CSE 574 Assignment 3

Group 104

Raman Keswani

Abhinav Kumar

Implementation of Logistic Regression

Accuracy with respect to each category:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Accuracy % | Misclassified Images | Total Images |
| Training Data | 92.77 | 3,615 | 50,000 |
| Validation Data | 91.51 | 849 | 10,000 |
| Test Data | 91.99 | 801 | 10,000 |

We can see that the test accuracy error is only slightly lesser than training accuracy. If we see lesser testing accuracy then training error, then it means that our model didn’t see enough amount of samples. Also in our case we are not applying any regularization, so it can also be case of over fitting to training data.

Misclassified images for each category (on training data):

|  |  |
| --- | --- |
| Label/Digit Value | Number of times wrongly classified |
| 0 | 111 |
| 1 | 114 |
| 2 | 434 |
| 3 | 525 |
| 4 | 299 |
| 5 | 531 |
| 6 | 182 |
| 7 | 296 |
| 8 | 587 |
| 9 | 538 |

Misclassified images for each category (on test data):

|  |  |
| --- | --- |
| Label/Digit Value | Number of times wrongly classified |
| 0 | 19 |
| 1 | 20 |
| 2 | 113 |
| 3 | 71 |
| 4 | 66 |
| 5 | 131 |
| 6 | 49 |
| 7 | 83 |
| 8 | 122 |
| 9 | 109 |

Observations:

1. Our model works fairly well for all the three data sets (training, validation and test)
2. Digit 0 was the misclassified least number of times whereas digit 5 was misclassified most number of times.
3. It took 28 minutes to train the model with max number of iterations = 100

Extra Credit: Multi-class Logistic Regression

Accuracy with respect to each category:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Accuracy % | Misclassified Images | Total Images |
| Training Data | 93.45 | 3,275 | 50,000 |
| Validation Data | 92.48 | 752 | 10,000 |
| Test Data | 92.55 | 745 | 10,000 |

Misclassified images for each category (on training data)

|  |  |
| --- | --- |
| Label/Digit Value | Number of times wrongly classified |
| 0 | 137 |
| 1 | 149 |
| 2 | 432 |
| 3 | 463 |
| 4 | 258 |
| 5 | 419 |
| 6 | 137 |
| 7 | 268 |
| 8 | 442 |
| 9 | 416 |

Misclassified images for each category (on test data)

|  |  |
| --- | --- |
| Label/Digit Value | Number of times wrongly classified |
| 0 | 20 |
| 1 | 25 |
| 2 | 102 |
| 3 | 92 |
| 4 | 60 |
| 5 | 109 |
| 6 | 35 |
| 7 | 75 |
| 8 | 97 |
| 9 | 79 |

Observations:

1. Our model works fairly well for all the three data sets (training, validation and test)
2. Digit 0 was the misclassified least number of times whereas digit 5 was misclassified most number of times.
3. It took only 1 minute to train the model with max number of iterations = 100

We can see that multi class strategy is much faster than one-vs-all strategy. Multi class strategy takes around 1 minute to train the model whereas one-vs-all took takes around 28 minutes to train the model.

Accuracy wise multi class strategy (92.55%) is slightly better than one-vs-all strategy (91.99%)

Support Vector Machines

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kernel | Training Accuracy | Validation Accuracy | Test Accuracy | Training Time (sec) |
| Linear Kernel | 97.29 | 93.64 | 93.78 | 754 |
| Radial Basis Function, Gamma =1 | 100.00 | 15.48 | 17.14 | 12527 |
| Radial Basis Function, Gamma Default | 94.30 | 94.02 | 94.42 | 1450 |
| RBF Gamma = Default C = 1 | 94.30 | 94.02 | 94.40 | 1456 |
| RBF Gamma = Default C = 10 | 97.13 | 96.17 | 96.10 | 868 |
| RBF Gamma = Default C = 20 | 97.95 | 96.90 | 96.67 | 726 |
| RBF Gamma = Default C = 30 | 98.37 | 97.10 | 97.04 | 812 |
| RBF Gamma = Default C = 40 | 98.70 | 97.23 | 97.19 | 845 |
| RBF Gamma = Default C = 50 | 99.00 | 97.31 | 97.19 | 782 |
| RBF Gamma = Default C = 60 | 99.19 | 97.38 | 97.16 | 749 |
| RBF Gamma = Default C = 70 | 99.34 | 97.36 | 97.26 | 788 |
| RBF Gamma = Default C = 80 | 99.44 | 97.39 | 97.33 | 725 |
| RBF Gamma = Default C = 90 | 99.54 | 97.36 | 97.34 | 685 |
| RBF Gamma = Default C = 100 | 99.61 | 97.41 | 97.40 | 711 |

We can see that SVM with RBF Kernel C=100 and Default Gamma gave the best testing accuracy whereas SVM with RBF Kernel Gamma = 1 gave the worst testing accuracy.

Although gamma = 1 gives the perfect training accuracy (100%), it gives extremely poor testing and validation accuracy. This is a classic case of overfitting because of such large value of gamma. For large value of gamma, the radius of the area of influence of the support vectors only includes the support vector itself. In other words, it creates a Gaussian function which has small variance and two points are considered similar just if they are close to each other.

Ideal value of gamma should be around 0.1

In terms of performance, RBF kernel with gamma = 1 takes the most time to train the model (208 minutes), whereas RBF kernel with C=90 takes least amount of time to train the model (1 minute)

Also there is little variation in training time for all the kernels except for RBF with gamma = 1