CSCI.635.01 - Intro to Machine Learning (CSCI63501...

Homework 4

1. [25 points] The `np_nn.py` training phase often fails due to numerical errors. Propose and implement fixes for the problem. The fixes can occur during initialization, during backprop, or during gradient descent; any time during the training phase is OK, as long as the system performs substantially more reliably than it did before (i.e., few to zero numerical errors.)

At the top of the file, insert a long quote that begins like this:

```
"""
Homework 4.1: Fixes I made to improve training:

"""
```

And explain your changes. Submit two files: `np_nn_before.py`, for before you made any of these changes (i.e., it should be identical to what you submitted for HW3), and `np_nn_after.py`, for after you made the changes. This way the grader can verify changes you made using the diff command.

2. [25 points] Download the files `markov_model.py` and `graph_data_continuous.csv`.
The code as written currently runs a graphical lasso. Modify the code so that it fits the data to the two graphical models given in the long quotes at the top of the file, using the Langrange multiplier model and simple gradient descent for the Θ (represented as `t` in `markov_model.py`) and Γ variables (not currently represented in the code). Note that there are ways to do this without actually discovering the Γ variables. It is OK if you take this path; the important thing is to estimate Θ.

Submit two versions of this code, each adapted to one of the graphs given in the long quotes. Call them `markov_model_2a.py` and `markov_model_2b.py`, respectively.

3. [25 points] Now download the source file `gibbs.py`. The source file contains only a 64x64 array, representing the parameters Θ of an Ising model on 64 binary variables. Write a Gibbs sampler and use it to estimate $\Phi(\Theta)$.

Call your program `gibbs.py` and the last line should be:

```
# We estimate \Phi to be:
```

and write your estimate for \Phi there (ideally the grader should be able to run the code and it should output close to the same value each time).

4. [25 points] Using `graph_data_continuous.csv:`
a. Create a scatterplot of the first two columns of this dataset. Call this plot `first_two.pdf`
b. Using only commands from numpy and scipy, write a program that that performs PCA on the data. Then project the data into the top two principal components and create a scatterplot of the results. Call the program `pca.py`. Call the scatterplot `top_two.pdf`.
c. Project the data into the bottom two principal components and create a scatterplot of the results. Call this `bottom_two.pdf`.

100 % 3 of 3 topics complete

graph_data_continuous
CSV File
Updated ✓

markov_model
PY File
✓

gibbs
PY File
✓