REPORT

The UC Irvine machine learning data repository hosts a collection of data on adult income, donated by Ronny Kohavi and Barry Becker. You can find this data at https://archive.ics.uci.edu/ml/datasets/Adult For each record, there is a set of continuous attributes, and a class "less than 50K" or "greater than 50K". There are 48842 examples. You should use only the continuous attributes (see the description on the web page) and drop examples where there are missing values of the continuous attributes. Separate the resulting dataset randomly into 10% validation, 10% test, and 80% training examples.

Write a program to train a support vector machine on this data using stochastic gradient descent. You should not use a package to train the classifier (that's the point), but your own code. You should ignore the id number, and use the continuous variables as a feature vector. You should scale these variables so that each has unit variance. You should search for an appropriate value of the regularization constant, trying at least the values [1e-3, 1e-2, 1e-1, 1]. Use the validation set for this search. You should use at least 50 epochs of at least 300 steps each. In each epoch, you should separate out 50 training examples at *random for evaluation (call this the set held out for the epoch). You should compute the accuracy of the current classifier on the set held out for the epoch every 30 steps.*

As far as I can tell, the code implements the assignment to the letter and runs without errors on my machine. I have the following answers for the specific sub-tasks of the assignment:

- *A plot of the accuracy every 30 steps, for each value of the regularization constant.*

Please see file “accuracies.jpeg” re-generated each time you run the code. The values for each regularization constant lambda are shown with a different color

- *A plot of the magnitude of the coefficient vector every 30 steps, for each value of the regularization constant.*

Please see file “magnitudes.jpeg” re-generated each time you run the code. The values for each lambda are shown with a different color

- *Your estimate of the best value of the regularization constant, together with a brief description of why you believe that is a good value.*

According to the last run of the code, the resulting best value of the regularization constant lambda was “0.01”. The best value of lambda is based on the best accuracy after running each classifier on the validation dataset. I noticed that sometimes it is “0.001”, and sometimes it is “0.01”. When I also tried “0.0001” as the fifth value of lambda (not implemented in the final submitted code), there were also instances when the best value was “0.0001”. However, the accuracy of the best classifier on the test dataset was always around 0.8. I believe that different values of the best estimate of lambda depend on the way how the original dataset is split (which is a random event) and also on what exact 50 random examples are held out for each epoch.

- *Your estimate of the accuracy of the best classifier on the 10% test dataset data*

According to the last run of the code, the accuracy of the best classifier on the test dataset was “0.80696.”

My comments:

The results obtained after running the code and shown on the two submitted graphs do confirm the main points about an SVM with SGD:

- the accuracy makes large changes early, then settles down to make slight changes;
- quite large changes in regularization constant have small effects on the outcome, but there is a best choice;
- for larger values of the regularization constant, aT*a is smaller;
- the method doesn’t need to see all the training data to produce a classifier.

•