Regularization, Lipschitz continuity

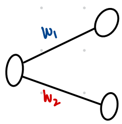
① What is regularization?

: A regularization is any technique that helps your neural network to solve your objective (loss) function. A regularization enforce smoothness for preventing overfitting. Usually, regularizations are added as an extra term to the loss function.

☆ Two very common regularizations (L_1 and L_2: Tikhonov regularization) are already built in most optimizers.

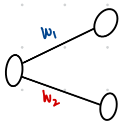
L_1 / L_2 Regularization

L_1 Regularization



Loss $+= \lambda(w_1 + w_2)$

L_2 Regularization



Loss $+= \lambda(w_1^2 + w_2^2)$

- L_1 regularization just adds the absolute sum of all weights of the network to the loss term
- L_2 regularization adds the squared weights of all weights.
  ↳ why? : As we square each weight, this is always punishing the highest weight the most.

② Lipschitz continuity

: Lipschitz continuity is a mathematical concept related to the smoothness or regularity of functions. A function is said to be Lipschitz continuous if $\exists$ a positive constant, often denoted as L, such that the absolute difference in the function values for any two points in its domain is bounded by the product of the Lipschitz constant and the distance between those two points.

Mathematically, $f: \mathbb{R}^n \to \mathbb{R}$ is Lipschitz continuous with $L > 0$ if:

$$|f(x_1) - f(x_2)| \le L \cdot \|x_1 - x_2\| \quad \forall x_1, x_2 \text{ in the domain of } f$$

↳ Upperbound. Lipschitz constant is in fact a natural notion of complexity.

Ex) $f(x) = \cos(wx)$, then $f'(x) = w\sin(wx)$

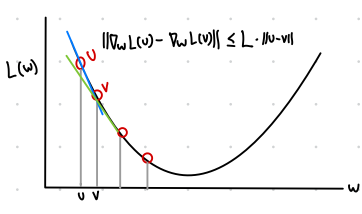
$$|f(x) - f(x')| = \left|\int_x^{x'} f'(t)dt\right| \le \int_x^{x'} |f'(t)|dt \le w \cdot |x - x'|$$

Hence, $f(x)$ is Lipschitz constant since $|f(x) - f(x')| \le w|x - x'|$

↳ Lipschitz constant, and it's a measure of how complex f is.

Then, if the gradient of our loss, $\nabla_w L$, is Lipschitz continuous, i.e.,

$$\|\nabla_w L(u) - \nabla_w L(v)\| \le L\|u - v\|$$



It's a useful property in various context including optimization, numerical analysis, and ML.