#Regularizations

- **L2 Regularization (Weight Decay)**

- original loss: $L = \text{error}(y, \hat{y})$
- L2 adds a penalty for large weights: $L = \text{error}(y, \hat{y}) + \lambda \|w\|^2$
- where:
  - w = model weights
  - λ (lambda) = regularization strength
- gradient update becomes:

$$w \leftarrow w - \eta\left(\frac{\partial L}{\partial w} + 2\lambda w\right)$$

- effect:
  - pushes weights toward zero
  - Prevents any weight from becoming too large.

- **Dropout**

- during training, each neuron is kept with probability p
- dropped neurons output 0
- mathematically:

$$\boxed{\tilde{h} = h \odot m/p}$$

- where:
  - h = neuron output
  - m = random mask (0 or 1)
  - $\odot$ = element-wise multiplication
- at test time:
  - no dropout
  - full network is used

- L2 controls model complexity by shrinking weights
- Dropout prevents co-adaptation by randomly removing neurons.