In the name of GOD.



Sharif University of Technology

Machine Learning

Spring 2025

Hamid R. Rabiee, Zahra Dehghanian

Homework 6 | Deadline: 1404/03/20

1. Consider the following neural network. Single-circled nodes denote variables (e.g., x_1 is an input variable, h_1 is an intermediate variable, $\hat{y}$ is an output variable), and double-circled nodes denote functions (e.g., Σ takes the sum of its inputs, and σ denotes the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$). In the network below,

$$h_1 = \frac{1}{1 + e^{-x_1 w_1 - x_2 w_2}}.$$

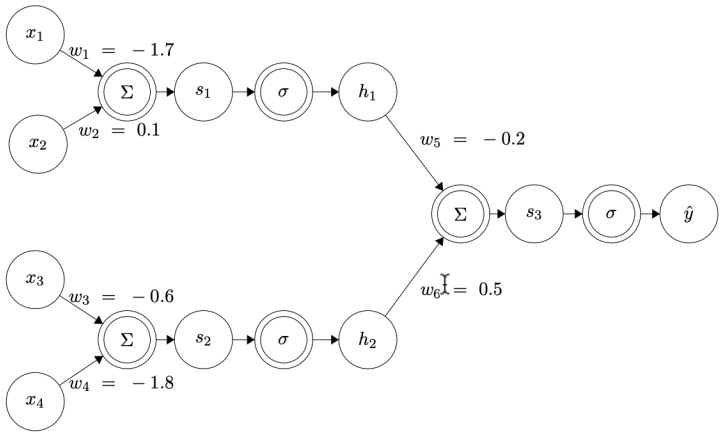


Figure 1: Neural Net architecture

Suppose we use an L2 loss function:

$$L(y, \hat{y}) = \|y - \hat{y}\|_2^2.$$

We are given a data point $(x_1, x_2, x_3, x_4) = (-0.7,\ 1.2,\ 1.1,\ -2)$ with true label $y = 0.5$.
Use the backpropagation algorithm to compute the partial derivative: $\frac{\partial L}{\partial w_1}$.
Solution:

Computing the following values during the forward pass:

$$\begin{aligned} s_1 &= 1.31 \\ s_2 &= 2.94 \\ h_1 &= 0.7875 \\ h_2 &= 0.9498 \\ s_3 &= 0.3173 \\ \hat{y} &= 0.5787 \end{aligned}$$

Using backpropagation, we compute the partial derivative of the loss E with respect to weight w_1:

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial s_3} \cdot \frac{\partial s_3}{\partial h_1} \cdot \frac{\partial h_1}{\partial s_1} \cdot \frac{\partial s_1}{\partial w_1}$$

Substituting the values:

$$= 2||\hat{y} - y|| \cdot \sigma'(s_3) \cdot w_5 \cdot \sigma'(s_1) \cdot x_1$$

$$= 2(0.5787 - 0.5) \cdot 0.2438 \cdot (-0.2) \cdot 0.1673 \cdot (-0.7)$$

$$= 0.0008988$$

2. The standard cost function for k-means clustering is defined as:

$$L = \sum_{j=1}^{k} \sum_{x_i \in S_j} ||x_i - \mu_j||_2^2$$

where S_j is the set of data points x_i assigned to cluster j, and μ_j is the centroid of cluster j.

Answer the following:

(a) **Optimality of the Centroid and Distance Metrics:**

i. Prove that for a given set of points S_j assigned to a cluster j, the unique vector μ_j that minimizes $\sum_{x_i \in S_j} ||x_i - \mu_j||_2^2$ is the sample mean of the points in S_j.

ii. How would the optimal cluster center μ_j change if the L_2 norm (Euclidean distance squared) in the cost function were replaced with the L_1 norm (Manhattan distance), i.e., $L' = \sum_{j=1}^{k} \sum_{x_i \in S_j} ||x_i - \mu_j||_1$? Provide a characterization of this new optimal center (e.g., geometric median).

(b) **Convergence and Initialization:**

i. The k-means objective function L is non-convex. Explain the implications of this non-convexity for the convergence of the algorithm. Why does the standard k-means algorithm (Lloyd's algorithm) guarantee convergence to a local minimum but not necessarily the global minimum?

(c) **Relationship to Gaussian Mixture Models (GMMs):**

i. What specific assumptions must be made about the covariance matrices and mixing coefficients of a GMM for its EM algorithm to effectively reduce to the k-means algorithm?

ii. Explain how the "hard" assignment of k-means (each point belongs to exactly one cluster) contrasts with the "soft" assignment in GMMs.

Solution:

a) Optimality of the Centroid and Distance Metrics

[label=)]**Proof for L_2 norm minimization:** Let $S_j = \{x_1, x_2, \ldots, x_m\}$ be the set of m points assigned to cluster j. We want to find μ_j that minimizes the sum of squared Euclidean distances for this cluster:

$$f(\mu_j) = \sum_{p=1}^{m} ||x_p - \mu_j||_2^2 = \sum_{p=1}^{m} (x_p - \mu_j)^T (x_p - \mu_j)$$

To find the minimum, we compute the gradient of $f(\mu_j)$ with respect to μ_j and set it to zero.

$$\begin{aligned}
\nabla_{\mu_j} f(\mu_j) &= \nabla_{\mu_j} \sum_{p=1}^{m} (x_p - \mu_j)^T (x_p - \mu_j) \\
&= \sum_{p=1}^{m} -2(x_p - \mu_j) \\
&= -2 \sum_{p=1}^{m} (x_p - \mu_j)
\end{aligned}$$

Setting the gradient to zero:

$$\begin{aligned}
-2 \sum_{p=1}^{m} (x_p - \mu_j) &= 0 \\
\sum_{p=1}^{m} (x_p - \mu_j) &= 0 \\
\sum_{p=1}^{m} x_p - \sum_{p=1}^{m} \mu_j &= 0 \\
\sum_{p=1}^{m} x_p - m\mu_j &= 0 \\
\mu_j &= \frac{1}{m} \sum_{p=1}^{m} x_p
\end{aligned}$$

This is the sample mean. To confirm it is a minimum, we examine the Hessian matrix:

$$\nabla_{\mu_j}^2 f(\mu_j) = \nabla_{\mu_j} \left(-2 \sum_{p=1}^{m} (x_p - \mu_j) \right) = -2 \sum_{p=1}^{m} (-I) = 2mI$$

where I is the identity matrix. Since $m > 0$, $2mI$ is positive definite, so the sample mean is the unique global minimum.

(b) **Optimal center for L_1 norm:**

If we replace the squared L_2 norm with the L_1 norm, the objective for a single cluster j with points $x_p = (x_{p1}, \ldots, x_{pd})$ and center $\mu_j = (\mu_{j1}, \ldots, \mu_{jd})$ becomes minimizing:

$$g(\mu_j) = \sum_{x_p \in S_j} ||x_p - \mu_j||_1 = \sum_{x_p \in S_j} \sum_{l=1}^{d} |x_{pl} - \mu_{jl}|$$

This sum can be decoupled by dimension:

$$g(\mu_j) = \sum_{l=1}^{d} \left(\sum_{x_p \in S_j} |x_{pl} - \mu_{jl}| \right)$$

To minimize $g(\mu_j)$, we minimize $\sum_{x_p \in S_j} |x_{pl} - \mu_{jl}|$ for each dimension l independently. The value μ_{jl} that minimizes the sum of absolute deviations from a set of scalar values $\{x_{1l}, x_{2l}, \ldots, x_{ml}\}$ is the **median** of those values. Therefore, the optimal cluster center μ_j would be the **component-wise median** of the points in S_j.

b) Convergence and Initialization

(a) [label=)]**Implications of non-convexity and local vs. global convergence:** The k-means objective function L is non-convex. **Implications of non-convexity**:

- **Multiple Local Minima**: The objective function landscape has multiple local minima.
- **Dependence on Initialization**: The specific local minimum found depends on the initial centroids.
- **No Guarantee of Global Optimum**: The algorithm might not find the best possible clustering.

Why Lloyd's algorithm guarantees local convergence but not global: Lloyd's algorithm alternates between two steps:

i. **Assignment Step**: Each point is assigned to the closest current centroid. This step decreases or keeps L the same for fixed μ_j.

ii. **Update Step**: Each centroid is recalculated as the mean of its assigned points. This step decreases or keeps L the same for fixed assignments S_j.

Since both steps ensure L is non-increasing and L is bounded below by 0, the algorithm converges. However, it converges to a state where neither step can further reduce L, which is a **local minimum**, not necessarily the global one, due to the non-convex landscape.

c) Relationship to Gaussian Mixture Models (GMMs)

(a) [label=)]**Assumptions for GMM to reduce to k-means:** K-means can be seen as a special case of the EM algorithm for GMMs under these assumptions:

- **Spherical Covariance Matrices**: All Gaussian components have covariance matrices $\Sigma_j = \sigma^2 I$.
- **Equal and Fixed Variance**: The variance σ^2 is the same for all k components.

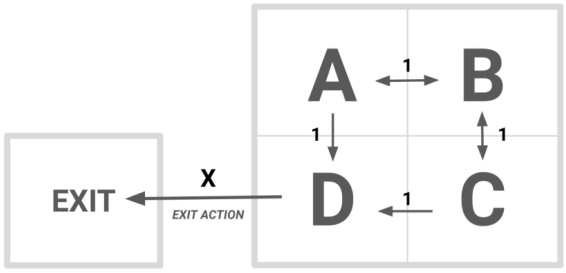
- **Equal Mixing Coefficients (Priors)**: The mixing coefficients π_j are equal for all components ($\pi_j = 1/k$).

Under these conditions, especially as $\sigma^2 \to 0$, the posterior probability $p(j|x_i)$ becomes 1 for the cluster whose mean μ_j is closest to x_i (in Euclidean distance), leading to hard assignments. The M-step for updating means μ_j becomes identical to the k-means centroid update.

(b) **"Hard" assignment of k-means vs. "soft" assignment in GMMs:**

- **K-means (Hard Assignment)**: Each data point x_i is assigned exclusively to **one** cluster S_j. Membership is binary (0 or 1). The sets S_j form a partition of the data.
- **GMMs (Soft Assignment)**: Each data point x_i has a **probability** (or responsibility) r_{ij} of belonging to each cluster j, where $r_{ij} \in [0, 1]$ and $\sum_{j=1}^{k} r_{ij} = 1$. A point can partially belong to multiple clusters, reflecting uncertainty. Centroids and other GMM parameters are updated using these responsibilities as weights.

3. Consider below Markov Decision Process (MDP) with four states: A, B, C, and D. From states A, B, and C, the agent can choose from the actions: `LEFT`, `RIGHT`, `UP`, or `DOWN`, unless obstructed by a wall in that direction. From state D, the only available action is a special `EXIT` action, which grants the agent a terminal reward of x. All other actions (non-exit actions) yield a reward of 1.



(a) **(Deterministic actions)**
Assume all actions are deterministic, and the discount factor is $\gamma = \frac{1}{2}$. Express the value function $V^*(s)$ for the following states in terms of x, given that the optimal value function satisfies:

$$V^*(D) = \qquad V^*(A) =$$
$$V^*(C) = \qquad V^*(B) =$$

(b) **(Stochastic actions)**
Now suppose each non-exit action succeeds with probability $\frac{1}{2}$; otherwise, the agent remains in the same state and receives a reward of 0. The `EXIT` action from state D is still deterministic and always succeeds. Let $\gamma = \frac{1}{2}$ as before.

Find the value of x for which the agent is indifferent between two actions from state A: taking action `DOWN` to go to D, and taking action `RIGHT` to go to B. In other words, solve for x such that:

$$Q^*(A, \texttt{DOWN}) = Q^*(A, \texttt{RIGHT})$$

Solution:

(a)

$$
\begin{aligned}
V^*(D) &= x \\
V^*(A) &= \max\left(1 + \frac{1}{2}x,\ 2\right) \\
V^*(C) &= \max\left(1 + \frac{1}{2}x,\ 2\right) \\
V^*(B) &= \max\left(1 + \frac{1}{2}\left(1 + \frac{1}{2}x\right),\ 2\right)
\end{aligned}
$$

Note: The value 2 corresponds to the infinite sum of discounted rewards equal to 1, calculated as $\frac{1}{1-\gamma} = 2$.

(b)

$$Q^*(A, \texttt{DOWN}) = \frac{1}{2}\left(0 + \frac{1}{2}V^*(A)\right) + \frac{1}{2}\left(1 + \frac{1}{2}x\right) = \frac{1}{2} + \frac{1}{4}V^*(A) + \frac{1}{4}x \quad (1)$$

$$Q^*(A, \texttt{RIGHT}) = \frac{1}{2}\left(0 + \frac{1}{2}V^*(A)\right) + \frac{1}{2}\left(1 + \frac{1}{2}V^*(B)\right) = \frac{1}{2} + \frac{1}{4}V^*(A) + \frac{1}{4}V^*(B) \quad (2)$$

$$\Rightarrow V^*(A) = \frac{2}{3} + \frac{1}{3}x \quad \text{(from (1))}$$

$$\Rightarrow V^*(A) = \frac{2}{3} + \frac{1}{3}V^*(B) \quad \text{(from (2))}$$

From symmetry, let $V^*(B) = Q^*(B, \texttt{LEFT})$ or $Q^*(B, \texttt{DOWN})$:

$$V^*(B) = \frac{1}{2}\left(0 + \frac{1}{2}V^*(B)\right) + \frac{1}{2}\left(1 + \frac{1}{2}V^*(A)\right) = \frac{1}{2} + \frac{1}{4}V^*(B) + \frac{1}{4}V^*(A) \quad (3)$$

$$\Rightarrow V^*(B) = \frac{2}{3} + \frac{1}{3}V^*(A) \quad (4)$$

Substituting Equation (2) into Equation (4):

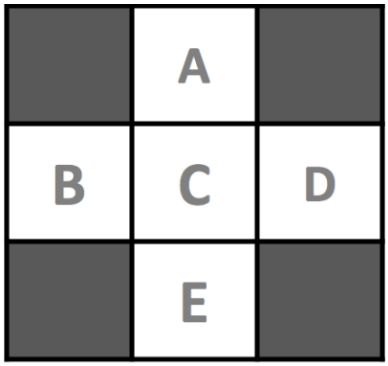
$$
\begin{aligned}
V^*(A) &= \frac{2}{3} + \frac{1}{3}V^*(B) \\
&= \frac{2}{3} + \frac{1}{3}\left(\frac{2}{3} + \frac{1}{3}V^*(A)\right) \\
&= \frac{2}{3} + \frac{2}{9} + \frac{1}{9}V^*(A) \\
\Rightarrow V^*(A) - \frac{1}{9}V^*(A) &= \frac{8}{9} \\
\Rightarrow \frac{8}{9}V^*(A) &= \frac{8}{9} \\
\Rightarrow V^*(A) &= 1
\end{aligned}
$$

Plugging into Equation (1):

$$1 = \frac{2}{3} + \frac{1}{3}x \Rightarrow x = 1$$

Final Answer: $x = 1$

4. Consider a Gridworld scenario where an agent aims to estimate the value function of each state using TD Learning and Q-Learning.



Suppose we observe the following $(s, a, s', R(s, a, s'))$ transitions and rewards (in order from left to right):

$$\text{(B, East, C, 2)}, \quad \text{(C, South, E, 4)}, \quad \text{(C, East, A, 6)}, \quad \text{(B, East, C, 2)}$$

Assume the initial value of each state is 0, the discount factor $\gamma = 1$, and the learning rate $\alpha = 0.5$.

(a) What are the learned state values from TD learning after all four transitions?

(b) What are the learned Q-values from Q-learning after all four observations? Use the same $\alpha = 0.5$ and $\gamma = 1$.

Solution:

(a) TD Learning updates the state value function $V(s)$ using the rule:

$$V(s) \leftarrow V(s) + \alpha \left(R(s, a, s') + \gamma V(s') - V(s)\right)$$

$$\begin{aligned}
\text{After (B, East, C, 2):} \quad & V(B) = 0 + 0.5(2 + 1 \cdot 0 - 0) = 1 \\
\text{After (C, South, E, 4):} \quad & V(C) = 0 + 0.5(4 + 1 \cdot 0 - 0) = 2 \\
\text{After (C, East, A, 6):} \quad & V(C) = 2 + 0.5(6 + 1 \cdot 0 - 2) = 4 \\
\text{After (B, East, C, 2):} \quad & V(B) = 1 + 0.5(2 + 1 \cdot 4 - 1) = 3.5
\end{aligned}$$

Final learned values:

$$V(B) = 3.5, \quad V(C) = 4, \quad V(s) = 0 \quad \text{for all } s \in \{A, D, E\}$$

(b) Q-Learning updates the action-value function $Q(s, a)$ using the rule:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left(R(s, a, s') + \gamma \max_{a'} Q(s', a') - Q(s, a)\right)$$

$$\text{Initial:} \quad Q(s,a) = 0 \quad \text{for all state-action pairs}$$

$$\text{After (B, East, C, 2):} \quad Q(B,\text{East}) = 0 + 0.5 \cdot (2 + 1 \cdot 0 - 0) = 1$$

$$\text{After (C, South, E, 4):} \quad Q(C,\text{North}) = 0 + 0.5 \cdot (4 + 1 \cdot 0 - 0) = 2$$

$$\text{After (C, East, A, 6):} \quad Q(C,\text{West}) = 0 + 0.5 \cdot (6 + 1 \cdot 0 - 0) = 3$$

$$\text{After (B, East, C, 2):} \quad \max_{a'} Q(C,a') = \max(2,3) = 3$$
$$Q(B,\text{East}) = 1 + 0.5 \cdot (2 + 1 \cdot 3 - 1) = 3$$

Final Q-Values:

$$Q(B,\text{East}) = 3, \quad Q(C,\text{South}) = 2, \quad Q(C,\text{East}) = 3$$

$$Q(s,a) = 0 \quad \text{for all other } (s,a)$$