Regularization

→ Regularization is the process of reducing overfitting in deep learning models.

Ways to solve overfitting

- Adding more data
  - Add more rows
  - Data augmentation (synthetic data generation)
- Reducing model complexity
  - Dropout
  - early stopping
  - Regularization
    - L1
    - L2
    - L1 + L2

How regularization works?

→ by reducing loss function.

→ by just adding a penalty term we can do regularization.

$$\text{Cost} = \underbrace{\frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i)}_{\text{cost function}} + \underbrace{\frac{\lambda}{2n}\sum_{i=1}^{k} \|w_i\|^2}_{\text{penalty (L2)}}$$

What it does ??

if you have 10 weights w_1 to w_{10} then we are just adding below penalty term in our existing cost function.

$$\frac{\lambda}{2n}\left[w_1^2+w_2^2+w_3^2+w_4^2+w_5^2+w_6^2+w_7^2+w_8^2+w_9^2+w_{10}^2\right]$$

-> Here λ is a hyper parameter.

-> as $\lambda \uparrow$ → panelty $\uparrow$ → regularization applies strongly
overfitting → underfitting

-> if $\lambda = 0 \Rightarrow$ panelty $= 0 =$ No regularization

-> By adding penalty in cost function, all weights moves towards 0 rapidly.

'L1' regularization

$$\text{Cost} = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i) + \frac{\lambda}{2n}\sum_{i=1}^{k} ||w_i||$$

For Artificial Neural Network

$$Cost = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{l=1}^{L}\sum_{i=1}\sum_{j=1} \|W_{ij}^{l}\|^2$$

where
l = number of layers
i = number of columns
j = number of rows

→ In regularization, we doesn't includes bias for penalty calculation.

Intuition behind regularization

→ Suppose we want to update a weight.

$$W_{new} = W_{old} - \alpha \frac{\partial L}{\partial W_{old}} \rightarrow L' = L + \frac{\lambda}{2}\|W_i\|^2$$

$$\frac{\partial L'}{\partial W_{old}} = \frac{\partial L}{\partial W_{old}} + \frac{\lambda}{2} 2W_{old}$$

$$= \frac{\partial L}{\partial W_{old}} + \lambda W_{old}$$

$$W_{new} = W_{old} - \alpha\left(\frac{\partial L}{\partial W_{old}} + \lambda W_{old}\right)$$

$$W_{new} = W_{old} - \alpha\lambda W_{old} - \alpha\frac{\partial L}{\partial W_{old}}$$

$$W_{new} = (1 - \alpha\lambda)W_{old} - \alpha\frac{\partial L}{\partial W_{old}}$$

L2 reg. = Weight decay

→ In L1 reg. many weights may become 0.