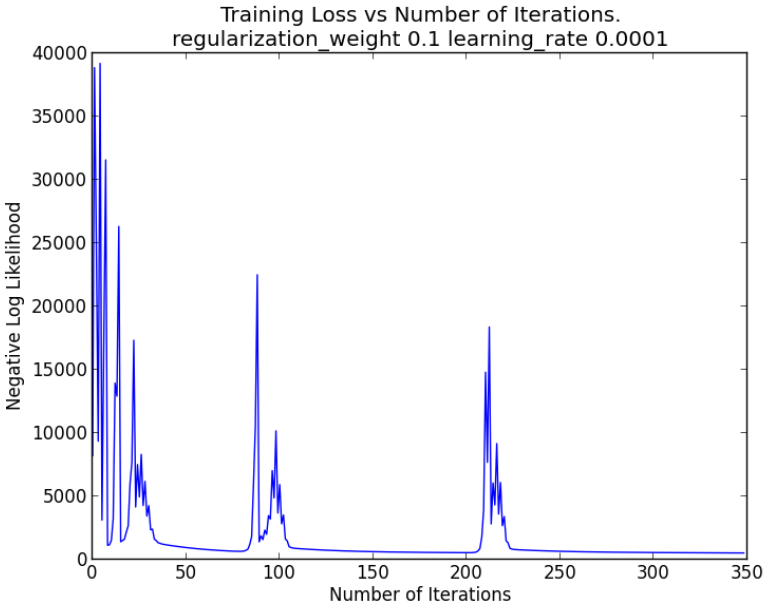


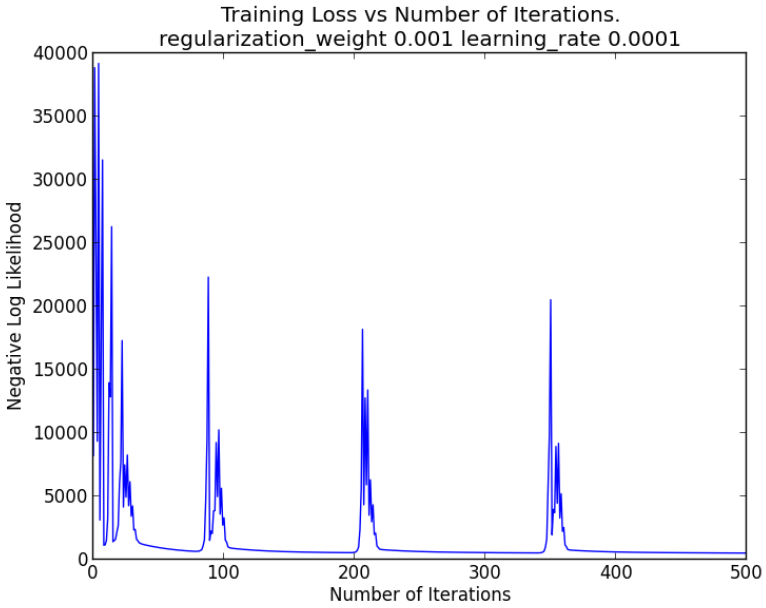


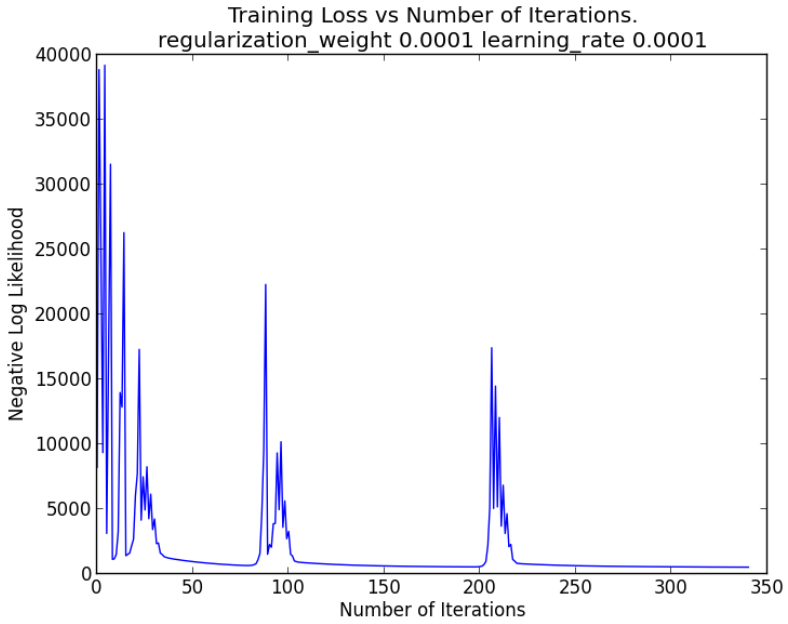
ii) Log transform

threshold = 0.1

regularization weight	learning rate	training error	test error
0.1	0.0001	0.0531810766721	0.060546875
0.001	0.0001	0.0515497553018	0.056640625
0.0001	0.0001	0.0531810766721	0.060546875



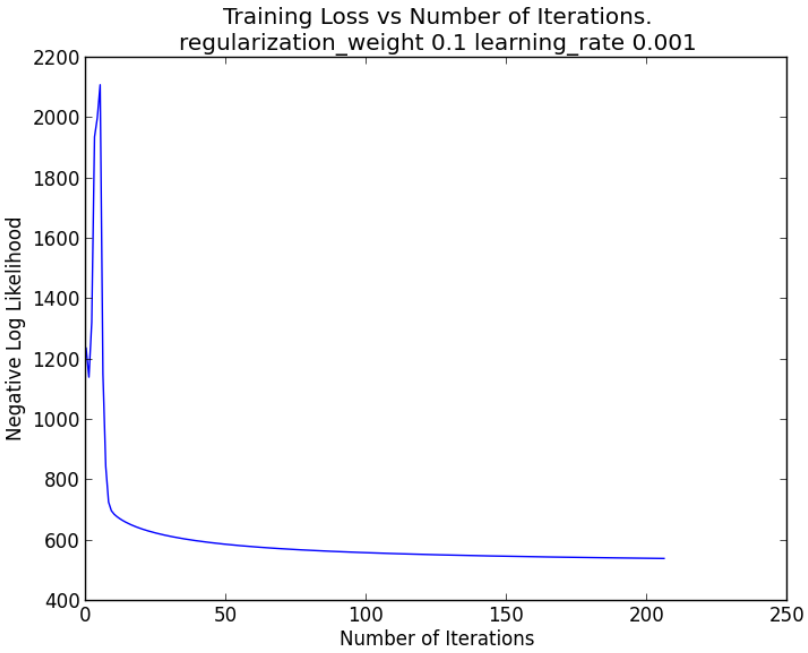


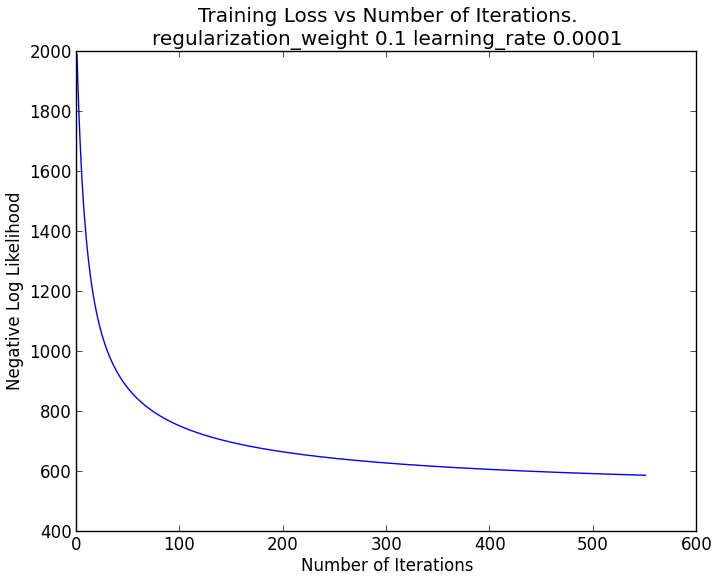


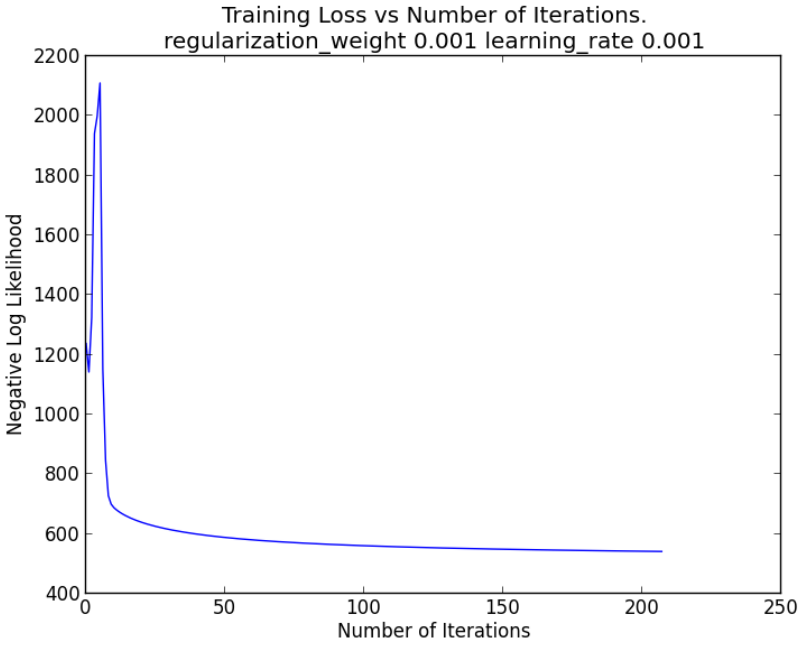
iii) Binarized

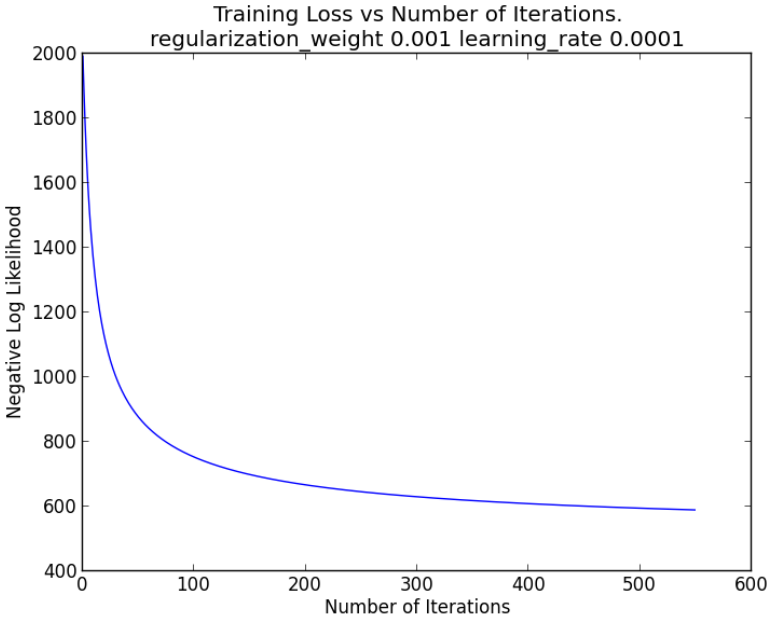
threshold = 0.1

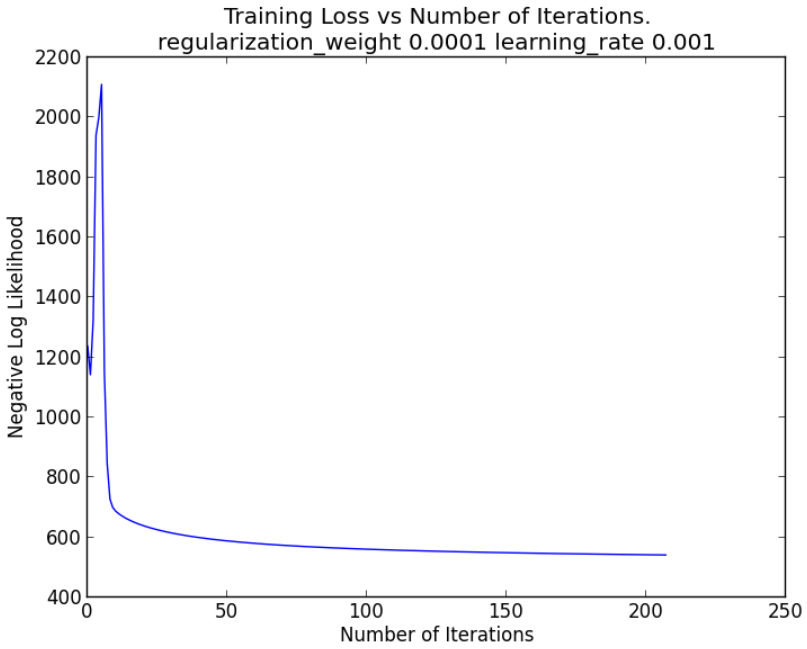
regularization weight	learning rate	training error	test error
0.1	0.001	0.0655791190865	0.0748697916667
0.1	0.0001	0.0662316476346	0.076171875
0.001	0.001	0.0639477977162	0.0748697916667
0.001	0.0001	0.0662316476346	0.0755208333333
0.0001	0.001	0.0649477977162	0.0750697916667
0.0001	0.0001	0.0662316476346	0.0755208333333

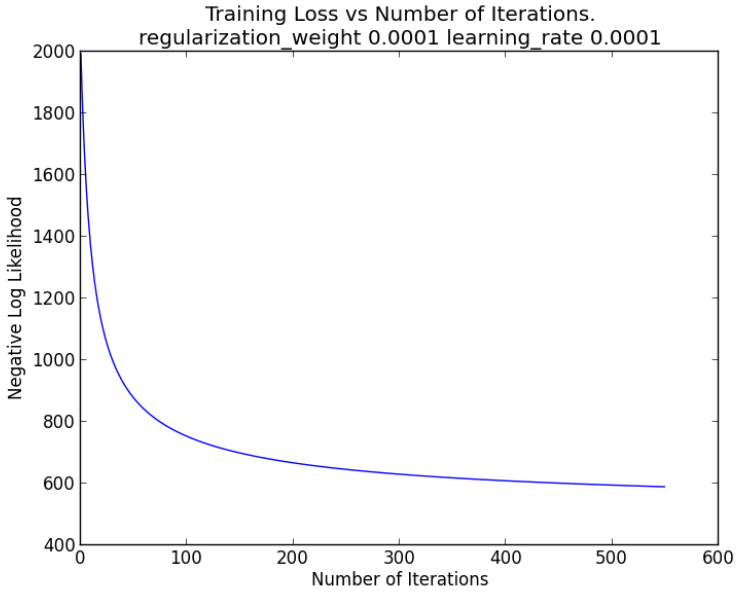

Training Loss vs Number of Iterations.
regularization_weight 0.1 learning_rate 0.001
Negative Log Likelihood
Number of Iterations
2200
2000
1800
1600
1400
1200
1000
800
600
400
0
50
100
150
200
250


Training Loss vs Number of Iterations.
regularization_weight 0.1 learning_rate 0.0001
Negative Log Likelihood
Number of Iterations
2000
1800
1600
1400
1200
1000
800
600
400
0
100
200
300
400
500
600


Training Loss vs Number of Iterations.
regularization_weight 0.001 learning_rate 0.001
Negative Log Likelihood
Number of Iterations
2200
2000
1800
1600
1400
1200
1000
800
600
400
0
50
100
150
200
250


Training Loss vs Number of Iterations.
regularization_weight 0.001 learning_rate 0.0001
Negative Log Likelihood
Number of Iterations
2000
1800
1600
1400
1200
1000
800
600
400
0
100
200
300
400
500
600





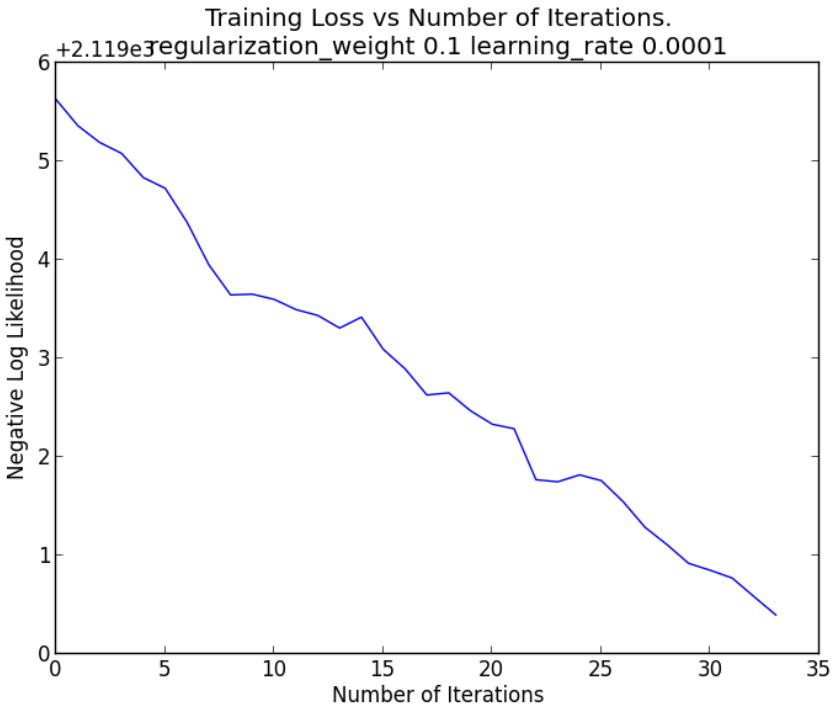
2.

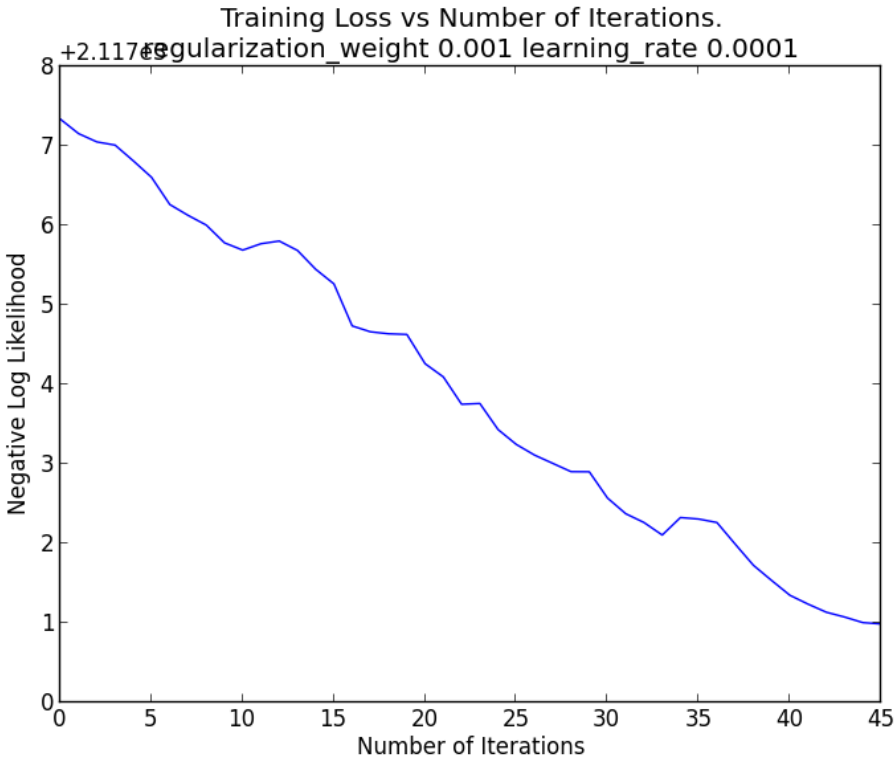
i) Standardized columns

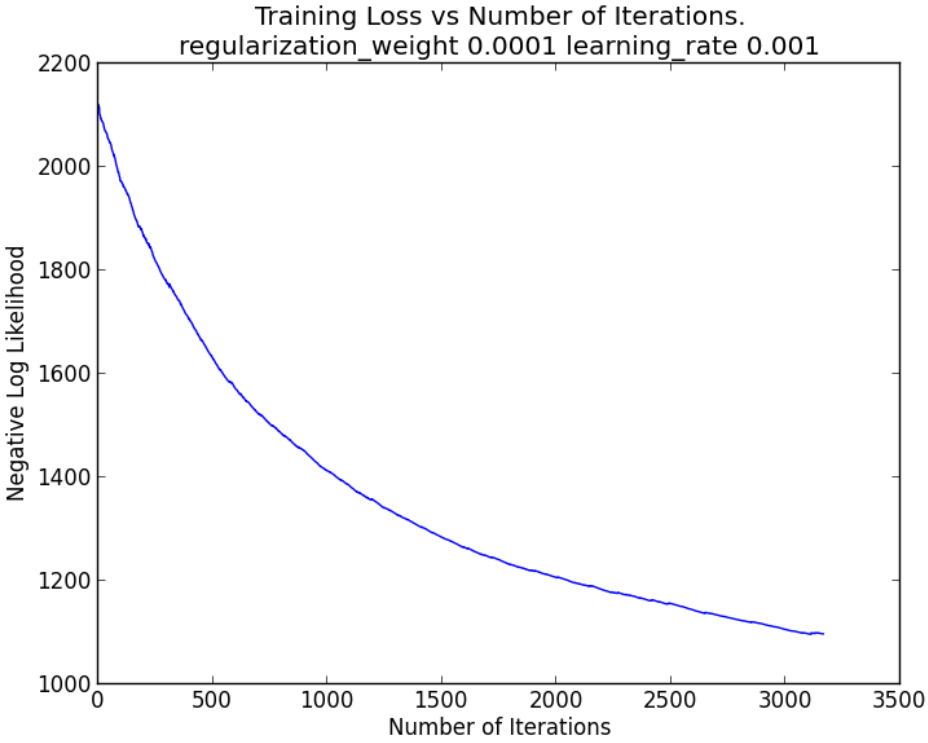
threshold = 0.001

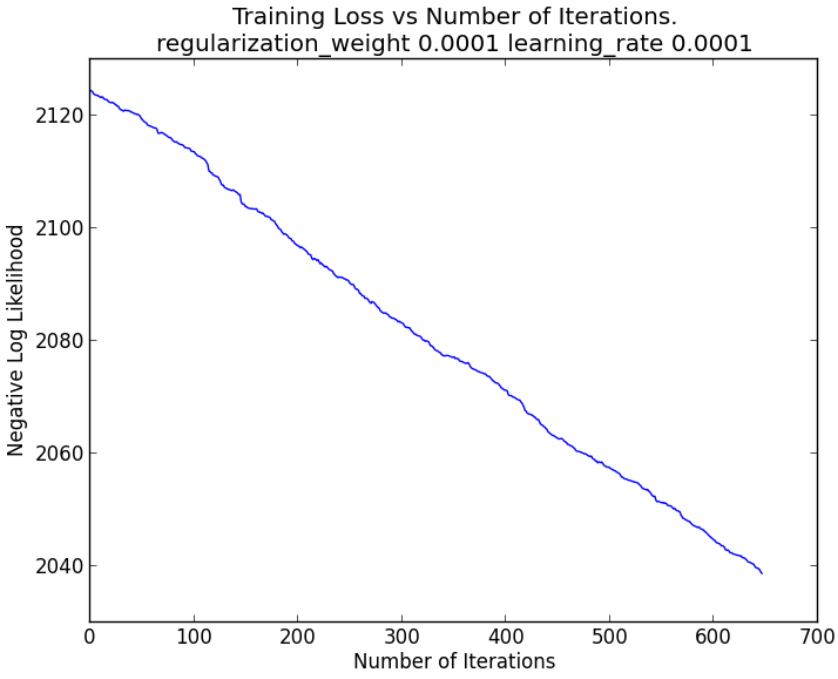
regularization weight	learning rate	training error	test error
0.1	0.001	0.179771615008	0.202473958333
0.1	0.0001	0.156280587276	0.166015625
0.001	0.001	0.0972267536705	0.106770833333

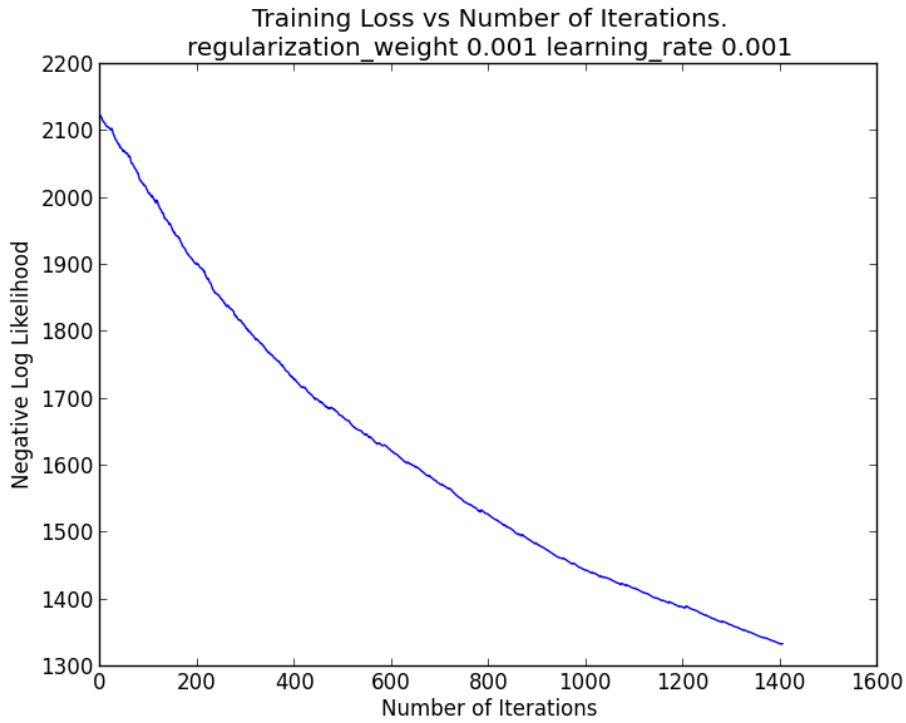
0.001	0.0001	0.122022838499	0.142578125
0.0001	0.001	0.0920065252855	0.106119791667
0.0001	0.0001	0.099836867863	0.10546875

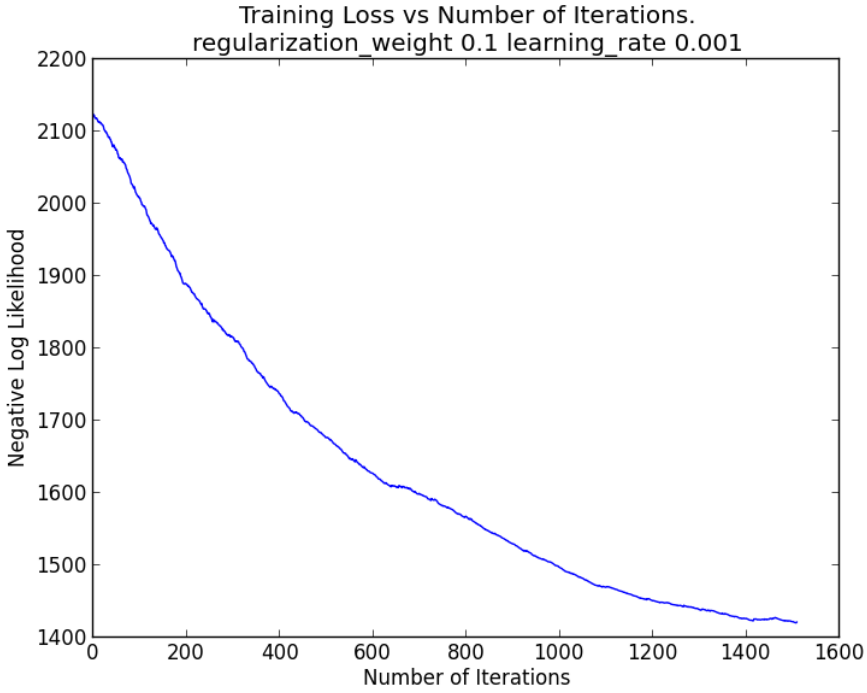





Training Loss vs Number of Iterations.
regularization_weight 0.0001 learning_rate 0.001
Negative Log Likelihood
Number of Iterations
2200
2000
1800
1600
1400
1200
1000
0
500
1000
1500
2000
2500
3000
3500


Training Loss vs Number of Iterations.
regularization_weight 0.0001 learning_rate 0.0001
Negative Log Likelihood
Number of Iterations
2120
2100
2080
2060
2040
0
100
200
300
400
500
600
700



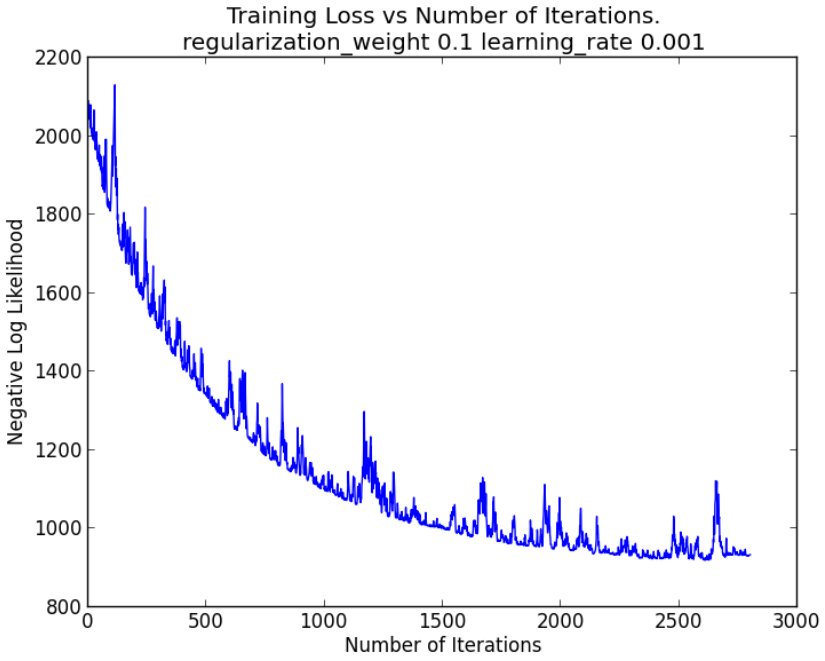


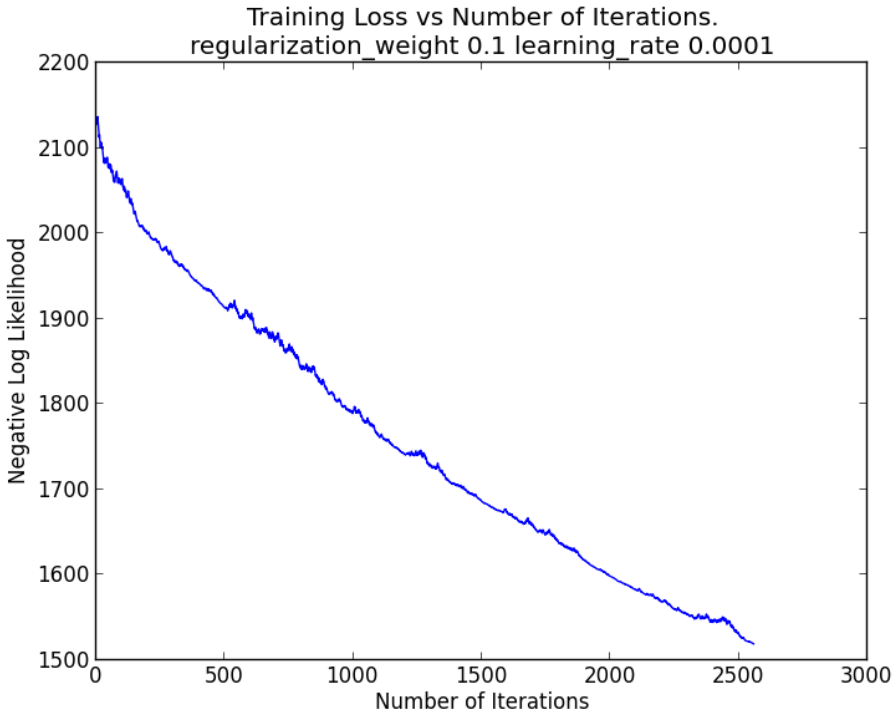
ii) Log transform

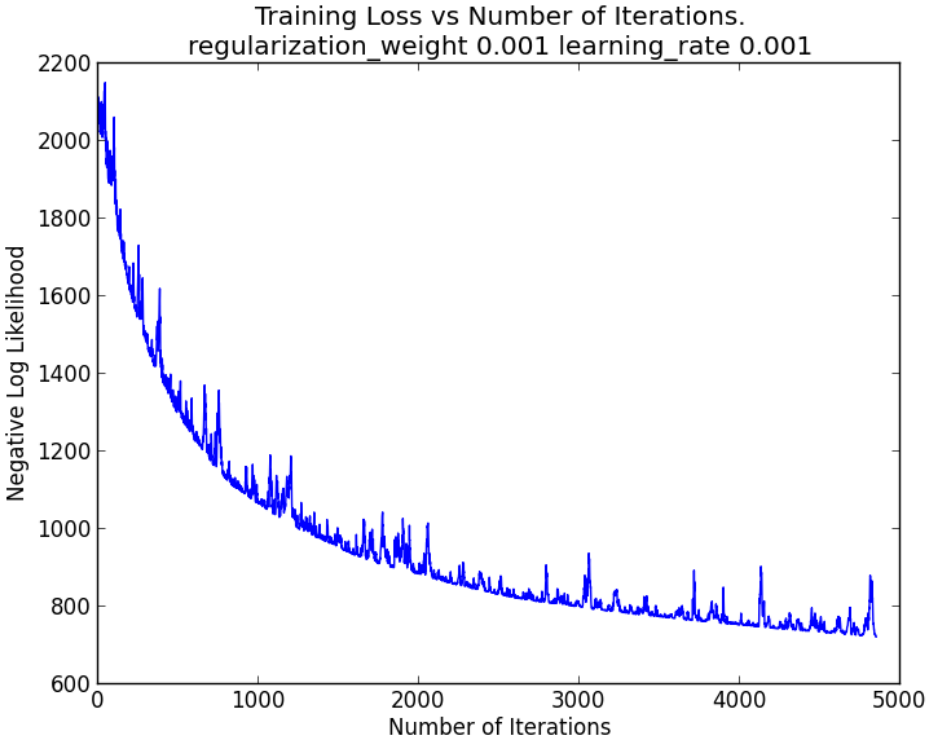
threshold = 0.001

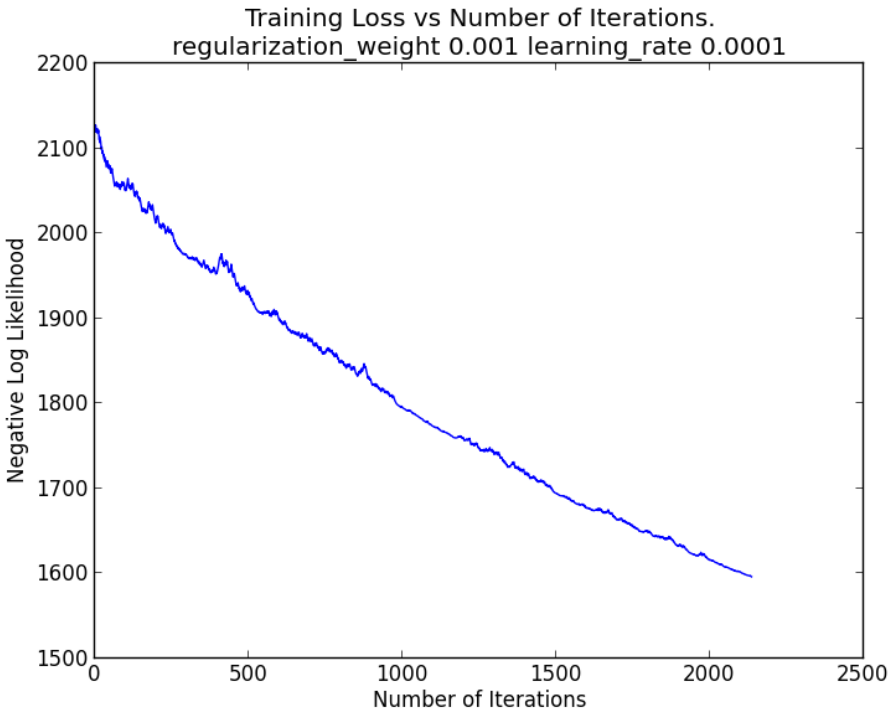
regularization weight	learning rate	training error	test error
0.1	0.001	0.0867862969005	0.0970052083333
0.1	0.0001	0.15823817292	0.15234375
0.001	0.001	0.0792822185971	0.08984375

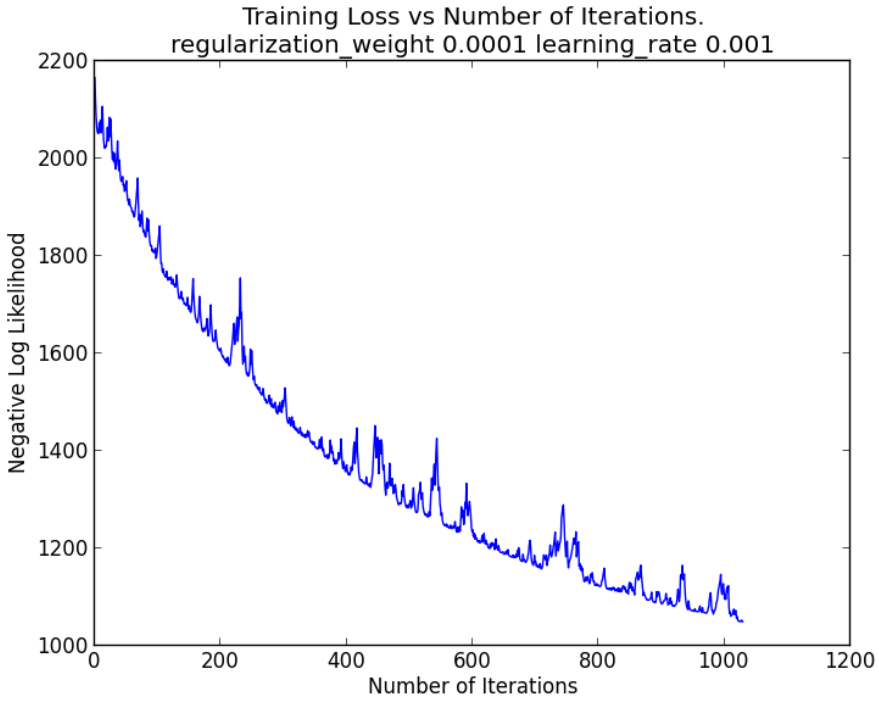
0.001	0.0001	0.187601957586	0.179036458333
0.0001	0.001	0.10277324633	0.110677083333
0.0001	0.0001	0.206525285481	0.192708333333

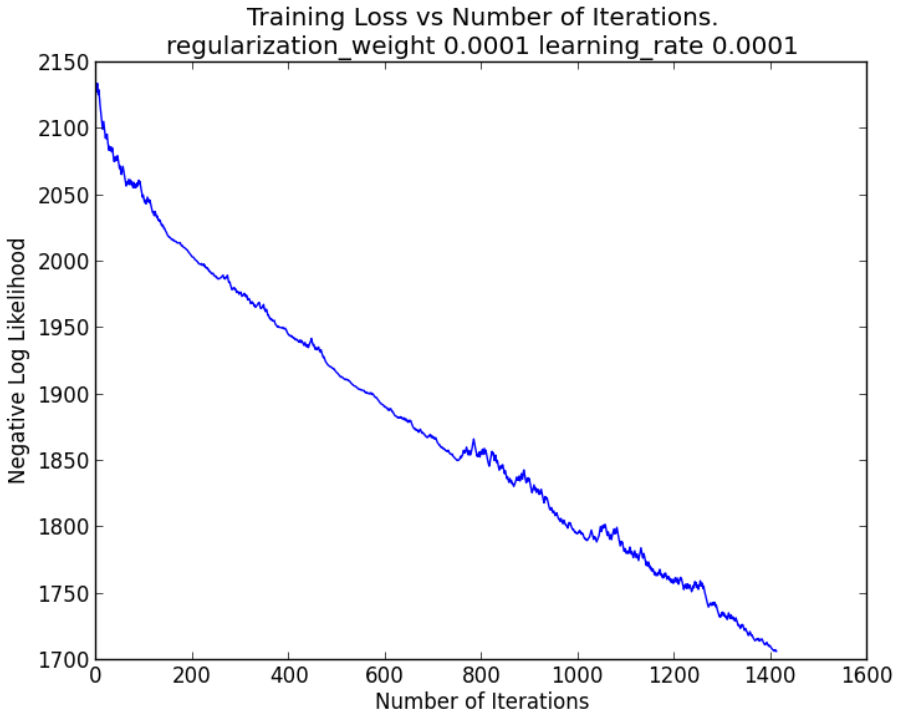





Training Loss vs Number of Iterations.
regularization_weight 0.001 learning_rate 0.001
Negative Log Likelihood
Number of Iterations
2200
2000
1800
1600
1400
1200
1000
800
600
0
1000
2000
3000
4000
5000


Training Loss vs Number of Iterations.
regularization_weight 0.001 learning_rate 0.0001
Negative Log Likelihood
Number of Iterations
2200
2100
2000
1900
1800
1700
1600
1500
0
500
1000
1500
2000
2500



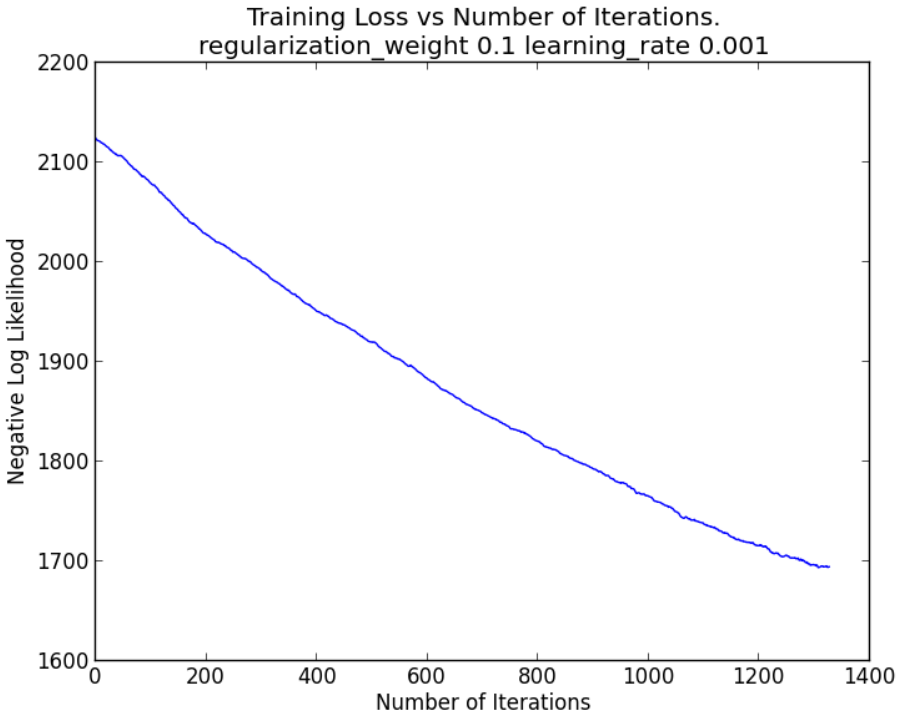


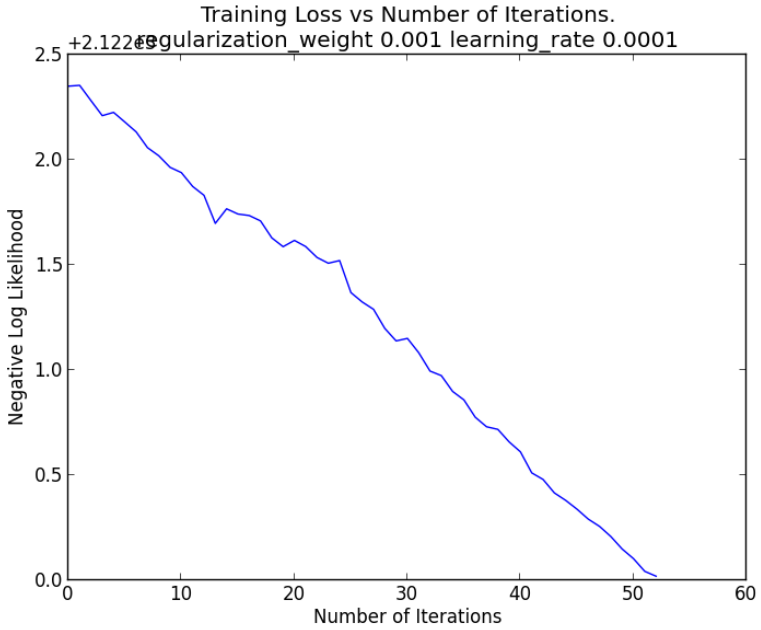
iii) Binarized

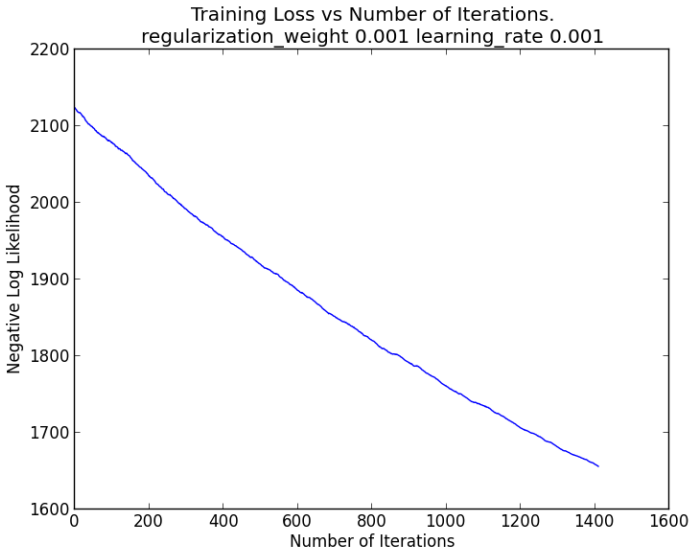
threshold = 0.001

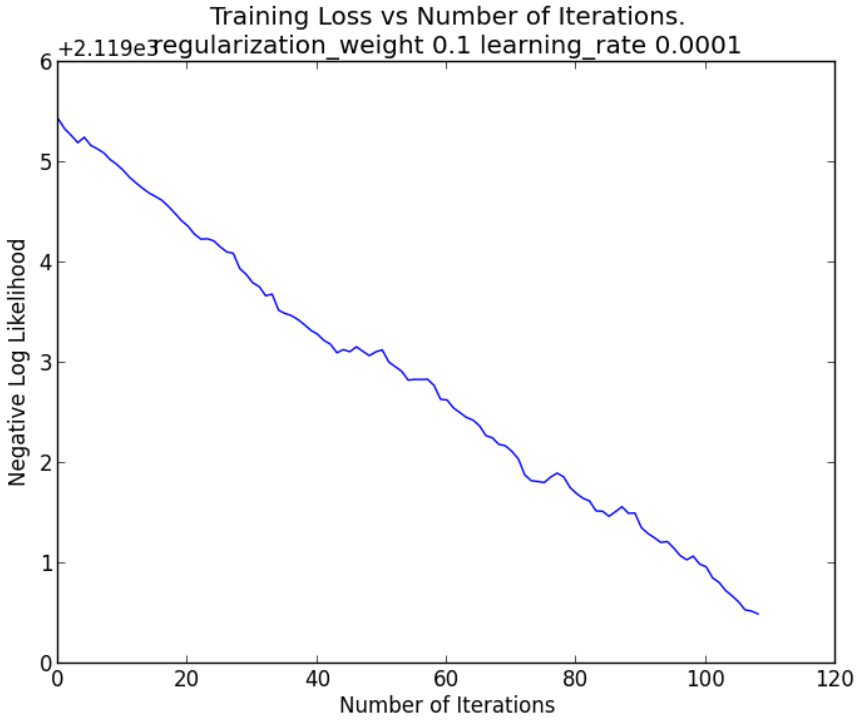
regularization weight	learning rate	training error	test error
0.1	0.001	0.203588907015	0.204427083333
0.1	0.0001	0.119738988581	0.117838541667
0.001	0.001	0.122022838499	0.119140625

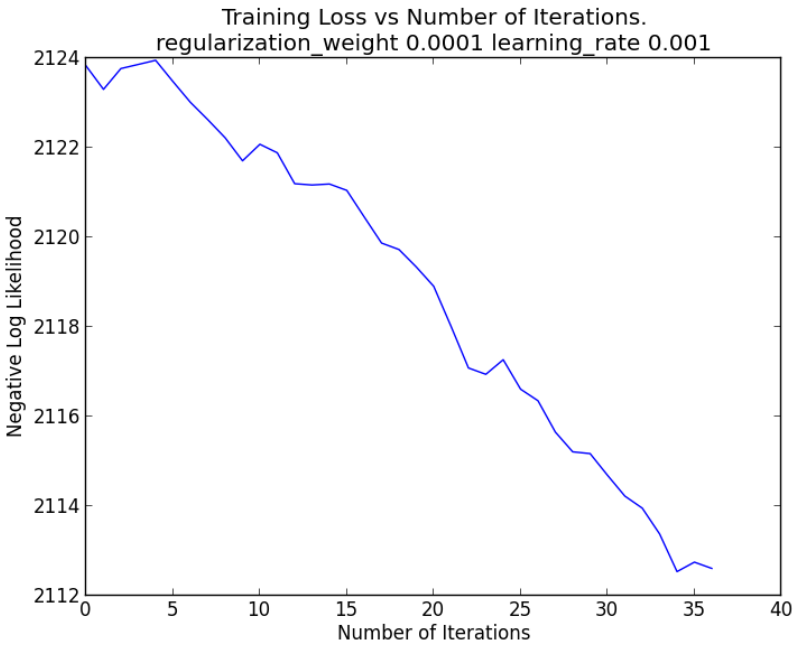
0.001	0.0001	0.160195758564	0.159505208333
0.0001	0.001	0.393148450245	0.384765625
0.0001	0.0001	0.111582381729	0.116536458333

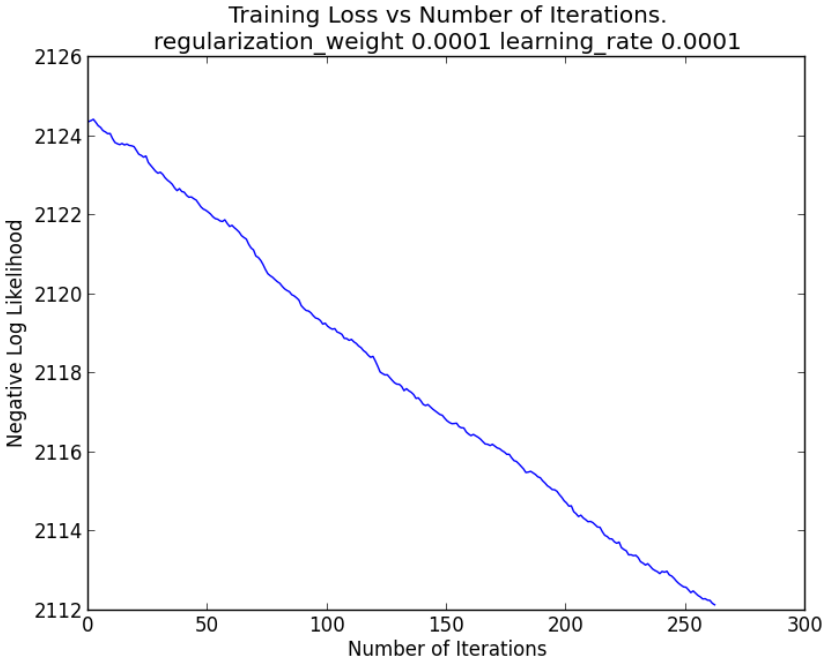





Training Loss vs Number of Iterations.
regularization_weight 0.001 learning_rate 0.001
Negative Log Likelihood
Number of Iterations
2200
2100
2000
1900
1800
1700
1600
0
200
400
600
800
1000
1200
1400
1600


Training Loss vs Number of Iterations.
regularization_weight 0.1 learning_rate 0.0001
+2.119e3
Negative Log Likelihood
Number of Iterations
6
5
4
3
2
1
0
0
20
40
60
80
100
120





3.

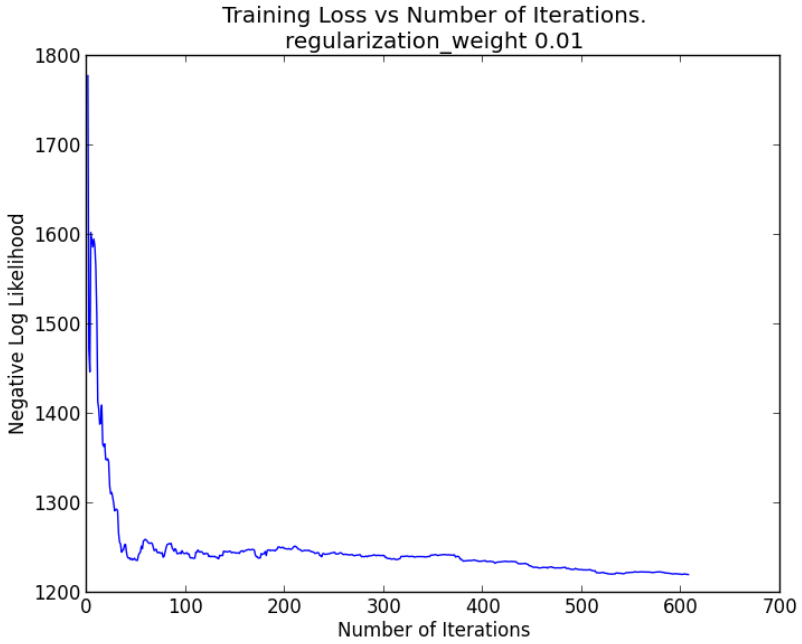
Other strategies, besides setting learning rate to 1/x for xth iteration that might work well:

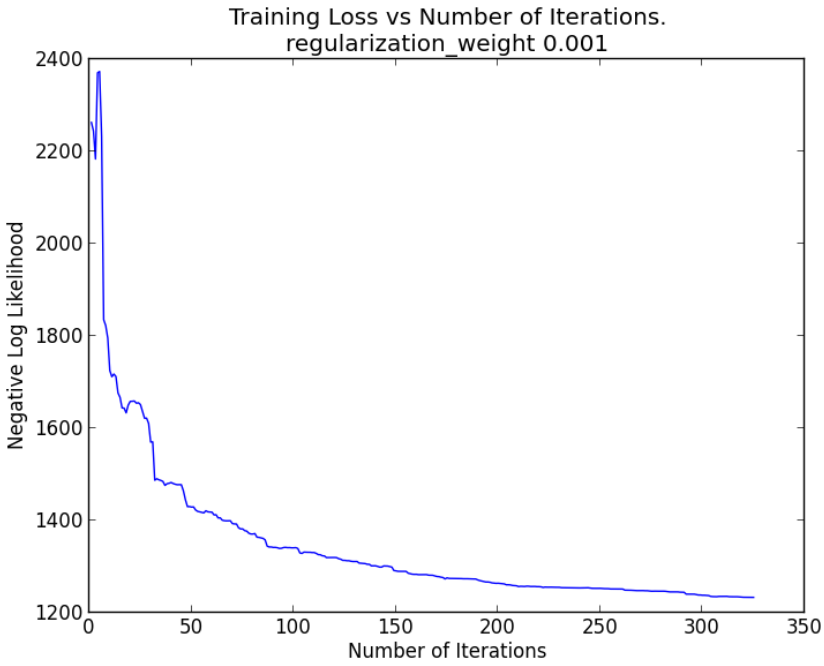
1. Update the learning rate alpha = alpha/(1 + t/T) where t is the number of iterations so far and T is a constant that can be tuned. This way the more iterations have passed, the lower the learning rate becomes allowing for faster convergence.
2. We could use momentum: we could add a fraction of the previous weight update to the current weight update. In this way, if we keep calculating gradients that point in the same direction, we'll move in that direction faster. If we encounter oscillating gradients, the momentum will help smooth out the variations.

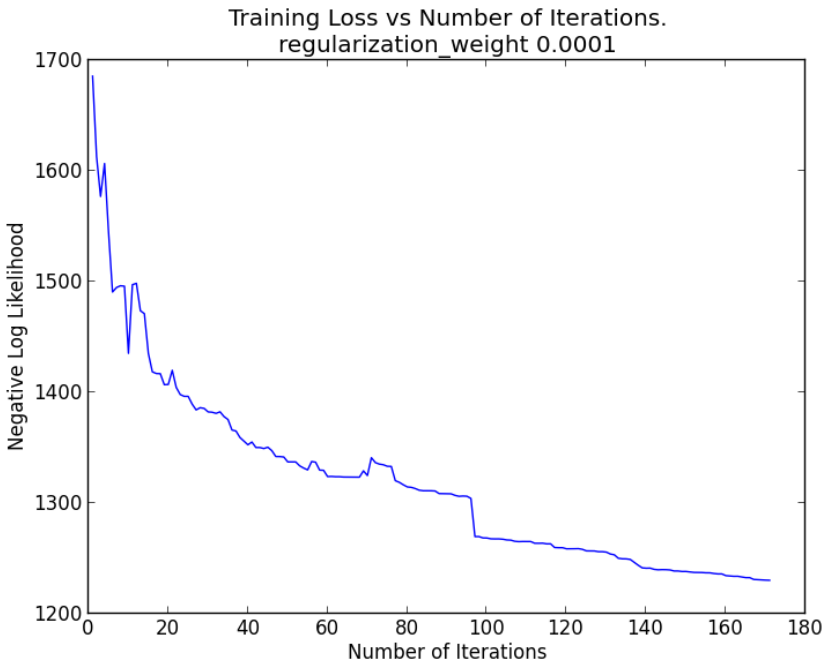
i) Standardized columns

threshold = 0.001

regularization weight	training error	test error
0.01	0.13572593801	0.150390625
0.001	0.140293637847	0.172526041667
0.0001	0.157911908646	0.170572916667



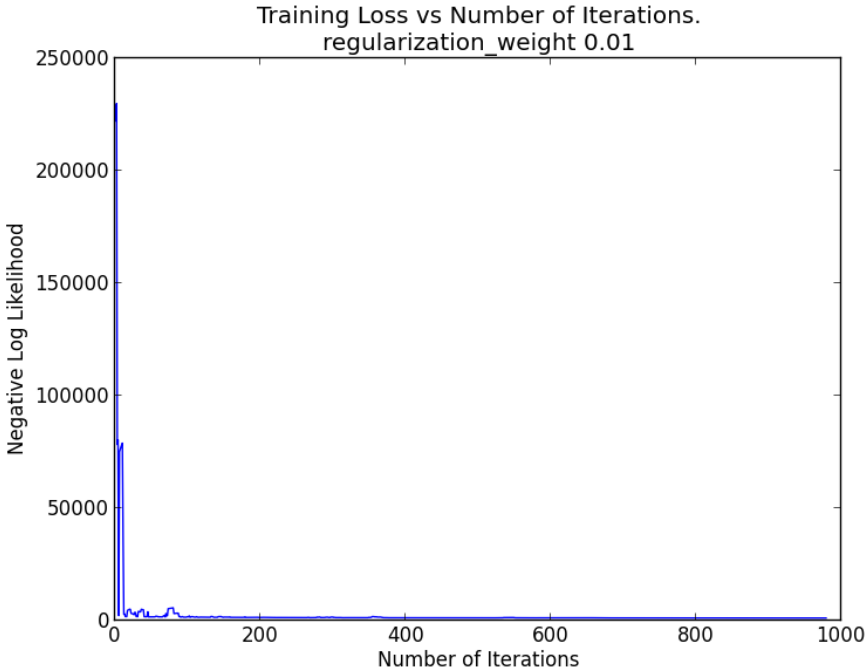


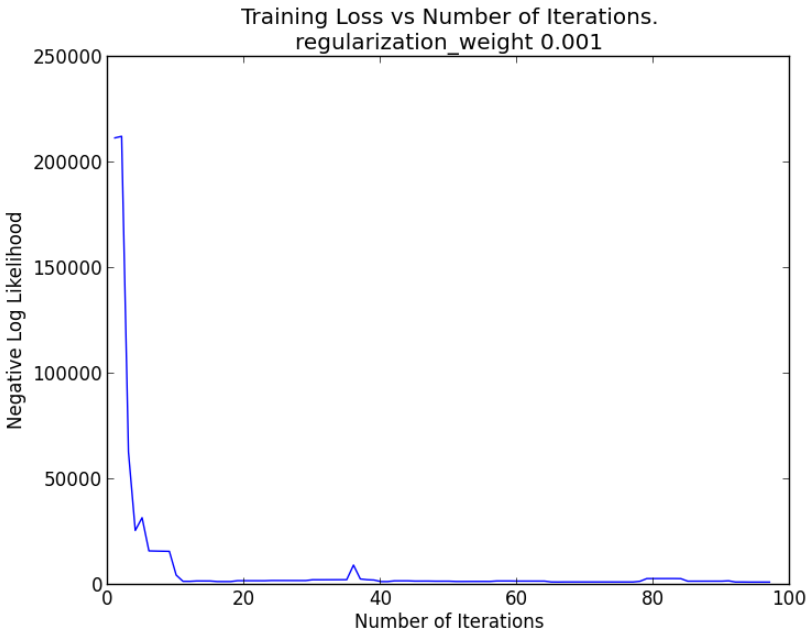


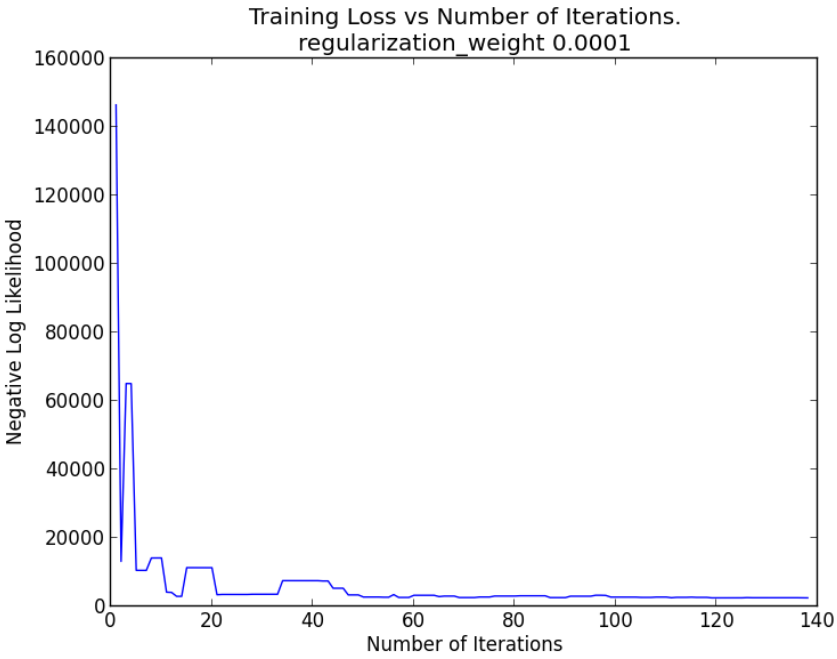
ii) Log transform

threshold = 0.001

regularization weight	training error	test error
0.01	0.142903752039	0.143229166667
0.001	0.122349102773	0.140625
0.0001	0.152365415987	0.16796875



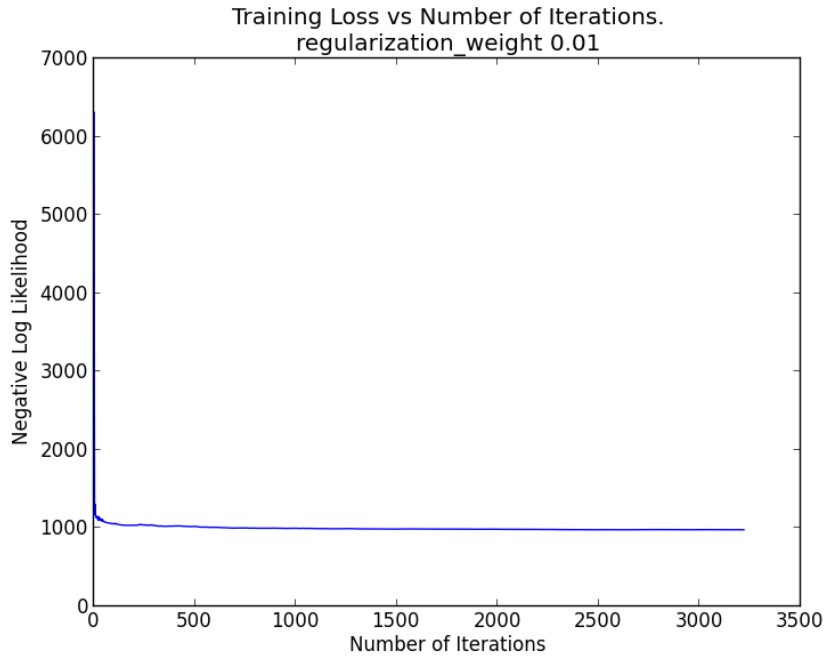


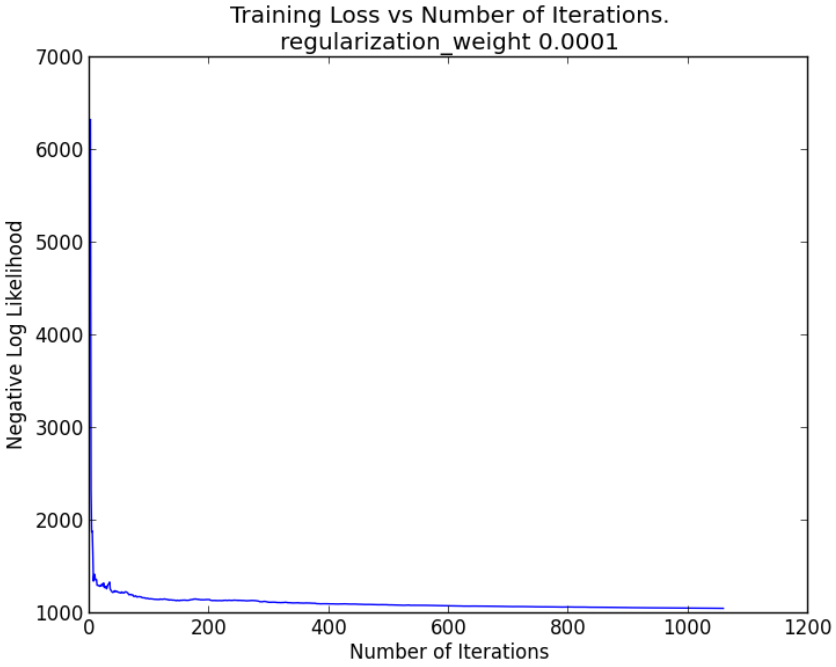


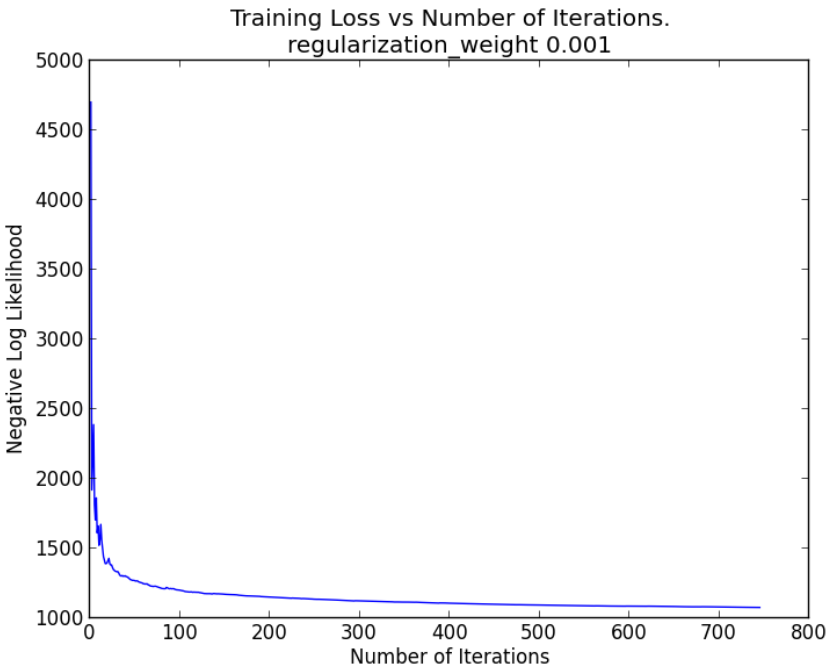
iii) Binarized

threshold = 0.001

regularization weight	training error	test error
0.01	0.11647634584	0.1171875
0.001	0.108972267537	0.110026041667
0.0001	0.123001631321	0.11328125


Training Loss vs Number of Iterations.
regularization_weight 0.01
Negative Log Likelihood
Number of Iterations
0
1000
2000
3000
4000
5000
6000
7000
0
500
1000
1500
2000
2500
3000
3500


Training Loss vs Number of Iterations.
regularization_weight 0.0001
Negative Log Likelihood
Number of Iterations
1000
2000
3000
4000
5000
6000
7000
0
200
400
600
800
1000
1200

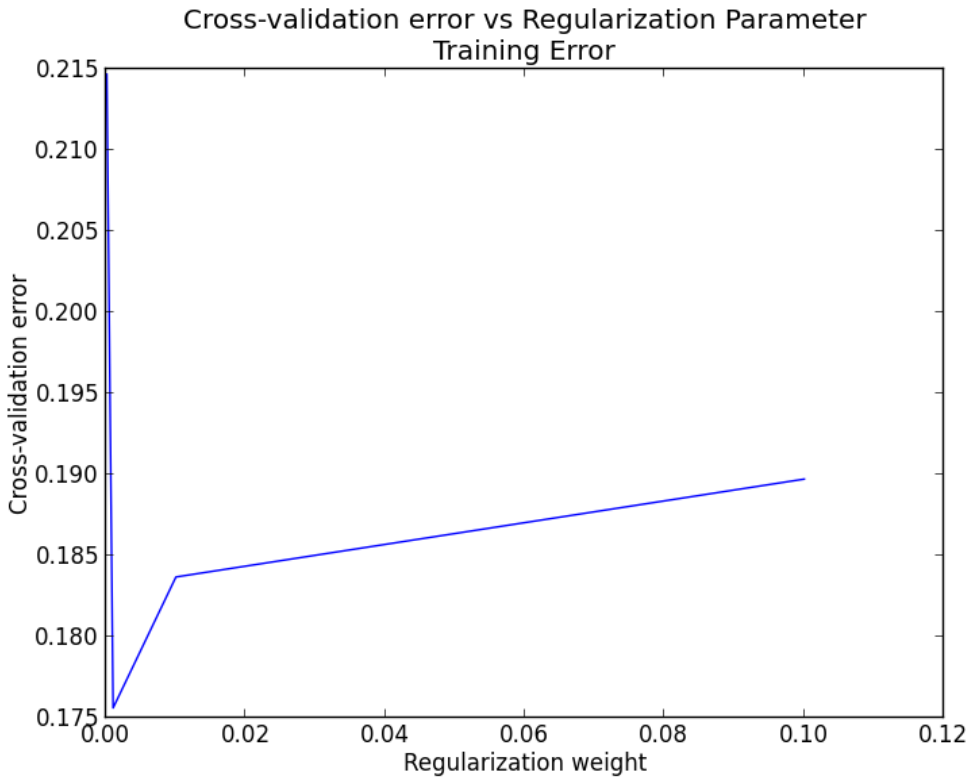


4.

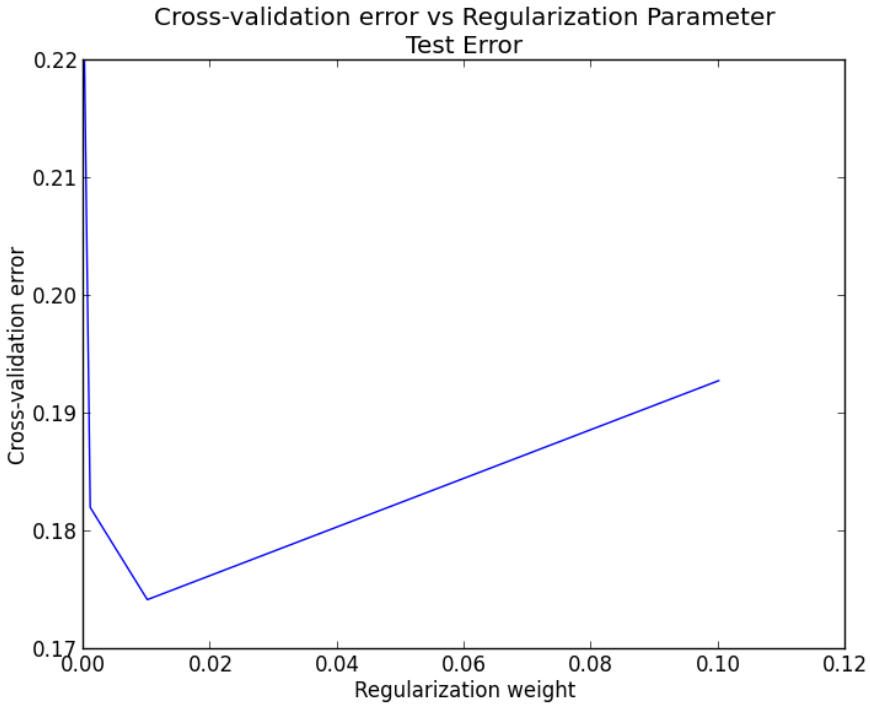
threshold = 0.0001, learning rate = 0.0001

regularization weight	cross validation training error	cross validation test error
0.1	0.189722675367	0.192822185971
0.01	0.183686786297	0.174225122349
0.001	0.175611745514	0.182055464927
0.0001	0.214681892333	0.219902120718

Using threshold = 0.0001, learning rate = 0.0001, regularization weight = 0.01, we got a error rate on the test set = 0.115885416667


Cross-validation error vs Regularization Parameter
Training Error
Cross-validation error
Regularization weight
0.215
0.210
0.205
0.200
0.195
0.190
0.185
0.180
0.175
0.00
0.02
0.04
0.06
0.08
0.10
0.12

\


Cross-validation error vs Regularization Parameter
Test Error
Cross-validation error
Regularization weight
0.22
0.21
0.20
0.19
0.18
0.17
0.00
0.02
0.04
0.06
0.08
0.10
0.12