COMP 540 Homework 4

Peiguang Wang, Xinran Zhou

Due: 3/05/2018

Part 4: Support vector machines for multi-class classification

1. It is possible that once in a while a dimension in the gradient check will not match exactly. What could such a discrepancy be caused by? Is it a reason for concern?

   Solution. The discrepancy is caused by the non-differentiable part of the loss function. The loss function is non-differentiable in when $\theta^{(j)^T}x(i) - \theta^{y_i^T}x(i) + \Delta = 0$.
   It is not a reason for concern. The loss will not increase if gradient is computed this way. So it is not a reason for concern.

2. Loss history figure.

   Solution. The loss changes with iteration times. The loss history plot is shown in Figure 1.

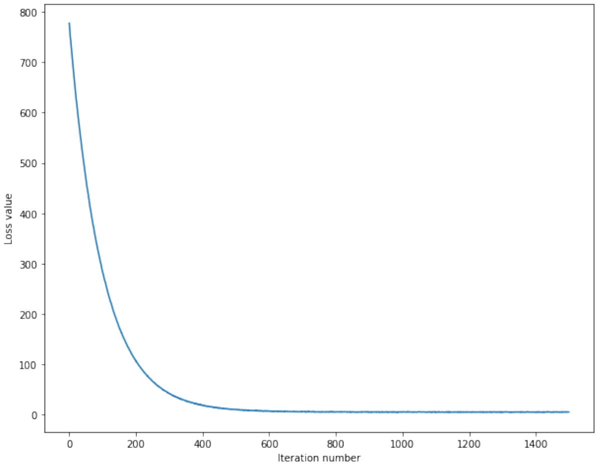

Figure 1: Loss History

3. Search for the best svm model.

   Solution. Experiment with 4 learning rates and 4 regularization strengths. The result are:

```
lr 1.000000e-08 reg 1.000000e+04 train accuracy: 0.230592 val accuracy: 0.249000
lr 1.000000e-08 reg 5.000000e+04 train accuracy: 0.264000 val accuracy: 0.258000
lr 1.000000e-08 reg 1.000000e+05 train accuracy: 0.301898 val accuracy: 0.321000
lr 1.000000e-08 reg 5.000000e+05 train accuracy: 0.323796 val accuracy: 0.340000
lr 5.000000e-08 reg 1.000000e+04 train accuracy: 0.315531 val accuracy: 0.323000
lr 5.000000e-08 reg 5.000000e+04 train accuracy: 0.373837 val accuracy: 0.392000
lr 5.000000e-08 reg 1.000000e+05 train accuracy: 0.360265 val accuracy: 0.374000
lr 5.000000e-08 reg 5.000000e+05 train accuracy: 0.313184 val accuracy: 0.331000
lr 1.000000e-07 reg 1.000000e+04 train accuracy: 0.372735 val accuracy: 0.376000
lr 1.000000e-07 reg 5.000000e+04 train accuracy: 0.368714 val accuracy: 0.376000
lr 1.000000e-07 reg 1.000000e+05 train accuracy: 0.354429 val accuracy: 0.357000
lr 1.000000e-07 reg 5.000000e+05 train accuracy: 0.317000 val accuracy: 0.332000
lr 5.000000e-07 reg 1.000000e+04 train accuracy: 0.369204 val accuracy: 0.369000
lr 5.000000e-07 reg 5.000000e+04 train accuracy: 0.339816 val accuracy: 0.336000
lr 5.000000e-07 reg 1.000000e+05 train accuracy: 0.291224 val accuracy: 0.292000
lr 5.000000e-07 reg 5.000000e+05 train accuracy: 0.280633 val accuracy: 0.289000
lr 1.000000e-06 reg 1.000000e+04 train accuracy: 0.362796 val accuracy: 0.375000
lr 1.000000e-06 reg 5.000000e+04 train accuracy: 0.254204 val accuracy: 0.266000
lr 1.000000e-06 reg 1.000000e+05 train accuracy: 0.265082 val accuracy: 0.277000
lr 1.000000e-06 reg 5.000000e+05 train accuracy: 0.227735 val accuracy: 0.232000
best validation accuracy achieved during cross-validation: 0.392000
```

Visualized the results and the results are shown in Figure 2.

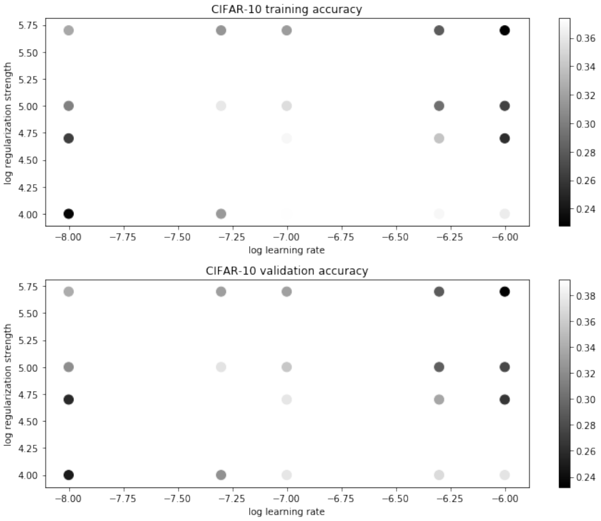

Figure 2: Visualized Results for Parameter Selection

4. Evaluate the test set accuracy on the best svm learned.

   Solution. After searching for the best values of learning rate and regularization. We obtain the best svm model. The results on the test set is: *linear SVM on raw pixels final test set accuracy: 0.368300.* The accuracy is similar to the accuracy of validation set. Visualization of the results is shown in Figure 3.

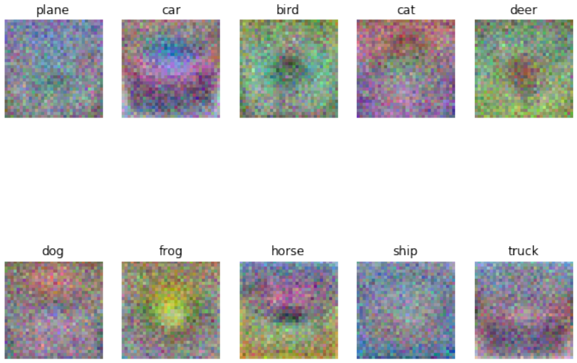

Figure 3: Visualized Results for Best SVM Multiclass Model

5. Comparing the performance of multi-class SVM and softmax regression. Which approach takes longer to train, which approach achieves higher performance? Compare the visualizations of the θ parameters learned by both methods - do you see any differences? Comment on hyper parameter selection for both methods.

Solution. First compare the accuracy.

- Softmax on raw pixels final test set accuracy: 0.405100
- Linear SVM on raw pixels final test set accuracy: 0.368300.

The accuracy on Softmax and SVM are almost the same level. Softmax is slightly higher than linear SVM method.

Then compare the visualization of these two methods.
Figure 4 shows the results on softmax model.

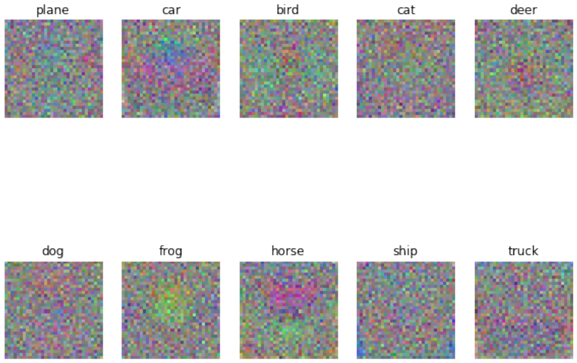

Figure 4: Visualized Results for Best Softmax Model

Figure 5 shows the results on SVM multiclass model.

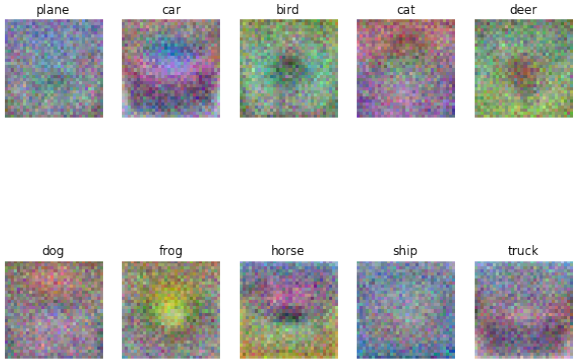

Figure 5: Visualized Results for Best SVM Multiclass Model

From the figure it seems the SVM model extracts a more ”meaningful” features. Here ”meaningful” means understandable for humans. The contour and edges are more clear in SVM model results.

Compare the time of these two methods.

- Softmax vectorized loss: 2.352202e+00 computed in 0.483000s
- Linear SVM Vectorized loss: 9.293820e+00 computed in 0.016000s

In terms of time cost. Softmax takes a much longer time to train. The reason is that SVM learns a sparse kernel.