CS440 Final project

Guantao Zhao(gz87),Chaoqin Chen(cc1504),Mingzhe Li(ML1252)

May 2018

1 Perceptron

1 Procedure Training:

We start with random weights and bias for the features. We loop through instances and update weights and bias. We continue until stopping condition is satisfied, the condition which is find the correct solution. For the update of weights and bias we use:

$\sum_{i=0}^{n} W_1 * X_1$
Then, we take the weighted sun of inputs and outputs:
1 if the sun is ¿ some adjustable threshold value (theta)
0 otherwise
$w = w + x_i$, if key is 1, but output is 0.
$w = w - x_i$, if key is 0, but output is 1.
In each iteration, we randomly shuffle the training samples, and use all the training data to update the weights if needed. And we run such training iteration 100 times for each training.

2 Accuracy:

Digits Test:
We trained the perceptron using 5000 digits training set and tested the prediction accuracy using the 1000 digits test set. Training iteration 100.
The average accuracy is 82.6%, with standard deviation of 2.34. The run-time accuracy and time are in Table 1.
Face Test:
We trained the perceptron using the 451 faces training set tested the prediction accuracy using the 150 faces test set. Training iteration 100. The average accuracy is 81.76%, with standard deviation of 7.47. The run-time accuracy and time are in Table 2.

Table 1: Size of digits training data and prediction accuracy

Training data size	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Accuracy	80.4%	80%	80.2%	80.5%	82.4%	84.3%	86.2%	82.2%	84.2%	85.6%
Training time	1.47s	2.94s	4.48s	5.83s	7.40s	8.89s	10.33s	11.90s	13.24s	15.75s

Table 2: Size of faces training data and prediction accuracy

Training data size	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Accuracy	64.5%	77.1%	78.4%	79.7%	81.7%	87.0%	89.7%	84.4%	88.7%	86.4%
Training time	0.56s	1.13s	1.67s	2.25s	2.78s	3.31s	3.88s	3.74s	4.96s	5.44s

1 Naive Bayes Classifier

Training

The task of training process of Naive Bayes classifier comprises of computing $P(F_i|Y)$ and $P(Y)$, Y is a random variable representing class label, $\{F1, F2, \cdots, F_N\}$ is a set of random variables representing features. $P(F_i|Y)$ is estimated by:

$$P(F_i = f_i|Y = y_i) = \frac{\sum c(f_i, y_i) + k}{\sum_{f_i} c(f_i, y_i) + k}$$

$c(f_i, y_i)$ is the number of occurrences of f_i in class y_i, k is a constant used to make sure $P(F_i = f_i|Y = y_i)$ will not receive a estimate of 0. $P(Y = y_i)$ is estimated by:

$$P(Y = y_i) = \frac{c(y_i)}{n}$$

$c(y_i)$ is number of training samples classified as y_i, n is the total number of training samples.

Accuracy

The size of the training set of the digit recognition is 5000, and the size of the testing set is 1000. Table shows that as the size of the training set increases, the training time and accuracy rate for digit classification also increases.

NB-digit:	Training time/ms	Test Accuracy/%	Validation Acc/%
10%, 500	193.930	0.684000	**0.731000**
20%, 1000	389.637	0.719000	**0.778000**
30%, 1500	589.118	0.731000	**0.794000**
40%, 2000	786.405	0.738000	**0.793000**
50%, 2500	989.123	0.756000	**0.803000**
60%, 3000	1176.958	0.758000	**0.801000**
70%, 3500	1406.531	0.762000	**0.808000**
80%, 4000	1551.578	0.766000	**0.809000**
90%, 4500	1774.223	0.763000	**0.820000**
100%, 5000	2182.545	0.771000	**0.818000**

The size of the training set of the face recognition is 451, and the size of testing set is 150. Table shows that as the size of the training set increase, the training time and accuracy rate for face recognition also increases.

NB-Face:	Training time/ms	Test Accuracy/%	Validation Acc/%
10%,45	89.819	0.760000	0.760797
20%,90	174.163	0.800000	0.784053
30%,135	261.998	0.846667	0.820598
40%,180	354.554	0.893333	0.870432
50%,225	441.524	0.873333	0.870432
60%,270	535.838	0.880000	0.863787
70%,315	600.080	0.893333	0.863787
80%,300	578.746	0.880000	0.867110
90%,405	791.500	0.893333	0.870432
100%,451	897.068	0.906667	0.870432

2 Neural Network

	Training time/ms	Test Accuracy/%	Validation Accuracy/%
10%---500	489.753	77	79.9
20%---1000	528.566	81.3	83.3
30%---1500	990.222	84.9	86.1
40%---2000	2217.444	84.1	86.7
50%---2500	1949.970	87.4	88.8
60%---3000	3254.699	87.7	88.5
70%---3500	3177.911	87.9	89.7
80%---4000	4267.397	87.5	90.3
90%---4500	3425.353	88	89.6
100%---5000	5206.297	89.2	90.9

----------neural_basic_digital----------

	Training time/ms	Test Accuracy/%	Validation Accuracy/%
10%---45	2276.521	60.6	62.5
20%---90	4241.541	84.7	82.1
30%---135	2858.942	86.7	87
40%---180	1594.504	88.7	83.3
50%---225	6052.751	89.3	90
60%---270	6508.720	90.7	87.7
70%---315	4915.414	88	88.7
80%---360	4506.720	90.7	89.4
90%---405	2455.431	88	87.7
100%---451	5940.150	88.7	89

----------neural_basic_face----------

For the neural web's input training data, it seen the training data as vector value, for example, if the training data is a 10x10, and the image is either marked or empty, we can store the data as a 10x10 vector as 0 or 1, where 1 means a marked pixel and 0 means empty pixel, in our code, we store those vector array in to .npy file using Numpy.

For the neural web's classifier, basically we can see that as to divide to group of different feature vector, just like use a line to divided 2 groups in the flat, left side output 0 and right side output 1. (as an example) , for every single training data, we might need to re-decide where to put that "line" to let the "line" divide them correctly, that's the basic idea of neural web function.

For our output data, we can see most likely the amount of training data and outputs Accuracy are directly proportional, and it becomes stable and not grow up from around 50% training data.

3 Conclusion

By comparison, Naive Bayes usually takes shortest time on traning, and perceptron is taking longest training time in the three function where the Neural Network function stays in the middle.But, Sometime our neural network might have precision loss, which cause unstable training time and result.

For the comparison of Accuracy, in digital, Naive Bayes has lowest average accuracy(0.7955), then the perceptron is higher than NB(0.826), and Neural Network have best accuracy(0.8738). For the face test, our Perceptron has lowest accuracy of 0.8176, Naive Bayes is 0.844 stay in the middle, and our Neural Network have highest accuracy of 0.8480.

Perceptron: standard deviation=0.0234(digital)/0.0747(face)
Average=0.826(digital)/0.8176(face)
NB:standard deviation=0.0006(digital)/0.00152(Face)
Average=0.7955(digital)/0.84419(face)
Neural Network:standard deviation=0.109196(digital)/0.60666(face)
Average=0.8738(digital)/0.8480(face)