CLO 2 - Neural Networks and Logistic Regression

Question 1: True/False

Instructions: Identify whether the statements are True or False. Reasoning is provided.

1. A deep neural network has many hidden layers, making it capable of learning complex and hierarchical features from data.
   Answer: True.
   Reasoning: Deep neural networks (DNNs) have multiple hidden layers, allowing them to learn hierarchical features — simple features in early layers and complex patterns in deeper layers.

2. In backpropagation neural networks, information flows in one direction, from input to output layers, without any loops or cycles.
   Answer: True.
   Reasoning: This describes feedforward networks. Backpropagation updates weights, but the forward pass has no loops.

3. The coefficients learned in Logistic Regression indicate the impact of each feature on the log-odds of the target variable.
   Answer: True.
   Reasoning: Logistic Regression models the log-odds as a linear function of features; coefficients reflect the change in log-odds per unit feature change.

4. A small learning rate may lead to slow convergence in gradient descent, while a large learning rate may lead to overshooting the minimum.
   Answer: True.
   Reasoning: Learning rate controls step size. Too small $\rightarrow$ slow; too large $\rightarrow$ may diverge.

5. Gradient descent is guaranteed to find the global minimum of a cost function for any given problem.
   Answer: False.
   Reasoning: For non-convex functions (like deep neural network loss functions), gradient descent may get stuck in local minima.

Question 2: Single Neuron Gradient Descent using Logistic Regression

Given

- Inputs (three examples):

$$x_1 = [0.2, 0.4], \quad x_2 = [0.4, 0.5], \quad x_3 = [0.6, 0.2]$$

- Initial weights:

$$\theta = [\theta_1, \theta_2] = [0.5,\ 0.5]$$

- Learning rate: $\alpha = 0.1$
- Activation: sigmoid

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \qquad z = \theta_1 x_1 + \theta_2 x_2$$

- Targets: $t = [1, 1, 1]$ (binary)
- Loss (binary cross-entropy), averaged over n examples:

$$J(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \big[t_i \log(h_i) + (1 - t_i) \log(1 - h_i) \big]$$

- Gradient formula (for each weight θ_j):

$$\frac{\partial J}{\partial \theta_j} = \frac{1}{n} \sum_{i=1}^{n} (h_i - t_i) x_{ij}$$

where $h_i = \sigma(z_i)$ and $z_i = \theta_1 x_{i1} + \theta_2 x_{i2}$.

We will perform **three** gradient-descent iterations and show all intermediate values.

Note on sign: Because $t_i = 1$ and $h_i < 1$, the terms $(h_i - t_i)$ are negative, giving negative gradients. Updating with $\theta \leftarrow \theta - \alpha \nabla_\theta J$ therefore *increases* the weights.

Initial linear combinations and activations (Step 0)

Using $\theta = [0.5, 0.5]$:

$$\begin{aligned}
z_1 &= 0.5 \cdot 0.2 + 0.5 \cdot 0.4 = 0.3, \\
z_2 &= 0.5 \cdot 0.4 + 0.5 \cdot 0.5 = 0.45, \\
z_3 &= 0.5 \cdot 0.6 + 0.5 \cdot 0.2 = 0.4.
\end{aligned}$$

Apply sigmoid:

$$\begin{aligned}
h_1 &= \sigma(0.3) \approx 0.5744425168, \\
h_2 &= \sigma(0.45) \approx 0.6106392339, \\
h_3 &= \sigma(0.4) \approx 0.5986876601.
\end{aligned}$$

The initial loss (with $t_i = 1$) is

$$J^{(0)} = -\frac{1}{3}\sum_{i=1}^{3}\log(h_i) \approx -\frac{1}{3}\big(\log 0.5744425 + \log 0.6106392 + \log 0.5986877\big) \approx 0.52020648.$$

Iteration 1

Gradients (using $n = 3$ and $t_i = 1$):

$$\frac{\partial J}{\partial \theta_1} = \frac{1}{3}\sum_{i=1}^{3}(h_i - 1)\,x_{i1} = \frac{1}{3}\big[(0.5744425 - 1)\cdot 0.2 + (0.6106392 - 1)\cdot 0.4 + (0.59868766 - 1)\cdot 0.6\big]$$
$$\approx -0.1605477357,$$

$$\frac{\partial J}{\partial \theta_2} = \frac{1}{3}\sum_{i=1}^{3}(h_i - 1)\,x_{i2} = \frac{1}{3}\big[(0.5744425 - 1)\cdot 0.4 + (0.6106392 - 1)\cdot 0.5 + (0.59868766 - 1)\cdot 0.2\big]$$
$$\approx -0.1483886148.$$

Weight update

$$\theta_1 \leftarrow 0.5 - 0.1\cdot(-0.1605477357) \approx 0.5160547736,$$
$$\theta_2 \leftarrow 0.5 - 0.1\cdot(-0.1483886148) \approx 0.5148388615.$$

Loss after update (still reporting using the h_i computed before update): $J^{(1,\text{before next forward})} = 0.52020648$ as above; we recompute loss after the next forward pass below.

Iteration 2

Forward pass with updated $\theta = [0.51605477, 0.51483886]$. Compute z and h:

$$z_1 \approx 0.5160547736\cdot 0.2 + 0.5148388615\cdot 0.4 \approx 0.3091464993,$$
$$z_2 \approx 0.5160547736\cdot 0.4 + 0.5148388615\cdot 0.5 \approx 0.4638413402,$$
$$z_3 \approx 0.5160547736\cdot 0.6 + 0.5148388615\cdot 0.2 \approx 0.4126006364.$$

$$h_1 \approx \sigma(0.3091464993) \approx 0.5766769176,$$
$$h_2 \approx \sigma(0.4638413402) \approx 0.6139250522,$$
$$h_3 \approx \sigma(0.4126006364) \approx 0.6017112984.$$

Gradients

$$\frac{\partial J}{\partial \theta_1} \approx \frac{1}{3}\big[(0.5766769176 - 1)\cdot 0.2 + (0.6139250522 - 1)\cdot 0.4 + (0.6017112984 - 1)\cdot 0.6\big]$$
$$\approx -0.1593559388,$$

$$\frac{\partial J}{\partial \theta_2} \approx \frac{1}{3}\big[(0.5766769176 - 1)\cdot 0.4 + (0.6139250522 - 1)\cdot 0.5 + (0.6017112984 - 1)\cdot 0.2\big]$$
$$\approx -0.1473414824.$$

Weight update

$$\theta_1 \leftarrow 0.5160547736 - 0.1 \cdot (-0.1593559388) \approx 0.5319903675,$$
$$\theta_2 \leftarrow 0.5148388615 - 0.1 \cdot (-0.1473414824) \approx 0.5295730097.$$

Loss after forward pass

$$J^{(2)} = -\frac{1}{3}\sum_{i=1}^{3} \log(h_i) \approx -\frac{1}{3}\big(\log 0.5766769 + \log 0.6139251 + \log 0.6017113\big) \approx 0.5154443488.$$

Iteration 3

Forward pass with updated $\theta = [0.5319903675, 0.5295730097]$. Compute z and h:

$$z_1 \approx 0.5319903675 \cdot 0.2 + 0.5295730097 \cdot 0.4 \approx 0.3182272774,$$
$$z_2 \approx 0.5319903675 \cdot 0.4 + 0.5295730097 \cdot 0.5 \approx 0.4775826518,$$
$$z_3 \approx 0.5319903675 \cdot 0.6 + 0.5295730097 \cdot 0.2 \approx 0.4251088224.$$

$$h_1 \approx \sigma(0.3182272774) \approx 0.5788921654,$$
$$h_2 \approx \sigma(0.4775826518) \approx 0.6171768909,$$
$$h_3 \approx \sigma(0.4251088224) \approx 0.6047050976.$$

Gradients

$$\frac{\partial J}{\partial \theta_1} \approx -0.1581759173,$$
$$\frac{\partial J}{\partial \theta_2} \approx -0.1463045563.$$

Weight update

$$\theta_1 \leftarrow 0.5319903675 - 0.1 \cdot (-0.1581759173) \approx 0.5478079592,$$
$$\theta_2 \leftarrow 0.5295730097 - 0.1 \cdot (-0.1463045563) \approx 0.5442034653.$$

Loss after update

$$J^{(3)} = -\frac{1}{3}\sum_{i=1}^{3} \log(h_i) \approx -\frac{1}{3}\big(\log 0.5788922 + \log 0.6171769 + \log 0.6047051\big) \approx 0.5107510148.$$

Summary table (rounded)

Iteration	θ_1	θ_2	Loss $J(\theta)$
0 (initial)	0.500000	0.500000	0.52020648
1	0.51605477	0.51483886	— (recomputed below)
2	0.53199037	0.52957301	0.51544435
3	0.54780796	0.54420347	0.51075101

Observations and intuition

- Since targets are 1, the prediction errors $(h_i - t_i)$ are negative (because $h_i < 1$), so gradients are negative and the gradient-descent update increases the weights.
- As θ increases, the linear score z_i for each example increases, the sigmoid outputs h_i move closer to 1, and the cross-entropy loss (which penalizes $\log(1/h)$ for target 1) decreases.
- The change per iteration is small here (small learning rate $\alpha = 0.1$ combined with small feature magnitudes), so convergence is gradual and stable.