Machine Learning HW2

Student ID: 111652015

Problem 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 $\nabla a^{[L]}(x)$.

(3.1) $a^{[1]} = x \in \mathbb{R}^{n_1}$.

(3.2) $a^{[\ell]} = \sigma\big(W^{[\ell]} a^{[\ell-1]} + b^{[\ell]}\big) \in \mathbb{R}^{n_\ell} \quad \text{for } \ell \in \{2, 3, \dots, L\}$.

$$\nabla a^{[L]}(x) = \left(\frac{\partial a^{[L]}}{\partial x}\right)^\top = \left(\frac{\partial a^{[L]}}{\partial a^{[L-1]}} \cdot \frac{\partial a^{[L-1]}}{\partial a^{[L-2]}} \cdots \frac{\partial a^{[2]}}{\partial a^{[1]}}\right)^\top .$$

$$\text{Let} \quad J_\ell = \frac{\partial a^{[\ell]}}{\partial a^{[\ell-1]}} = \operatorname{diag}\big(\sigma'(z^{[\ell]})\big)\, W^{[\ell]}.$$

$$\nabla a^{[L]}(x) = \big(J_L J_{L-1} \cdots J_2\big)^\top = \big(J_2^\top J_3^\top \cdots J_L^\top\big) \in \mathbb{R}^{n_1}.$$

Problem 2 There are unanswered questions during the lecture, and there are likely more questions we haven't covered. Take a moment to think about them and write them down here.

When using a neural network to approximate a function, is the approximation error inversely proportional to the number of neurons?