1. We consider a feedforward network

$$a^{[0]} = x,\ z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]},\ a^{[l]} = \sigma^{[l]}(z^{[l]}),\ l = 1, \cdots, L$$

Assume all hidden layers use ReLU activation and the last layer is linear.

Since the output is linear, $g^{[L]} = 1$.

For $l = L-1, L-2, \cdots, 1$:

$$g^{[l]} = (W^{[l+1]})^T g^{[l+1]} \odot 1_{z^{[l]}>0},$$ where $1_{z^{[l]}>0}$ is an indicator function that equals 1 if $z^{[l]} > 0$, and 0 otherwise.

The final result is $\nabla_x a^{[L]}(x) = (W^{[1]})^T g^{[1]}$

2. (I) Practical issue

- What happens if weights are initialized poorly?
- Why do we sometimes add regularization in training, and how does it mathematically change the gradient?

(II). Optimization

- Why is stochastic gradient descent (SGD) usually preferred over full gradient descent?
- How does the learning rate impact convergence, and how should it be chosen in practice?