CS 532 (Sec. 4) – Project Update 2

Alex Guo

Due Date: December 1, 2020

1 Progress

Link to repo: https://github.com/ag262/CS532.git

Reminder: My dataset contains 15,000 images of chinese digits/characters (the number of classes for the data is 15).

I modified my first algorithm in "kNN.py", where I switched to just using the scikit-learn library (since it's much faster than the code I wrote before). I did 10-fold validation to find an ideal k parameter, which from the figure below looks to be $k = 1$. The percent error for $k = 1$ on the test data is 13.5%, which is pretty good.

```
k                  = [ 1    5    7  ]
avg CV percent err = [14.3 16.4 19.4]
standard deviation = [ 0.8  0.9  0.9]
```

I think $k = 1$ is the best here because images are "far" away from each other, and thus there's no noise/variance to be reduced by increasing k and we might as well lower the bias as much as possible. Also, I found that using the L1-norm instead of the L2-norm doesn't affect anything.

I wrote up my second algorithm in "SVM.py", where I also used the scikit-learn library to implement the algorithm. The two parameters are the kernel and the regularization parameter λ in the objective function. I looked at the ['linear','poly','rbf','sigmoid'] kernels and the numbers [0.01,0.1,1.0,5.0] for λ. I didn't have time to do cross validation, so I just used 1-fold validation (i.e. just the test data) to get the table of percent errors below (note: i, j element corresponds to the ith kernel and jth λ value).

```
[[ 8.63 11.23 11.73 11.73]
 [12.03  7.53  3.7   3.17]
 [15.3  11.3   3.93  2.23]
 [22.77 18.93 22.33 26.03]]
```

The kernels that lead to smaller errors make sense, but I'm not sure why increasing the margin by increasing λ decreases the error (because I conjectured previously that the image data are far away from each other and thus a large margin doesn't really matter).

I've just started writing my third algorithm in "NN.py".

2 Plan ahead

I need to finish up the investigation of SVM by doing k-fold validation. I will then need to finish up coding my NN algorithm. I'm planning to do a simple feedforward NN first and then try to add some convolutional layers after everything's working.

3 Project timeline (revised)

12/2 - 12/11: execute my plans I wrote about in "Plan ahead" and write up the final report

12/12 (milestone): final report due