Brianna Shade
CS545: Machine Learning
1/23/14

Project 1: Perceptron Training Distinguishing Letters

OVERVIEW

This project explored training perceptrons to recognize handwritten alpha characters, specifically distinguishing **A**s from the other letters of the alphabet. A text file consisting of 16 feature values for several hundred instances of each letter was used to train and test the perceptrons, calculating final accuracy statistics.

IMPLEMENTATION

This project was coded in Java 1.7.0_09, using Eclipse on a MacBook Pro running Mavericks. All data points for both the target letter and test letter are first extracted from the input file, alternatingly populating a designated training set and a test set (approximately half of the input data used for training and half for testing). The training data features are then scaled to between 0 and 1, and the sgn value is calculated for each instance, beginning with randomly generated weights and adjusting them stochastically after each training example. The entire training set is processed repeatedly until the change in weights converge to below a specified threshold. Finally, the test data is processed with the learned weights, calculating and outputting the perceptron's accuracy, precision, recall, and confusion matrix. Data points for an ROC graph are also calculated, slicing the range between everything classified as positive and everything classified as negative according to a provided value.

I first ran some tests to determine what would be the optimal weight delta convergence threshold. I tested only **A** vs **B** to get a rough idea for thresholds, finding 0.0000001 to produce the best accuracy. However, using this threshold revealed many of the other letters' weight changes never seemed to converge to below this value, so I also implemented a cap of 500 epochs. Adjusting this threshold, I found a balance between overall accuracy of all letters at about 0.000001.

The program can be compiled with *javac Perceptrons.java*, with a CLASSPATH environment variable set to the project's bin folder. The program can then be executed with *java hw1Perceptrons.Perceptrons input_file.data learning_rate*, where *learning_rate* is a double type decimal value.

OBSERVATIONS

With a learning rate of 0.2, most letters fall in the 98-99% accuracy range, with **H** and **O** closer to 95-97%. **A** tends to get confused the most with **H** and **O**. In particular, these letters were not commonly classified as **A**s, but rather **A**s were incorrectly classified as **H**s or **O**s. **H** would be understandable, as this is likely the most similar letter to **A** when written. However, **O** surprises me, as I wouldn't suspect these to be easily confused. This could depend entirely on the style of a written **A**. If it is cursive or lower-case, it would, indeed, look very much like an **O**.

Letters such as **D**, **H**, **O**, **Q**, and **U** had the lowest recall values (highest false negative rates). Not surprising that these are similar in shape to **O** and **H**, with which the perceptrons most commonly confuse **A**. Conversely, **J**, **K**, and **S** have the lowest precision rates (highest number of false positives). These have similar qualities to **A**, depending on how they are written.

RESULTS

Learning rate of 0.2

Across different perceptrons, the ROC curves (see second document) did not vary greatly. Instead, they all followed the same basic shape, with most having one metric at 100% or 0% for much of the curve.

A vs B (74 epochs)
TP-400, FP-1, FN-4, TN-372
Acc: 99.36%, Precision: 99.75%, Recall: 99.01%

A vs C (500 epochs)
TP-388, FP-0, FN-9, TN-365
Acc: 98.82%, Precision: 100%, Recall: 97.73%

A vs D (200 epochs)
TP-393, FP-7, FN-4, TN-393
Acc: 98.62%, Precision: 98.25%, Recall: 98.99%

A vs E (19 epochs)
TP-393, FP-2, FN-0, TN-383
Acc: 99.74%, Precision: 99.49%, Recall: 100%

A vs F (33 epochs)
TP-395, FP-5, FN-2, TN-380
Acc: 99.1%, Precision: 98.75%, Recall: 99.5%

A vs G (500 epochs)
TP-393, FP-4, FN-7, TN-377
Acc: 98.59%, Precision: 98.99%, Recall: 98.25%

A vs H (500 epochs)
TP-395, FP-21, FN-2, TN-343
Acc: 96.98%, Precision: 94.95%, Recall: 99.5%

A vs I (500 epochs)
TP-391, FP-3, FN-2, TN-376
Acc: 99.35%, Precision: 99.24%, Recall: 99.49%

A vs J (500 epochs)
TP-384, FP-13, FN-4, TN-367
Acc: 97.79%, Precision: 96.73%, Recall: 98.97%

A vs K (500 epochs)
TP-397, FP-12, FN-3, TN-352
Acc: 98.04%, Precision: 97.07%, Recall: 99.25%

A vs L (197 epochs)
TP-400, FP-1, FN-7, TN-367
Acc: 98.97%, Precision: 99.75%, Recall: 98.28%

A vs M (500 epochs)
TP-379, FP-2, FN-7, TN-402
Acc: 98.86%, Precision: 99.48%, Recall: 98.19%

A vs N (116 epochs)
TP-391, FP-4, FN-5, TN-386
Acc: 98.85%, Precision: 98.99%, Recall: 98.74%

A vs O (500 epochs)
TP-363, FP-0, FN-29, TN-379
Acc: 96.24%, Precision: 100%, Recall: 92.6%

A vs P (41 epochs)
TP-391, FP-3, FN-2, TN-400
Acc: 99.37%, Precision: 99.24%, Recall: 99.49%

A vs Q (500 epochs)
TP-394, FP-4, FN-4, TN-384
Acc: 98.98%, Precision: 98.99%, Recall: 98.99%

A vs R (269 epochs)
TP-375, FP-3, FN-6, TN-389
Acc: 98.84%, Precision: 99.21%, Recall: 98.43%

A vs S (157 epochs)
TP-382, FP-1, FN-5, TN-380
Acc: 99.22%, Precision: 99.74%, Recall: 98.71%

A vs T (121 epochs)
TP-392, FP-2, FN-2, TN-396
Acc: 99.49%, Precision: 99.49%, Recall: 99.49%

A vs U (500 epochs)
TP-383, FP-1, FN-17, TN-400
Acc: 97.75%, Precision: 99.74%, Recall: 95.75%

A vs V (500 epochs)
TP-393, FP-8, FN-0, TN-375
Acc: 98.97%, Precision: 98%, Recall: 100%

A vs W (13 epochs)
TP-400, FP-0, FN-2, TN-368
Acc: 99.74%, Precision: 100%, Recall: 99.5%

A vs X (17 epochs)
TP-391, FP-1, FN-1, TN-395
Acc: 99.75%, Precision: 99.74%, Recall: 99.74%

A vs Y (500 epochs)
TP-389, FP-36, FN-1, TN-361
Acc: 95.3%, Precision: 91.53%, Recall: 99.74%

A vs Z (231 epochs)
TP-394, FP-1, FN-1, TN-365
Acc: 99.74%, Precision: 99.75%, Recall: 99.75%

Learning rate of 0.8

This higher learning rate resulted in poorer accuracy. Weights took longer to converge - in fact, many more letters maxed out the instituted epoch cap. Also accuracy values fell to 97-98%.

A vs B (32 epochs)
TP-400, FP-3, FN-4, TN-370
Acc: 99.1%, Precision: 99.26%, Recall: 99.01%

A vs C (500 epochs)
TP-388, FP-0, FN-9, TN-365
Acc: 98.82%, Precision: 100%, Recall: 97.73%

A vs D (304 epochs)
TP-392, FP-6, FN-5, TN-394
Acc: 98.62%, Precision: 98.49%, Recall: 98.74%

A vs E (23 epochs)
TP-393, FP-1, FN-0, TN-384
Acc: 99.87%, Precision: 99.75%, Recall: 100%

A vs F (27 epochs)
TP-395, FP-5, FN-2, TN-380
Acc: 99.1%, Precision: 98.75%, Recall: 99.5%

A vs G (500 epochs)
TP-392, FP-2, FN-8, TN-379
Acc: 98.72%, Precision: 99.49%, Recall: 98%

A vs H (500 epochs)
TP-380, FP-5, FN-17, TN-359
Acc: 97.11%, Precision: 98.7%, Recall: 95.72%

A vs I (500 epochs)
TP-391, FP-3, FN-2, TN-376
Acc: 99.35%, Precision: 99.24%, Recall: 99.49%

A vs J (500 epochs)
TP-388, FP-25, FN-0, TN-355
Acc: 96.74%, Precision: 93.95%, Recall: 100%

A vs K (500 epochs)
TP-397, FP-10, FN-3, TN-354
Acc: 98.3%, Precision: 97.54%, Recall: 99.25%

A vs L (194 epochs)
TP-399, FP-1, FN-8, TN-367
Acc: 98.84%, Precision: 99.75%, Recall: 98.03%

A vs M (500 epochs)
TP-382, FP-4, FN-4, TN-400
Acc: 98.99%, Precision: 98.96%, Recall: 98.96%

A vs N (107 epochs)
TP-390, FP-3, FN-6, TN-387
Acc: 98.85%, Precision: 99.24%, Recall: 98.48%

A vs O (500 epochs)
TP-360, FP-0, FN-32, TN-379
Acc: 95.85%, Precision: 100%, Recall: 91.84%

A vs P (43 epochs)
TP-391, FP-3, FN-2, TN-400
Acc: 99.37%, Precision: 99.24%, Recall: 99.49%

A vs Q (500 epochs)
TP-395, FP-5, FN-3, TN-383
Acc: 98.98%, Precision: 98.75%, Recall: 99.25%

A vs R (263 epochs)
TP-376, FP-9, FN-5, TN-383
Acc: 98.19%, Precision: 97.66%, Recall: 98.69%

A vs S (154 epochs)
TP-381, FP-1, FN-6, TN-380
Acc: 99.09%, Precision: 99.74%, Recall: 98.45%

A vs T (139 epochs)
TP-392, FP-2, FN-2, TN-396
Acc: 99.49%, Precision: 99.49%, Recall: 99.49%

A vs U (500 epochs)
TP-383, FP-1, FN-17, TN-400
Acc: 97.75%, Precision: 99.74%, Recall: 95.75%

A vs V (500 epochs)
TP-393, FP-9, FN-0, TN-374
Acc: 98.84%, Precision: 97.76%, Recall: 100%

A vs W (22 epochs)
TP-400, FP-0, FN-2, TN-368
Acc: 99.74%, Precision: 100%, Recall: 99.5%

A vs X (25 epochs)
TP-391, FP-1, FN-1, TN-395
Acc: 99.75%, Precision: 99.74%, Recall: 99.74%

A vs Y (500 epochs)
TP-389, FP-28, FN-1, TN-369
Acc: 96.32%, Precision: 93.29%, Recall: 99.74%

A vs Z (205 epochs)
TP-394, FP-1, FN-1, TN-365
Acc: 99.74%, Precision: 99.75%, Recall: 99.75%

Learning rate of 0.05

The lower learning rate understandably initiated more epochs being executed. However, I felt this led to some perceptrons becoming over-trained, resulting in some accuracy values falling into the 95% range.

A vs B (71 epochs)
TP-399, FP-0, FN-5, TN-373
Acc: 99.36%, Precision: 100%, Recall: 98.76%

A vs C (500 epochs)
TP-393, FP-0, FN-4, TN-365
Acc: 99.48%, Precision: 100%, Recall: 98.99%

A vs D (214 epochs)
TP-394, FP-8, FN-3, TN-392
Acc: 98.62%, Precision: 98.01%, Recall: 99.24%

A vs E (20 epochs)
TP-392, FP-2, FN-1, TN-383
Acc: 99.61%, Precision: 99.49%, Recall: 99.75%

A vs F (24 epochs)
TP-396, FP-6, FN-1, TN-379
Acc: 99.1%, Precision: 98.51%, Recall: 99.75%

A vs G (157 epochs)
TP-392, FP-2, FN-8, TN-379
Acc: 98.72%, Precision: 99.49%, Recall: 98%

A vs H (500 epochs)
TP-392, FP-14, FN-5, TN-350
Acc: 97.5%, Precision: 96.55%, Recall: 98.74%

A vs I (500 epochs)
TP-391, FP-2, FN-2, TN-377
Acc: 99.48%, Precision: 99.49%, Recall: 99.49%

A vs J (500 epochs)
TP-384, FP-15, FN-4, TN-365
Acc: 97.53%, Precision: 96.24%, Recall: 98.97%

A vs K (500 epochs)
TP-398, FP-13, FN-2, TN-351
Acc: 98.04%, Precision: 96.84%, Recall: 99.5%

A vs L (195 epochs)
TP-399, FP-1, FN-8, TN-367
Acc: 98.84%, Precision: 99.75%, Recall: 98.03%

A vs M (500 epochs)
TP-386, FP-29, FN-0, TN-375
Acc: 96.33%, Precision: 93.01%, Recall: 100%

A vs N (55 epochs)
TP-394, FP-10, FN-2, TN-380
Acc: 98.47%, Precision: 97.52%, Recall: 99.49%

A vs O (217 epochs)
TP-361, FP-0, FN-31, TN-379
Acc: 95.98%, Precision: 100%, Recall: 92.09%

A vs P (39 epochs)
TP-392, FP-3, FN-1, TN-400
Acc: 99.5%, Precision: 99.24%, Recall: 99.75%

A vs Q (500 epochs)
TP-395, FP-5, FN-3, TN-383
Acc: 98.98%, Precision: 98.75%, Recall: 99.25%

A vs R (213 epochs)
TP-376, FP-3, FN-5, TN-389
Acc: 98.97%, Precision: 99.21%, Recall: 98.69%

A vs S (327 epochs)
TP-385, FP-4, FN-2, TN-377
Acc: 99.22%, Precision: 98.97%, Recall: 99.48%

A vs T (107 epochs)
TP-392, FP-3, FN-2, TN-395
Acc: 99.37%, Precision: 99.24%, Recall: 99.49%

A vs U (500 epochs)
TP-385, FP-2, FN-15, TN-399
Acc: 97.88%, Precision: 99.48%, Recall: 96.25%

A vs V (164 epochs)
TP-393, FP-15, FN-0, TN-368
Acc: 98.07%, Precision: 96.32%, Recall: 100%

A vs W (16 epochs)
TP-399, FP-3, FN-3, TN-365
Acc: 99.22%, Precision: 99.25%, Recall: 99.25%

A vs X (20 epochs)
TP-390, FP-1, FN-2, TN-395
Acc: 99.62%, Precision: 99.74%, Recall: 99.49%

A vs Y (500 epochs)
TP-388, FP-7, FN-2, TN-390
Acc: 98.86%, Precision: 98.23%, Recall: 99.49%

A vs Z (61 epochs)
TP-395, FP-2, FN-0, TN-364
Acc: 99.74%, Precision: 99.5%, Recall: 100%

CONCLUSION

Overall, this was a very intriguing project. Small changes in variables such as learning rate, weight convergence thresholds, and the number of examples used for training and testing can alter the results significantly. It was an interesting experiment to find the ideal values for each.

This also brings to light how difficult letter recognition can be. Throughout our lives, people have become quite adept at distinguishing between handwritten letters, mostly based on experience. A computer obviously lacks this, so it is impressive that it can accomplish the success rate it does with only a few mere seconds during which to become acquainted with the greatly varying styles of written characters.