Week 2 written assignment

鄭凱謙

1 問題 1

Read《 Deep Learning: An Introduction for Applied Mathematicians 》. Consider a network as defined in (3.1) and (3.2). Assume that $n_L = 1$, find an algorithm to calculate $\nabla a^{[L]}(\mathbf{x})$

1.1 神經網路定義

根據《 Deep Learning: An Introduction for Applied Mathematicians 》，一個具有 L 層的神經網路定義如下：

- 輸入層 $(l = 1)$**:** 網路的輸入為 $\mathbf{x} \in \mathbb{R}^{n_1}$，第一層的 activation 為 $\mathbf{a}^{[1]} = \mathbf{x}$。
- 隱藏層與輸出層 $(l = 2, \ldots, L)$**:** 第 l 層的激活值 $\mathbf{a}^{[l]}$ 是由前一層的 activation $\mathbf{a}^{[l-1]}$ 計算而來。

其計算方式為：

$$\mathbf{z}^{[l]} = \mathbf{W}^{[l]}\mathbf{a}^{[l-1]} + \mathbf{b}^{[l]} \tag{1}$$

$$\mathbf{a}^{[l]} = \sigma(\mathbf{z}^{[l]}) \tag{2}$$

其中：

- $\mathbf{W}^{[l]} \in \mathbb{R}^{n_l \times n_{l-1}}$ 是第 l 層的權重矩陣。
- $\mathbf{b}^{[l]} \in \mathbb{R}^{n_l}$ 是第 l 層的 bias 向量。
- $\mathbf{z}^{[l]}$ 是第 l 層的加權輸入 (weighted input)。
- $\sigma(\cdot)$ 為 Sigmoid 激活函數，此處以 component-wise 方式作用於向量上。其導數形式為 $\sigma'(z) = \sigma(z)(1 - \sigma(z))$。

作業的假設為輸出層只有一個神經元，即 $n_L = 1$，因此 $a^{[L]}$ 為一個純量 (scalar)。

1.2 演算法推導

我們的目標是計算 $\nabla_{\mathbf{x}} a^{[L]}(\mathbf{x})$，由於 $\mathbf{x} = \mathbf{a}^{[1]}$，這等同於計算 $\nabla_{\mathbf{a}^{[1]}} a^{[L]}$。我們將使用 chain rule 從最後一層反向推導至第一層。

首先，我們定義 $\boldsymbol{\Delta}^{[l]}$ 為最終輸出 $a^{[L]}$ 對第 l 層 activation $\mathbf{a}^{[l]}$ 的梯度：

$$\boldsymbol{\Delta}^{[l]} := \nabla_{\mathbf{a}^{[l]}} a^{[L]}$$

根據連鎖律，相鄰兩層的梯度關係可以表示為：

$$\boldsymbol{\Delta}^{[l-1]} = \left(\frac{\partial \mathbf{a}^{[l]}}{\partial \mathbf{a}^{[l-1]}} \right)^{\top} \boldsymbol{\Delta}^{[l]}$$

其中 $\dfrac{\partial \mathbf{a}^{[l]}}{\partial \mathbf{a}^{[l-1]}}$ 是第 l 層輸出的 Jacobian 矩陣。我們可以再次利用連鎖律來計算此 Jacobian：

$$\frac{\partial \mathbf{a}^{[l]}}{\partial \mathbf{a}^{[l-1]}} = \frac{\partial \mathbf{a}^{[l]}}{\partial \mathbf{z}^{[l]}} \frac{\partial \mathbf{z}^{[l]}}{\partial \mathbf{a}^{[l-1]}}$$

從公式 (1) 和 (2) 可知：

- $\dfrac{\partial \mathbf{z}^{[l]}}{\partial \mathbf{a}^{[l-1]}} = \mathbf{W}^{[l]}$
- $\dfrac{\partial \mathbf{a}^{[l]}}{\partial \mathbf{z}^{[l]}}$ 是一個對角矩陣，我們記為 $\mathbf{D}^{[l]} = \operatorname{diag}(\sigma'(\mathbf{z}^{[l]}))$。

因此，Jacobian 矩陣為 $\dfrac{\partial \mathbf{a}^{[l]}}{\partial \mathbf{a}^{[l-1]}} = \mathbf{D}^{[l]} \mathbf{W}^{[l]}$。

將此結果代入梯度關係式中，得到遞迴公式：

$$\boldsymbol{\Delta}^{[l-1]} = (\mathbf{D}^{[l]} \mathbf{W}^{[l]})^{\top} \boldsymbol{\Delta}^{[l]} = (\mathbf{W}^{[l]})^{\top} (\mathbf{D}^{[l]})^{\top} \boldsymbol{\Delta}^{[l]}$$

因為 $\mathbf{D}^{[l]}$ 是對角矩陣，所以 $(\mathbf{D}^{[l]})^{\top} = \mathbf{D}^{[l]}$。最終的遞迴公式為：

$$\boldsymbol{\Delta}^{[l-1]} = (\mathbf{W}^{[l]})^{\top} \mathbf{D}^{[l]} \boldsymbol{\Delta}^{[l]}$$

這個反向傳播過程需要一個起始條件。在輸出層 $l = L$，因為 $a^{[L]}$ 是一個純量，它對自身的梯度為 1。

$$\boldsymbol{\Delta}^{[L]} = \nabla_{a^{[L]}} a^{[L]} = 1$$

1.3 演算法流程

基於上述推導，我們可以將整個計算過程總結為一個包含前向傳播和後向傳播的演算法。

Algorithm 1 計算神經網路輸出梯度 $\nabla a^{[L]}(\mathbf{x})$

輸入：網路權重 $\mathbf{W}^{[l]}$，bias $\mathbf{b}^{[l]}$ (for $l = 2, \ldots, L$)，以及輸入向量 $\mathbf{x}$。

輸出：梯度 $\nabla a^{[L]}(\mathbf{x})$。

第一步：前向傳播 (Forward Propagation)

設定 $\mathbf{a}^{[1]} \leftarrow \mathbf{x}$。

for $l = 2, \ldots, L$ **do**

  $\mathbf{z}^{[l]} \leftarrow \mathbf{W}^{[l]}\mathbf{a}^{[l-1]} + \mathbf{b}^{[l]}$

  $\mathbf{a}^{[l]} \leftarrow \sigma(\mathbf{z}^{[l]})$

end for

第二步：梯度後向傳播 (Backward Propagation)

初始化 $\mathbf{\Delta}^{[L]} \leftarrow 1$。

for $l = L, \ldots, 2$ **do**

  計算導數矩陣 $\mathbf{D}^{[l]} \leftarrow \mathrm{diag}(\sigma'(\mathbf{z}^{[l]})) = \mathrm{diag}(\mathbf{a}^{[l]} \circ (1 - \mathbf{a}^{[l]}))$。

  更新梯度 $\mathbf{\Delta}^{[l-1]} \leftarrow (\mathbf{W}^{[l]})^\top \mathbf{D}^{[l]} \mathbf{\Delta}^{[l]}$。

end for

return $\mathbf{\Delta}^{[1]}$。

2 問題 2

上課提到在處理分類問題時，可以將類別用 one-hot encoding 如 [1, 0] 和 [0, 1] 來表示，並使用 MSE 作為損失函數來訓練模型。然而，在分類任務中，交叉熵 (cross-entropy) 通常是更常見的選擇。請問，選擇 MSE 而非交叉熵來處理分類任務，在理論或實務上有什麼樣的影響？