# L2 Regularization

03 January 2025 18:58

optimization

L → mse
L → logloss → weights/biases

Applied to Model Weights:

- Regularization is applied to the weights of the model to penalize large values and encourage smaller, more generalizable weights.

Introduced via Loss Function or Optimizer:

- Adds a penalty term λ∑wi2 to the loss function in L2 regularization.

$$\text{Loss}_{\text{reg}} = \text{Loss}_{\text{original}} + \lambda \sum w_i^2$$

$\lambda \left( w_1^2 + w_2^2 + w_3^2 + w_4^2 \right)$

- In weight decay, directly modifies the gradient update rule to include λwi, effectively shrinking weights during training.

$$\boxed{w \leftarrow w - \eta(\nabla \text{Loss} + \lambda w)}$$

weight_decay

Penalizes Large Weights:

- Encourages the network to distribute learning across multiple parameters, avoiding reliance on a few large weights.

Reduces Overfitting:

- Helps the model generalize better to unseen data by discouraging overly complex representations.

Controlled by a Hyperparameter:

- A regularization coefficient (λ) often set via weight_decay in optimizers) controls the strength of the penalty. Larger values lead to stronger regularization.

No Effect on Bias Terms:

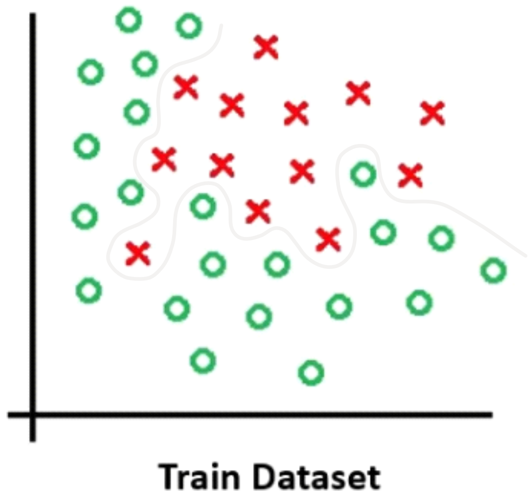
- Regularization is typically applied only to weights, not biases, as biases don't directly affect model complexity.

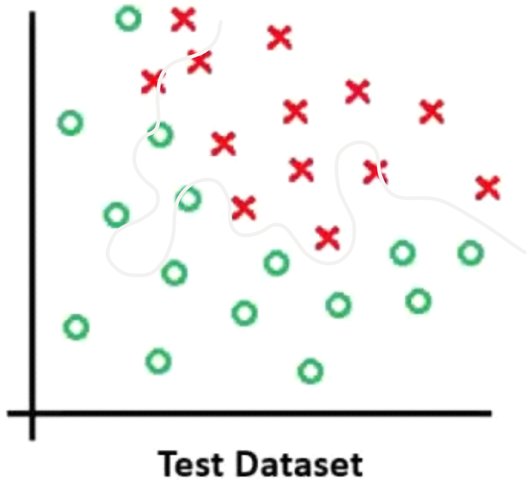
Active During Training:

- Regularization affects weight updates only during training. It does not explicitly influence the model during inference.

# Overfitting

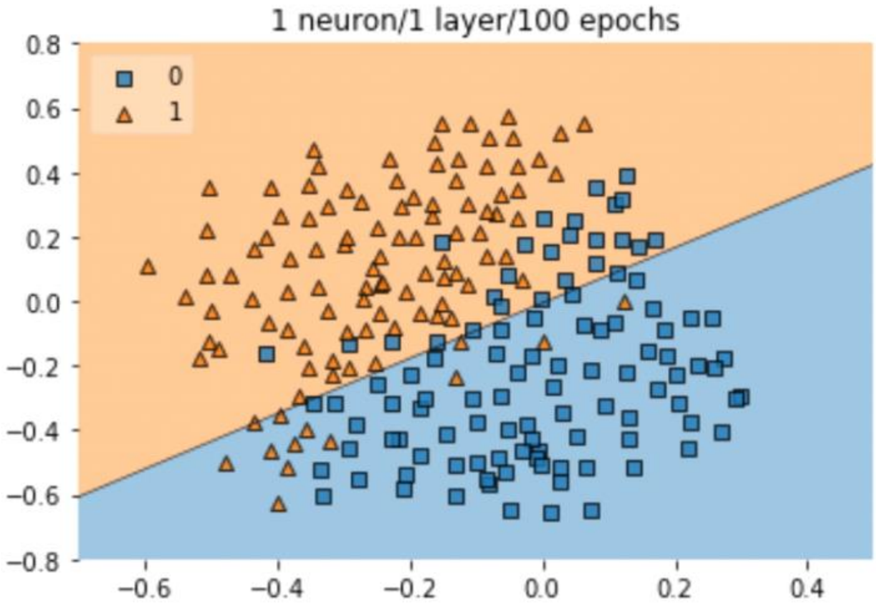
20 May 2022 08:14

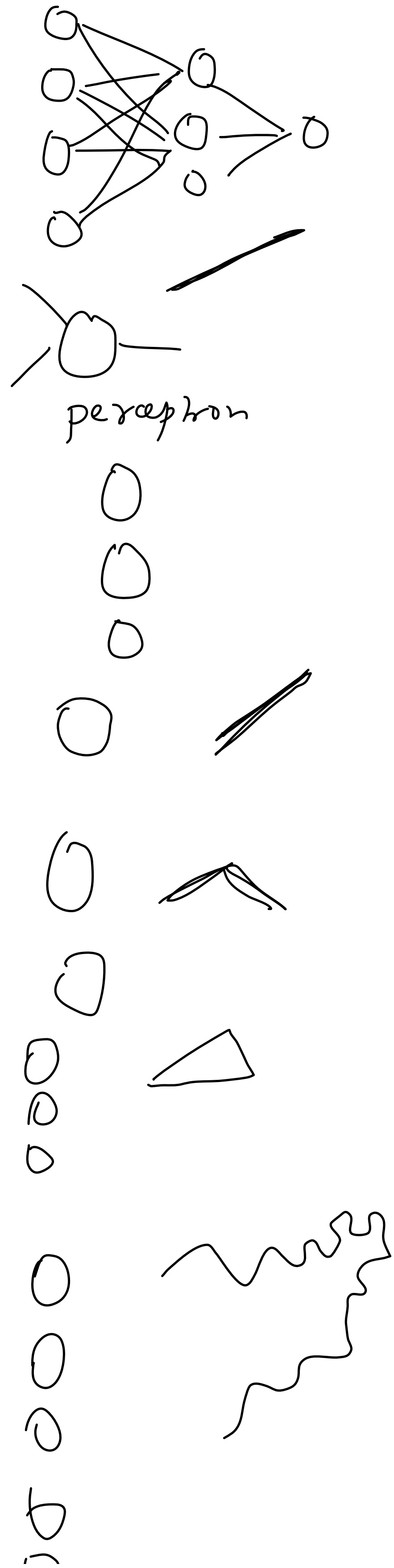


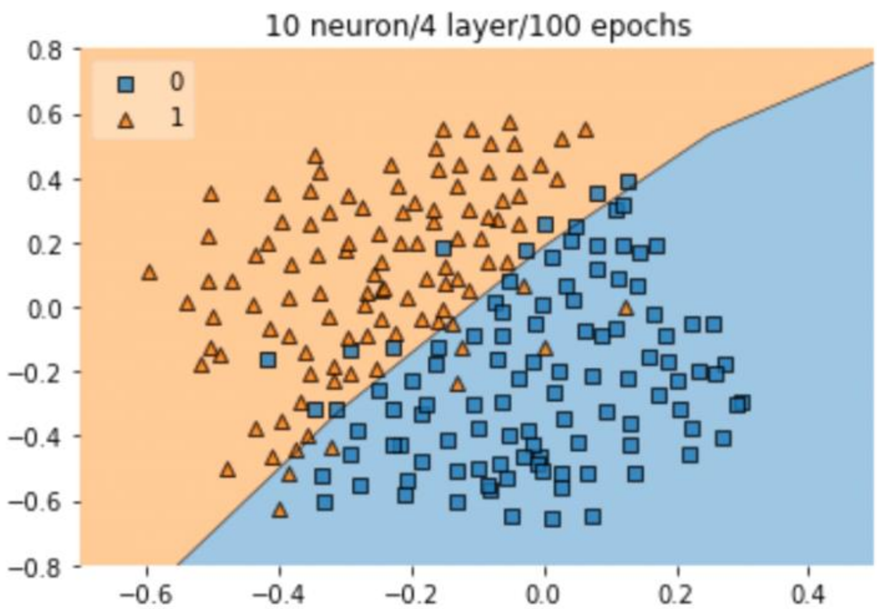



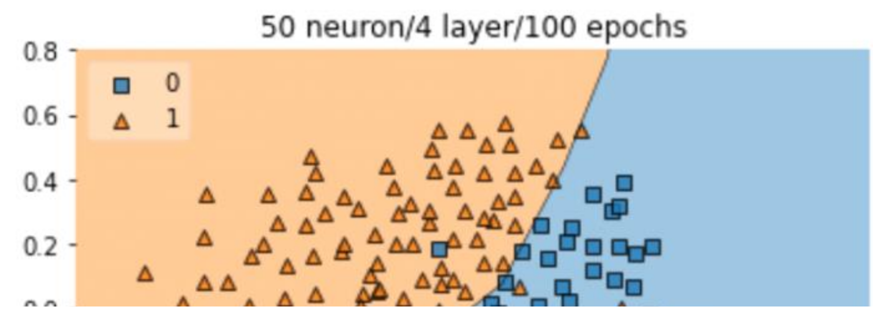
# Why Neural Networks Overfit?

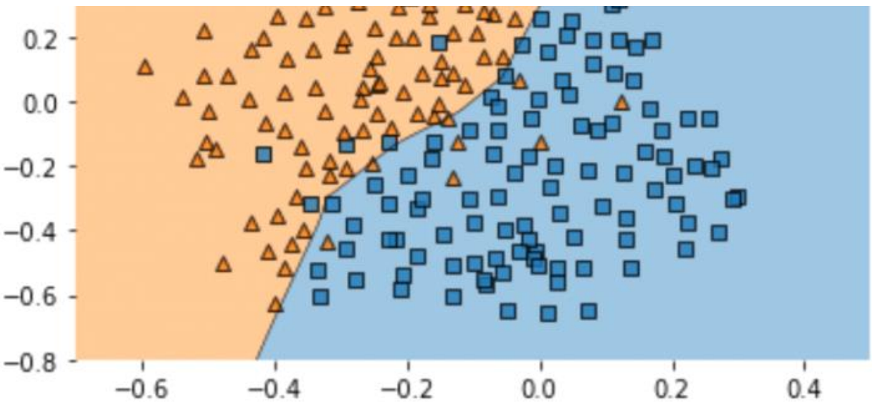
19 May 2022 07:34

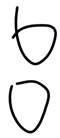


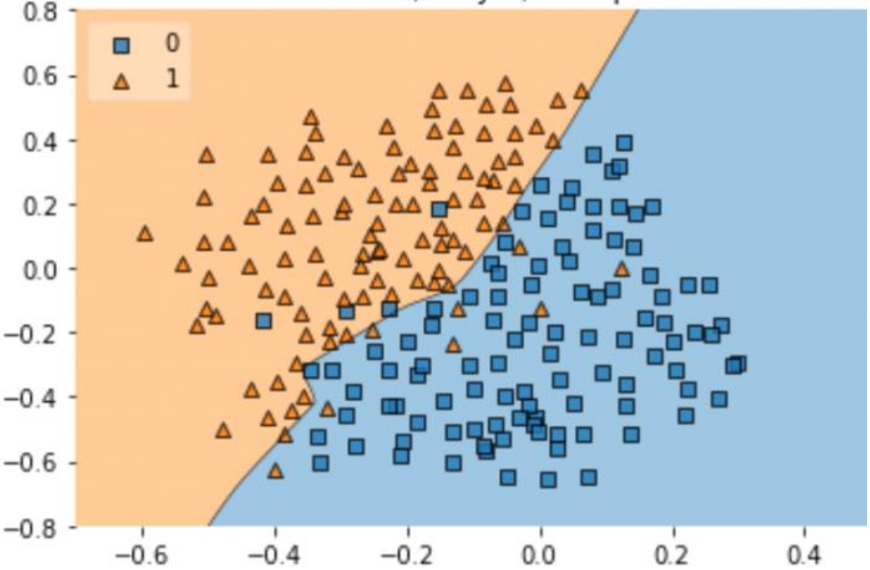



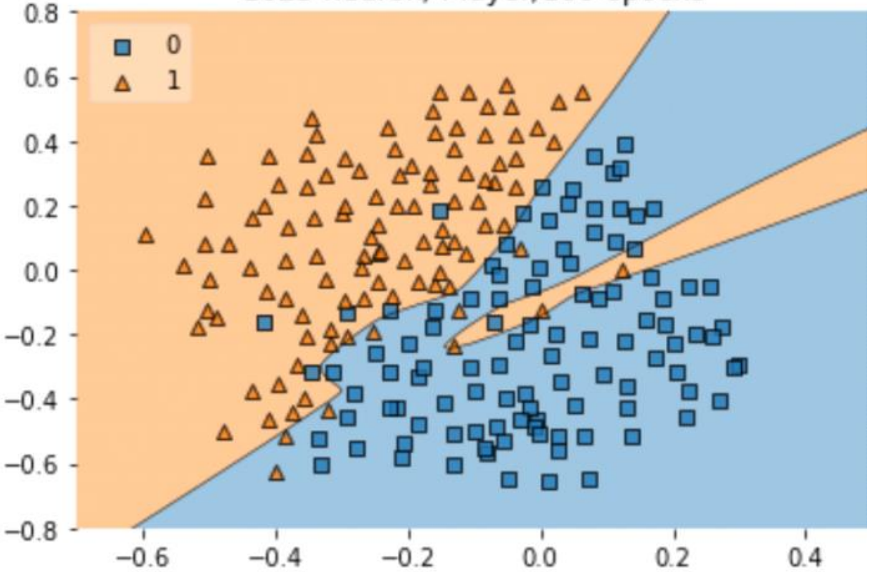


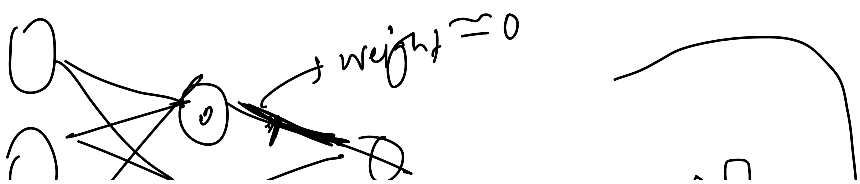



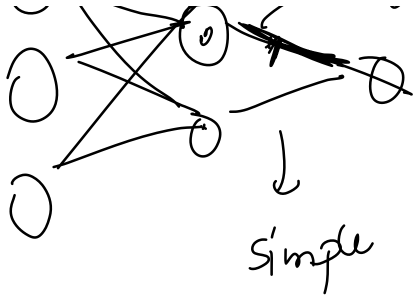
257 neuron/4 layer/100 epochs

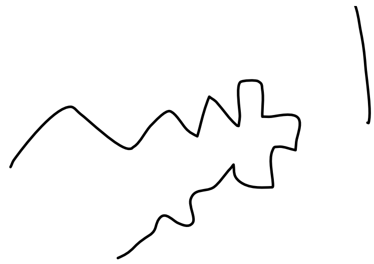



1025 neuron/4 layer/100 epochs




simple



# Ways to solve overfitting

20 May 2022 08:13

**Adding more data**

- → Add more rows
- → Data augmentation

**Reducing the complexity model**

- → Dropout
- → Early stopping
- → Regularization
  - → L1
  - → L2 → ✓
  - → L1 + L2

# Regularization

20 May 2022 08:14

ANN → weights | bias

↳ min Lossfunction

$L =$ mse

↳ binary

$$C = \boxed{\frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i)} + \text{penalty term}$$

$\boxed{L2}$

L1

$\boxed{w_1 \to w_{10}}$

$$C = L + \boxed{\frac{\lambda}{2n}\sum_{i=1}^{k} \|w_i\|^2}$$

weightage

$$\frac{\lambda}{2n}\left[w_1^2 + w_2^2 + \ldots + w_{10}^2\right]$$

↑ $\lambda$ = hyperparameter ↑

$\boxed{\lambda = 0}$

$$C = \sum L(y_i, \hat{y}_i) + P$$

↓

$w \simeq 0$

L2 → L1

→ L1 norm

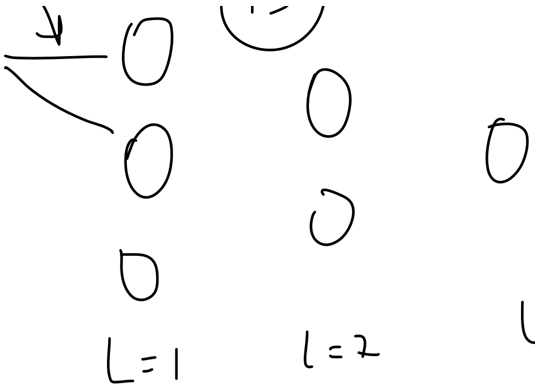
$$C = L + \frac{\lambda}{2n}\sum \|w_i\|$$

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

$\boxed{w \simeq 0}$

$l, j = d$

0 15




$w_1^2 + w_2^2 + w$

$\sum_{i=1}^{K} \|b_i\|^2$

$\boxed{\text{bias X}}$

# Intuition behind Regularization

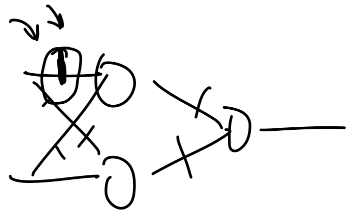
20 May 2022 08:15

$$\boxed{w_n = w_0 - \eta \frac{\partial L}{\partial w_0}}$$

$\boxed{1-\eta\lambda}$

positive

$L$



$$L' = L + \frac{\lambda}{2} \sum \|w_i\|^2$$

$w_2^1 + \boxed{w_2^2 + w_3 + \dots}$

$2w$

$$\frac{\partial L'}{\partial w_0} = \frac{\partial L}{\partial w_0} + \frac{\lambda}{\cancel{2}} \cancel{2} w_0$$

$$= \frac{\partial L}{\partial w_0} + \lambda w_0$$

$w\downarrow$

$$w_n = w_0 - \eta\left(\frac{\partial L}{\partial w_0} + \lambda w_0\right)$$

$$w_n = w_0 - \eta\lambda w_0 - \eta \frac{\partial L}{\partial w_0}$$

$$\boxed{w_n = (1-\eta\lambda) w_0 - \eta \frac{\partial L}{\partial w_0}}$$

$w_0 << w_0$

L2 reg → weight decay

weight decay ↑ $w_0$