HOMEWORK 8

>>NAME HERE<<
>>ID HERE<<

Instructions:

- Please submit your answers in a single pdf file. Preferably made using latex. No need to submit latex code.

- See piazza post @114 for some recommendations on how to write the answers.

- Submit code for programming exercises. You can use any programming language you like, but we highly recommend python and pytorch.

- Submit all the material on time.

1 Principal Component Analysis [60 pts]

Download three.txt and eight.txt. Each has 200 handwritten digits. Each line is for a digit, vectorized from a 16x16 gray scale image.

1. (10 pts) Each line has 256 numbers: they are pixel values (0=black, 255=white) vectorized from the image as the first column (top down), the second column, and so on. Visualize the two gray scale images corresponding to the first line in three.txt and the first line in eight.txt.

2. (10 pts) Putting the two data files together (threes first, eights next) to form a $n \times D$ matrix X where $n = 400$ digits and $D = 256$ pixels. Note we use $n \times D$ size for X instead of $D \times n$ to be consistent with the convention in linear regression. The ith row of X is $x_i^\top$, where $x_i \in \mathbb{R}^D$ is the ith image in the combined data set. Compute the sample mean $y = \frac{1}{n} \sum_{i=1}^{n} x_i$. Visualize y as a 16x16 gray scale image.

3. (10 pts) Center X using y above. Then form the sample covariance matrix $S = \frac{X^\top X}{n-1}$. Show the 5x5 submatrix $S(1 \ldots 5, 1 \ldots 5)$.

4. (10 pts) Use appropriate software to compute the two largest eigenvalues $\lambda_1 \geq \lambda_2$ and the corresponding eigenvectors v_1, v_2 of S. For example, in Matlab one can use eigs(S,2). Show the value of λ_1, λ_2. Visualize v_1, v_2 as two 16x16 gray scale images. Hint: their elements will not be in [0, 255], but you can shift and scale them appropriately. It is best if you can show an accompany "colorbar" that maps gray scale to values.

5. (10 pts) Now we project (the centered) X down to the two PCA directions. Let $V = [v_1 v_2]$ be the $D \times 2$ matrix. The projection is simply XV. Show the resulting two coordinates for the first line in three.txt and the first line in eight.txt, respectively.

6. (10 pts) Now plot the 2D point cloud of the 400 digits after projection. For visual interest, color points in three.txt red and points in eight.txt blue. But keep in mind that PCA is an unsupervised learning method and it does not know such class labels.

2 Q-learning [40 pts]

Consider the following Markov Decision Process. It has two states s. It has two actions a: move and stay. The state transition is deterministic: "move" moves to the other state, while "stay' stays at the current state. The reward r is 0 for move, 1 for stay. There is a discounting factor $\gamma = 0.8$.
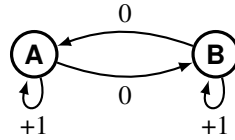
The reinforcement learning agent performs Q-learning. Recall the Q table has entries $Q(s, a)$. The Q table is initialized with all zeros. The agent starts in state $s_1 = A$. In any state s_t, the agent chooses the action a_t according to a behavior policy $a_t = \pi_B(s_t)$. Upon experiencing the next state and reward s_{t+1}, r_t the update is:

$$Q(s_t, a_t) \Leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha \left(r_t + \gamma \max_{a'} Q(s_{t+1}, a') \right).$$

Let the step size parameter $\alpha = 0.5$.

1. (10 pts) Run Q-learning for 200 steps with a deterministic greedy behavior policy: at each state s_t use the best action $a_t \in \arg\max_a Q(s_t, a)$ indicated by the current Q table. If there is a tie, prefer move. Show the Q table at the end.

2. (10 pts) Reset and repeat the above, but with an ϵ-greedy behavior policy: at each state s_t, with probability $1 - \epsilon$ choose what the current Q table says is the best action: $\arg\max_a Q(s_t, a)$; Break ties arbitrarily. Otherwise (with probability ϵ) uniformly chooses between move and stay (move or stay both with 1/2 probability). Use $\epsilon = 0.5$.

3. (20 pts) Without doing simulation, use Bellman equation to derive the true Q table induced by the MDP.