CSE 446 HW4

Rohan Mukherjee

May 30, 2024

A1.

a. If the vectors are already lying in a k dimensional subspace, then PCA will give us the matrix that projects onto that k dimensional subspace. Since projecting a vector onto a subspace that it already lives in will just return the same vector, we see that the projection of the data onto the subspace will be just the data itself. Thus we have a 0 reconstruction error.

b. False. The columns of V are the eigenvectors of X^TX, not the rows.

c. False. Choosing k to minimize the k means objective does give good clusters for the data, but in the examples below, I got a lot of 9s in the 10 clusters and no 4s.

d. False. The SVD of a matrix is in general not unique. We could permute the columns of U, rows of V, and entries of S (by the same permutation) to get another singular value decomposition.

e. False. Consider the matrix $aI \in \mathbb{R}^{n\times n}$ for some nonzero constant $a \in \mathbb{R}$. This matrix obviously has rank n, being invertible (with inverse $a^{-1}I$), but this matrix has only one eigenvalue a.

A2.

a. The first thing I would do, like always, is to normalize the data by making the features have mean 0 and variance 1. Since we want to create an algorithm that learns the factors that contribute most to acquiring a specific disease, we want a relationship that puts weights on each of the factors and outputs the probability of having the disease. We can thus use binary logistic regression for this problem, because we want to predict if a person has a disease or not. The inputs in the weight vector will tell us which factors cause the disease.

   We are left with a few questions. The first problem is that the logistic regression model might get bad accuracy on the validation set. This is extremely hard to get around–the main way to get around this is to allow non-lineararity in the model using multi-layered neural networks, but then we can't explain which factors cause the disease. Similarly, if we have a very small amount of training data, but an enormous amount of factors, we will almost surely overfit to the training data, leading to poor generalization. The last thing is that we could have completely nonsense features that certainly don't cause the disease, but show up as hugely positive weights in the model. For example, one of the features could be the number of times the person has been to the moon, which is obviously not a factor in acquiring the disease. Finally, given the limited amount of personal data from the person, you can put their data into the model and get a prediction of whether they have the disease or not.

b. The main potential shortcoming for my model having different accuracy on different populations is that if a lot of the training data is only on a specific population, say in a specific country, then the model won't really use that all the people are living in that one country, because it can't gain much information from a feature with really small variance. Then when another population comes around, whose disease susceptibility is really correlated with what country they live in, the model will have very bad accuracy on that data. The way to fix this is to get a more diverse training set, with people from all over the world. We could also use cross validation to train the model on different populations at a time and see if it generalizes well to the other populations.

c. Some real world implications that would come from ignoring the issues described in the question is again the problem of correlation and causation. Police might spend a lot more time patrolling a more dangerous neighborhood, so the crime statistic will be higher along with the police patrol being higher. From the above, one would say that it must be that police patrols cause crime, which is not true and again the problem of correlation vs causation. If many of the shortcomings such as crimes being reported at higher rates in minority neighborhoods, the model might put high weight on the minority percentage in a community and hence one might conclude that being a minority is the reason for crime. This is a problem of the model and a shortcoming of trying to interpret numbers like this in a real world setting.

A3.

a. Here is the graph of the training epoch vs accuracy, for the best 3 performing models:

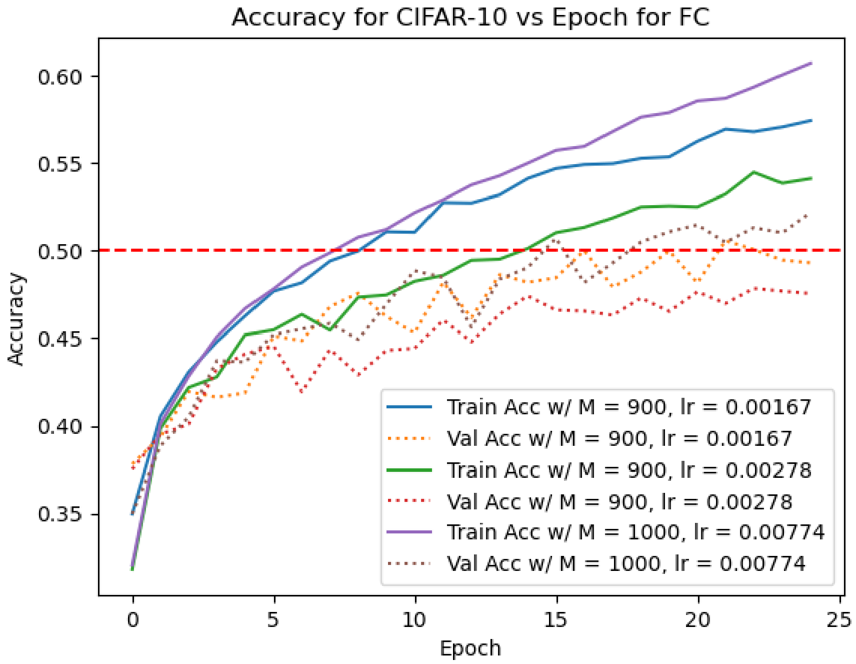

Figure 1: Training epoch vs accuracy

For this first model, I used random search with 30 iterations over learning rates in a logspace from 10^{-3} to 10^{-1}, the M's were in a linspcae from 10 to 1000 and the momentums were in a linspace from 0.9 to 0.99, each with 10 elements in the range. I thought it would be best to have 10 iterations per hyperparameter. The best performing model as seen above has $M = 1000$ with learning rate of 0.00774. It had a test accuracy of 0.5178.

b. Here is the graph of the training epoch vs accuracy for the best 3 performing models for the convolutional neural network:

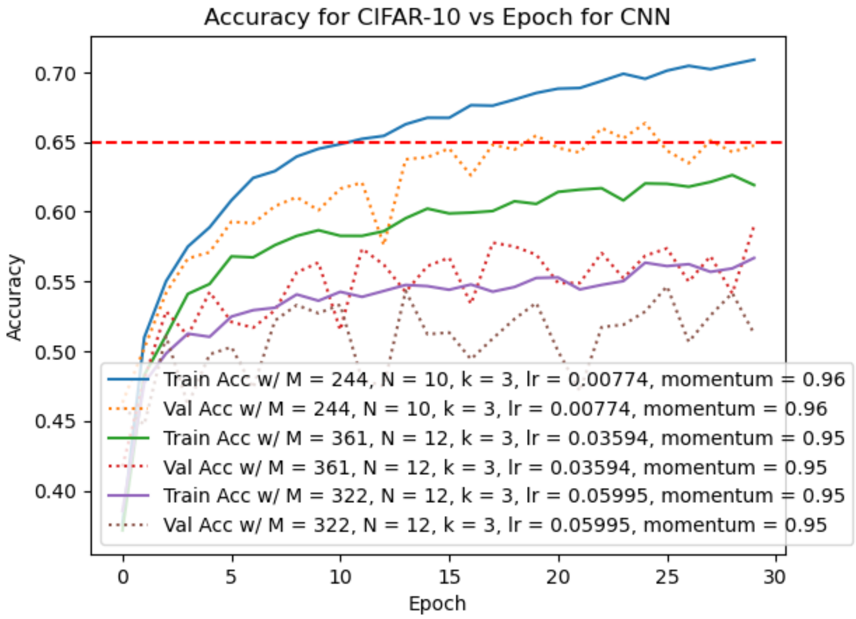

Figure 2: Training epoch vs accuracy

I used random search over these parameters:

```
lrs = np.logspace(-3, -1, 10)
Ms = np.linspace(50, 400, 10).astype(int)
Ns = np.linspace(10, 20, 5).astype(int)
ks = np.linspace(2, 8, 5).astype(int)
Momentums = np.linspace(0.9, 0.99, 10)
```

The best performing model got an accuracy of 0.66317 on the test data, with hyperparameters $M = 244$, $N = 10$, $k = 3$, $lr = 0.00774$ and $momentum = 0.96$.

A4.

The image representing the 10 clusters is shown below:

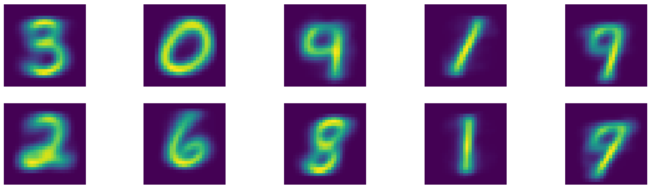

Figure 3: Clusters of the data

A5.

I spent around 15 hours on this assignment. Training the models was the most time consuming part of the assignment as the Ed Discussion hinted at. Coding in pytorch is really fun, because it makes everything so simple.