✍ Written assignment

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

1) define network background

input layer: $a^{[1]} = x \in \mathbb{R}^{n_1}$

pre-activation: $z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]} \in \mathbb{R}^{n_l}$

hidden layer: $a^{[l]} = \sigma(z^{[l]}) \in \mathbb{R}^{n_l}$, $l = 2 \cdots L$

output layer: $a^{[L]} \in \mathbb{R}^{n_L}$

$W^{[l]} \in \mathbb{R}^{n_l \times n_{l-1}}$, weight matrix

$b^{[l]} \in \mathbb{R}^{n_l}$, lth layor vector

Loss function (MSE): $C = \frac{1}{2}(a^{[L]} - y)^2$

y: target

2) define backprogation algorithm

A: Forward pass

1. Initial input: $a^{[1]} = x$

2、For each layer $l = 2, \cdots L$

linear combination : $z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]}$

apply activation function : $a^{[l]} = \sigma(z^{[l]})$

B: backward Pass :

1、compute output layer error :

$$\delta^{[L]} = (a^{[L]} - y) \cdot \sigma'(z^{[L]})$$

2、for hidden layer $l = L-1, \cdots, 2$

$$\delta^{[l]} = (W^{[l+1]})^T \delta^{[l+1]} \odot \sigma'(z^{[l]})$$

-by gpt

C: Gradient Computation

for each layer $l = 2, \cdots, L$

weights : $\dfrac{\partial C}{\partial W^{[l]}} = \delta^{[l]} (a^{[l-1]})^T$

biases : $\dfrac{\partial C}{\partial b^{[l]}} = \delta^{[l]}$

4) Answer

$\delta^{[L]} = \sigma'(z^{[L]})$, $\delta^{[l]} = (W^{[l+1]})^T \delta^{[l+1]} \odot \sigma'(z^{[l]})$

$$\nabla_x a^{[L]}(x) = (W^{[2]})^T \delta^{[2]}$$ #

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.

If the errors aren’t Gaussian but skewed or heavy-tailed, the usual t- or \chi^2-based confidence intervals might fail. Would robust methods like sandwich estimators or bootstrap give more reliable results?

Poorly conditioned weight matrices can make gradients blow up or vanish. Could normalization or residual connections help keep \nabla_x a^{[L]}(x) stable?
discussion with gpt