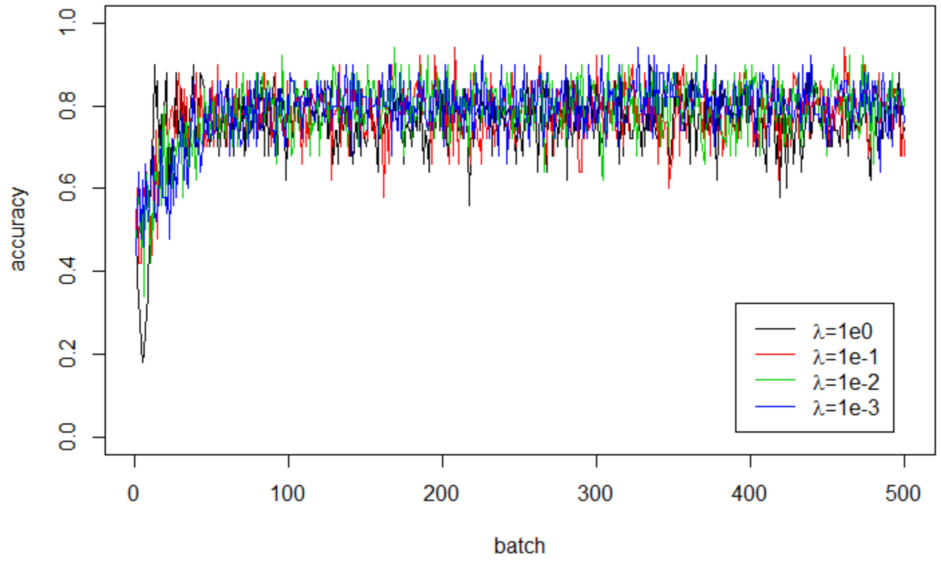
CS498DF HW2 Report

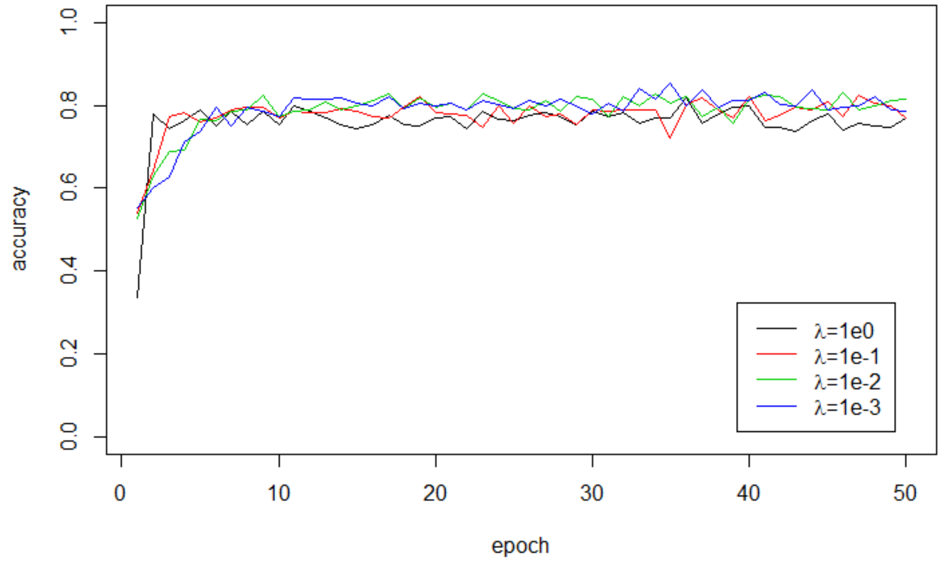
Jiayu Chen (jchen267)

Xuewei Zhang (xzhan160)

We wrote the SVM classifier from scratch, using methods and parameters in our textbook. First we select continues data, scale label into -1/1 value, and scale features into zero-mean one-standard deviation data. Using 80% data for training, 10% for validation of regularization constant, and 10% for testing. We use 50 epochs with 300 steps each, randomly set initial state of the SVM, and tried regularization constant as 1, 0.1, 0.01 and 0.001.

Using the Adult Data Set, we have estimated accuracy plotted against epochs or batches as the following graphs: (one batch is 30 steps)





Easy to see, as regularization constant goes down, they tend to be slower to converge at beginning, but they do end up with higher accuracy at the end. **We can see the best result of our SVM while regularization constant is 0.01**, while it has a good accuracy and begin in an increasing trend.

We chose the SVM with regularization constant as 0.01 and test it on the testing dataset, and got an accuracy as **81.3%**.

Here comes an interesting question, if we build a simple classifier, that for any input features, the simple classifier classify it as negative (<50K), we'll get an accuracy as 75.6%. So with all these trouble in SVM, seems like we didn't do that much.

But if we evaluate accuracy in a table, there is some good improvement. For our SVM, we have:

	SVM Output: True	SVM Output: False
Real Value: True	7.5%	16.8%
Real Value: False	1.8%	73.8%

While for the simple classifier, we have:

	SVM Output: True	SVM Output: False
Real Value: True	0%	24.4%
Real Value: False	0%	75.6%

Say if we really care about reducing false negative rate, or increasing true positive rate, SVM has some significant improvement over the simple classifier.