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

$a^{[1]} = x \in \mathbb{R}^{n_1}$, $a^{[l]} = \sigma(W^{[l]} a^{[l-1]} + b^{[l]}) \in \mathbb{R}^{n_l}$ for $l = 2, 3, \cdots, L$

Let $z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]}$ => $a^{[l]} = \sigma(z^{[l]}) \in \mathbb{R}^{n_l}$

Then $z^{[L]} = W^{[L]} a^{[L-1]} b^{[L]} \in \mathbb{R}^1$

=> $a^{[L]} = \sigma(z^{[L]}) \in \mathbb{R}$ for $n_L = 1$

Define $\delta_j^{[l]} = \dfrac{\partial a^{[L]}}{\partial z_j^{[l]}}$ for $l = 2, \cdots, L$, $1 \le j \le n_l$

Then $\delta^{[L]} = \dfrac{\partial a^{[L]}}{\partial z^{[L]}} = \sigma'(z^{[L]})$

① $\delta_j^{[l]} = \dfrac{\partial a^{[L]}}{\partial z_j^{[l]}} = \sum_{k=1}^{n_{l+1}} \dfrac{\partial a^{[L]}}{\partial z_k^{[l+1]}} \dfrac{\partial z_k^{[l+1]}}{\partial z_j^{[l]}}$

(i) $\dfrac{\partial a^{[L]}}{\partial z_k^{[l+1]}} = \delta_k^{[l+1]}$

(ii) $z_k^{[l+1]} = \sum_{s=1}^{n_l} W_{ks}^{[l+1]} a_s^{[l]} + b_k^{[l+1]} = \sum_{s=1}^{n_l} W_{ks}^{[l+1]} \sigma(z_s^{[l]}) + b_k^{[l+1]}$

=> $\dfrac{\partial z_k^{[l+1]}}{\partial z_j^{[l]}} = W_{kj}^{[l+1]} \sigma'(z_j^{[l]})$

Thus, $\delta_j^{[l]} = \sum_{k=1}^{n_{l+1}} \delta_k^{[l+1]} \left(W_{kj}^{[l+1]} \sigma'(z_j^{[l]}) \right) = \sigma'(z_j^{[l]}) \sum_{k=1}^{n_{l+1}} \delta_k^{[l+1]} W_{kj}^{[l+1]}$

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

② $\dfrac{\partial a^{[L]}}{\partial x_k} = \sum_{j=1}^{n_2} \dfrac{\partial a^{[L]}}{\partial z_j^{[2]}} \dfrac{\partial z_j^{[2]}}{\partial x_k}$

(i) $\dfrac{\partial a^{[L]}}{\partial z_j^{[2]}} = \delta_j^{[2]}$

(ii) $z_j^{[2]} = \sum_{s=1}^{n_1} W_{js}^{[2]} a_s^{[1]} + b_j^{[2]} = \sum_{s=1}^{n_1} W_{js}^{[2]} x_s + b_j^{[2]}$

$$\Rightarrow \frac{\partial z_j^{[2]}}{\partial x_k} = W_{jk}^{[2]}$$

Thus, $\frac{\partial a^{[L]}}{\partial x_k} = \sum_{j=1}^{n_2} \delta_j^{[2]} W_{jk}^{[2]}$

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

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.

This week we learned about linear regression and locally weighted linear regression (LWLR).

1. I'd like to ask, is linear regression a "parametric learning algorithm," while LWLR is a "non-parametric learning algorithm"?
2. In what data or application scenarios do they each perform better?
3. For example, LWLR seems to have a much higher computational cost. How does this trade-off hold up in practice?