MCE5901 Homework Set 5

November 24, 2023

NOTICE: The homework is due on Dec. 8 (Friday) 11:59pm. **Please draft your solution with steps and provide the codes that you use for the applied problems.** You will need to submit two files to the blackboard: a PDF file including the solutions and results for all problems, and a file containing the codes for the applied problems. You are allowed, and even encouraged, to discuss the homeworks with your classmates. However, you must **write up the solutions on your own**. Plagiarism and other anti-scholarly behavior will be dealt with severely.

Problem 4 is optional. You will receive 20 points bonus if you solve it and the maximum score of the assignment is 100 points.

Problem 1. Neural networks

Consider a fully connected neural network with two hidden layers: $p = 4$ input units, 5 units in the first hidden layer, 3 units in the second hidden layer, and a single output. The hidden units will use the ReLU activation, and for final prediction we will use the sigmoid activation to obtain a prediction between 0 and 1.

(a) Draw a picture of the network.

(b) Show the feed-forward process of the neural network. Write out an expression for $f(X)$ (the relationship between the input and output). Define the weight matrices and bias vectors for each layer clearly. How many parameters are there?

(c) Given a set of training data $\{(x_i, y_i) : i = 1, \ldots, n\}$, write out the loss function using cross-entropy.

Now consider another convolutional neural network that takes in 32×32 grayscale images and has a single convolution layer with **three** 5×5 convolution filters (without boundary padding), followed by a 3×3 pooling layer. The strides for the convolutional layer and pooling layer are 1 and 3 respectively.

(e) Draw a sketch of the input layer, the convolutional layer and the pooling layer. Indicate the size of each layer (i.e., height, width, depth). How many parameters are there?

(f) If we want to make the input and output volume of the convolutional layer have the same spatial dimensions, what is the required amount of zero padding?

(g) Explain how the convolutional layer can be thought of as a fully connected layer with constraints on the weights in the hidden units. What are the constraints?

Problem 2. Clustering

In this problem, you will perform clustering manually on a small example with $n = 6$ observations and $p = 2$ features. The observations are as follows.

Obs.	X_1	X_2
1	1	4
2	1	3
3	0	4
4	5	1
5	6	2
6	4	0

First consider performing K-means clustering, with $K = 2$.

(a) Plot the observations.

(b) Randomly assign a cluster label to each observation. Compute the centroid for each cluster. Assign each observation to the centroid to which it is closest, in terms of Euclidean distance. Repeat until the answers obtained stop changing. Report the cluster labels for each observation in all steps.

(c) Repeat K-means again with a different initialization and comment on what your find.

We continue to work on the previous data set and now consider hierarchical clustering.

(d) Compute a dissimilarity matrix for this six observations using the Euclidean distance metric. The matrix is of dimension 6×6, the entries are the Euclidean distances between any two observations.

(e) On the basis of this dissimilarity matrix, sketch the dendrogram that results from hierarchically clustering these six observations using complete linkage. Be sure to indicate on the plot the height at which each fusion occurs, as well as the observations corresponding to each leaf in the dendrogram.

(f) Repeat (e), this time using single linkage clustering.

(g) Suppose that we cut the dendrogram obtained in (e) such that two clusters result. Which observations are in each cluster? What about we cut the dendrogram obtained in (f) such that three clusters result?

Problem 3. Principal components analysis

Consider the following design matrix, each row representing a sample point $x_i \in \mathbb{R}^2$ for $i = 1, 2, 3, 4$.

$$X = \begin{bmatrix} 4 & 1 \\ 2 & 3 \\ 5 & 4 \\ 1 & 0 \end{bmatrix}$$

We want to represent the data in only one dimension, so we turn to principal components analysis (PCA).

(a) Compute the unit-length principal component directions of X. (Warning: Observe that X is not centered.) Hint: If you graph the points, you can probably guess the principdal component directions (then verify that they really are).

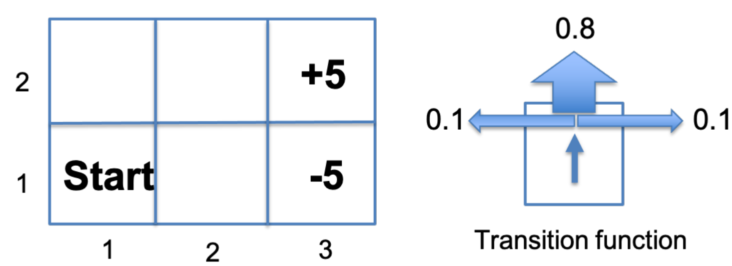
(b) Plot the data after centering. Draw the principal component direction (as a line) and the projections of all four sample points onto the principal direction.

(c) Represent the data in only one dimension and lable their coordinates.

(d) Now we rotate the data samples from X 30 degrees counterclockwise about the origin. Identify the principal component direction and draw it on the plot together with the rotated points. Also draw the projections of the rotated points onto the principal direction. Label each projected point with the exact value of its principal coordinate.

Problem 4. (optional) MDPs and reinforcement learning

This gridworld MDP operates like to the one we saw in class. The states are grid squares, identified by their row and column number (row first). The agent always starts in state $(1,1)$. There are two terminal goal states, $(2,3)$ with reward $+5$ and $(1,3)$ with reward -5. Rewards are 0 in non-terminal states. (The reward for a state is received as the agent moves into the state.) The transition function is such that the intended agent movement (North, South, West, or East) happens with probability 0.8. With probability 0.1 each, the agent ends up in one of the states perpendicular to the intended direction. If a collision with a wall happens, the agent stays in the same state.



(a) Draw the optimal policy for this grid.

(b) Suppose the agent knows the transition probabilities. Give the first two rounds of value iteration updates for each state, with a discount of 0.9. (Assume V_0 is 0 everywhere and compute V_i for times $i = 1, 2$).

(c) Suppose the agent does not know the transition probabilities. What does it need in order to learn the optimal policy?

(d) The agent starts with the policy that always chooses to go right, and executes the following three trials: 1) (1,1)–(1,2)–(1,3), 2) (1,1)–(1,2)–(2,2)–(2,3), and 3) (1,1)– (2,1)–(2,2)–(2,3). What are the monte carlo (direct utility) estimates for states (1,1) and (2,2), given these traces?

(e) Using a learning rate of 0.1 and assuming initial values of 0, what updates does the TD-learning agent make after trials 1 and 2, above?