hw03 regression report

1 requirements

1. Make sure that you have implemented a variable batch size using the constructor given for LogisticRegression. Try different batch sizes (e.g. 1, 5, 10, 75, etc.) and report the accuracy and number of epochs taken to converge.
a. What tradeoffs exist between good accuracy and quick convergence?

when batch size is 75:
finish traning
Test Accuracy: 77.2 percentage
Number of Epochs: 97 runtime:up to 20 seconds or less

when batch size is 10:
finish traning
Test Accuracy: 84.3 percentage
Number of Epochs: 45 runtime:up to 20 seconds or less

when batch size is 5:
finish traning
Test Accuracy: 83.5 percentage
Number of Epochs: 22 runtime:up to 10 seconds or less

when batch size is 1:
finish traning
Test Accuracy: 95.3 percentage
Number of Epochs: 20 runtime:up to 10 seconds or less

b. Why do you think the batch size led to the results you received?

I think smaller batch size can compute faster or less iterations of convergence in terms of calculation to get the correct convergence. If we pick large batch size, the accuracy could decrease significantly. However, if your batch is too small, then your gradient could be noisy, which means that might affect your running time performance since you will have more epochs...On the contrast, the nice thing about using large batch size, even though the accuracy went down, still less epochs which could be more efficient..

2. Take a look at the Colab notebook here we provided to clean and process the data. Which categories did we one-hot encode and why? Using Google Colab: To use our data, right click on hw03 in the Shared with me ¿ hw03 section, and click Add to My Drive. Within the hw03 folder, double click the file (cleaning data.ipynb). If this is your first time using Google Colab, you may need to click the dropdown on the top center, click Connect more apps, and connect Google Colab first. Afterwards, click the dropdown again to open the notebook in Google Colab. To run the notebook, you will either need to open in playground mode, or make a copy.

Many Thanks for TA Angie Kim, I now understood it. If you open the notebook, you should find 'workclass', 'marital-status', 'occupation', 'relationship', 'race', 'native-country' these are the things we are doing one-hot encode. Because the fact that we cannot assign a comparative values for these labels. For example, how do we compare the workclass: doctor and teacher? If we assign doctor a value higher than teacher, this will outrage a lot of people who works as a teacher. For these we cannot assign a value, the better way of doing it is to encode a one-hot encoding. For things that are comparative, we can just assign different values to represent, like educational level: a PHD degree is significantly valuable than a high school diploma except Bill Gates.....

3. Try to run the model with unnormalized data.csv instead of normalized data.csv. Report your findings when running the model on the unnormalized data. In a few short sentences, explain what normalizing the data does and

why it affected your model's performance.

batch size:5
thresold:0.00001
Test Accuracy: 34.9 percentage
Number of Epochs: 1
As you can tell we have a pretty low accuracy while the normalized data set got around 95 accuracy. This make sense, since all features have different range, like the range from age is 0-100 (most people's range is below 85) and salary range depends on different country is 50k-100k (could be more...)If we don't normalized our features' data on the same scale, we are running into troubles. Our predictions won't be accurate.

4. Try the model with normalized data nosens.csv; in this data file, we have removed the race and sex attributes. Report your findings on the accuracy of your model on this dataset (averaging over many random seeds here may be useful). Can we make any conclusion based on these accuracy results about whether there is a correlation between sex/race and education level? Why or why not?

In general, if you remove those attributes like race and sex which won't contribute the prediction as other attributes do or somethings cannot be distinguish by different values, you should get better accuracy.In terms of the running time shouldn't be changed too much.

My local performance:
batch:5
thresold:0.00001
Test Accuracy: 93.8 percentage
Number of Epochs: 125

5. (Optional) Any other thoughts on the assignment, dataset, or report?

I feel this project is not hard to code but the description is hard to understand and here are the things which I think is difficult to keep track of...

For example:
1. softmax confusion
2. the shuffling process
3. gradient part in the description
4. the outer product confusion