CSCI 1051 Homework 3

January 26, 2023

Submission Instructions

Please upload your solutions by **5pm Friday January 27, 2023.** Remember you have 24 hours no-questions-asked *combined* lateness across all assignments.

- You are encouraged to discuss ideas and work with your classmates. However, you **must acknowledge** your collaborators at the top of each solution on which you collaborated with others and you **must write** your solutions independently.
- Your solutions to theory questions must be typeset in LaTeX or markdown. I strongly recommend uploading the source LaTeX (found on the homepage of the course website) to Overleaf for editing.
- Your solutions to coding questions must be written in a Jupyter notebook. I strongly suggest working with colab as we do in the demos.
- You should submit your solutions as a **single PDF** via the assignment on Canvas.

Problem 1 (from January 23)

We calculated the policy gradient for the general reinforcement learning problem in class. For this problem, your task is to calculate the policy gradient in the special case where the trajectory is a single action. That is, $\tau = (a_i)$ where $i \in \{1, 2, \ldots, k\} = [k]$ and k is the number of actions. Further, assume that there is only one state and the reward for action a_i is always R_i.

Define the policy π as $\pi(a) = \text{softmax}(\theta_i)$ for $i \in [k]$ where $\theta_i \in \mathbb{R}$ is a scalar parameter encoding the value of action a_i. Suppose our learning rate is η. First, show that if action a_i is sampled, then the change in the parameters in REINFORCE is given by

$$\theta_i \leftarrow \theta_i + \eta R_i(1 - \pi(a_i)).$$

Hint: The answer uses the chain rule and log rules.

Second, explain the dynamics of the above change in weights. It may help to think about when the update is large.

Problem 2 (from January 24)

Suppose we are building a deep Q-learning neural network to play your favorite game. We are trying to decide what rewards to assign for different actions. There are two options:

a. We receive reward r_t^a for taking action a_t in state s_t.

b. We receive reward $r_t^b = r_t^a + \delta$ for taking action a_t in state s_t. Here, δ is a constant offset.

Is there a difference in the the Q learning algorithm (described by the pseudo code in class) if we use option b instead of option a? The answer is yes! Your task is to show why.

Recall

$$G_t = \sum_{i=0}^{\infty} r_{t+i}\gamma^i$$

and

$$Q(s,a) = \mathbb{E}[G_t | s_t = s, a_t = a].$$

You should:

- Write out G_t^a and G_t^b (these are the gains when we use the rewards r_t^a and reward r_t^b, respectively).
- Then write out Q_t^b in terms of Q_t^a (these are the Q values when we use r_t^a and reward r_t^b, respectively). You should conclude that $Q^b(s,a) = Q^a(s,a) + C$ for some constant C. Your job is to find this constant!
- Now see if there is a difference in the Q learning algorithm if we use Q_t^b instead of Q_t^a. Referring to the pseudo code for the Q learning algorithm from class, notice that we use the Q function twice (once in step 1 and once in step 3). You should check if there is a difference in both cases.

Problem 3 (from January 25)

In this problem, your task is to modify the demo so that a) we're generating images of FashionMNIST (instead of MNIST) and b) we're using a GAN (instead of a Conditional GAN).

Comment on the difference in quality of the fake images from your FashionMNIST GAN and the MNIST Conditional GAN we wrote in class.

Problem 4 (from January 26)

In class, we considered a *symmetric* matrix $\bar{\mathbf{A}}$. We proved that we can write it as

$$\sum_{i=1}^{n} \lambda_i \mathbf{v}_i \mathbf{v}_i^\top$$

where $0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n \leq 1$ are the eigenvalues and $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ are the corresponding eigenvectors. Remember eigenvectors are *orthonormal*: $\mathbf{v}_i^\top \mathbf{v}_j = 1$ if $i = j$ and 0 otherwise.

In the process of analyzing the Frobenius norm of $\bar{\mathbf{A}}$, we wanted to show that the eigenvalues of $\bar{\mathbf{A}}^\top \bar{\mathbf{A}}$ are $\lambda_1^2 \leq \lambda_2^2 \leq \ldots \lambda_n^2$. Your homework is to show this is true. **Hint**: Use our formulation of $\bar{\mathbf{A}}$ in terms of its eigenvalues and eigenvectors to compute $\bar{\mathbf{A}}^\top \bar{\mathbf{A}}$.