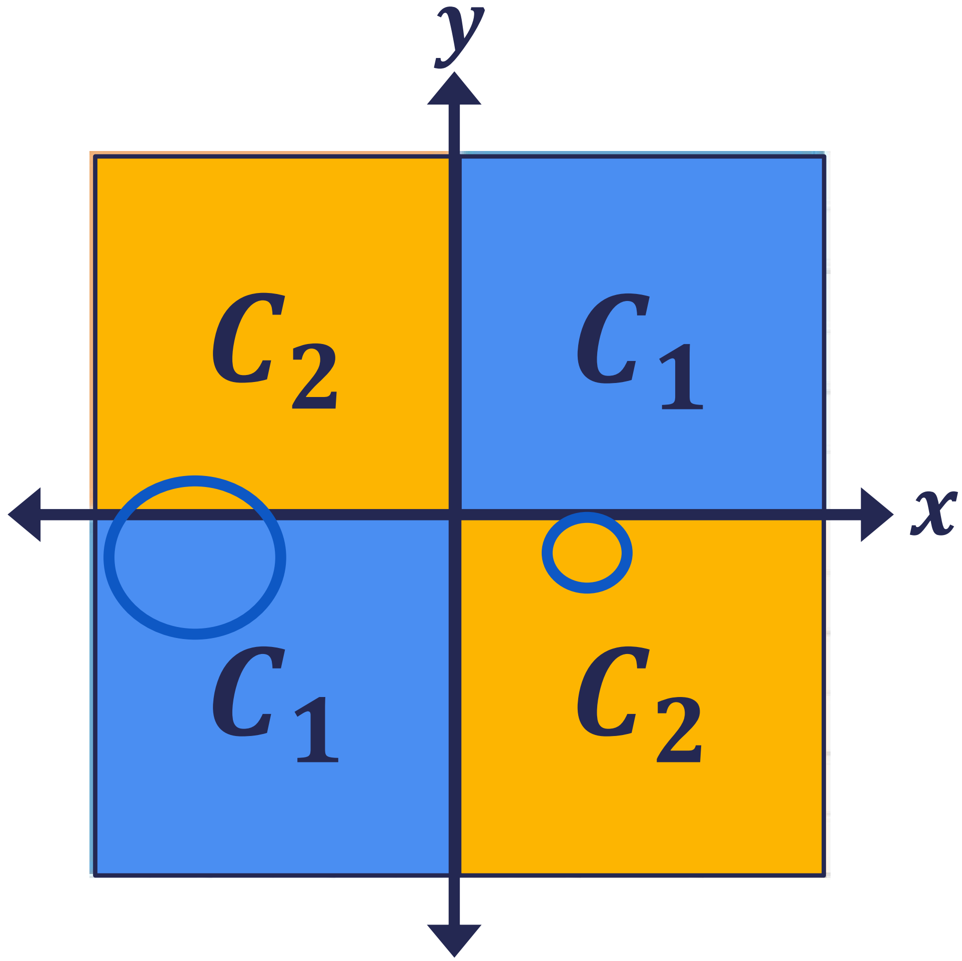
Learning Objectives

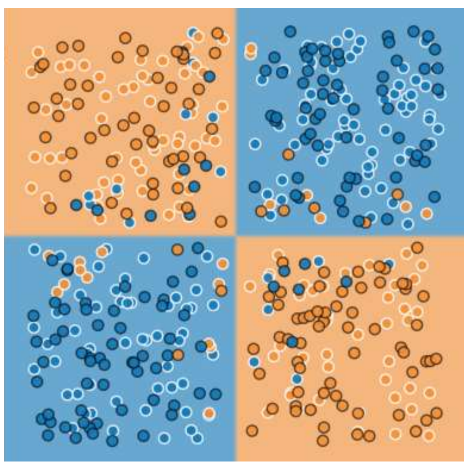
- Learn to remedy overfitting through various regularization strategies including:
  - Parameter norm penalties
  - Dropout
  - Early Stopping

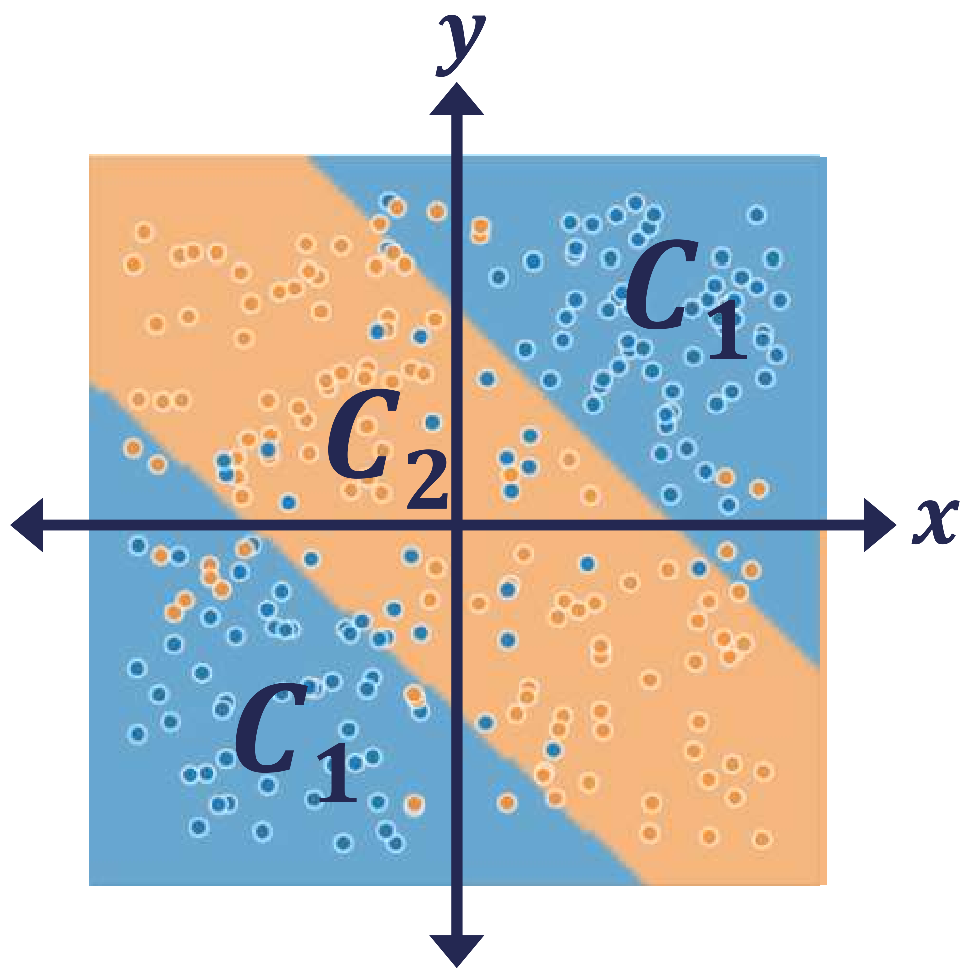
Toy Example



Toy Example

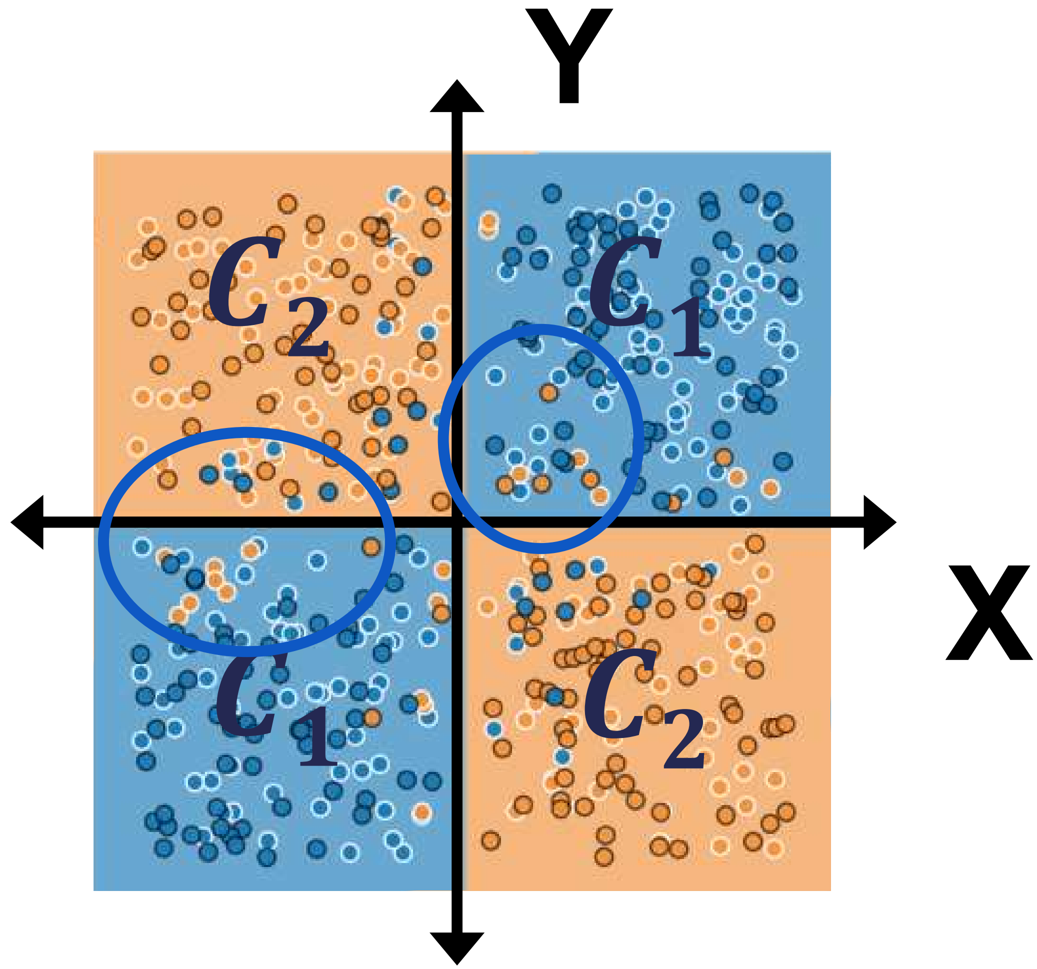
- **Initial Design:**
  - 1 Layer NN, 2 Hidden Units/Layer
  - **Train set Loss**: 0.264
  - **Val set Loss**: 0.268
  - **Minimum Loss achievable:** 0.1





Toy Example

- **New Design:**
  - 6 Layer NN, 6 Hidden Units/Layer
  - **Train set Loss**: 0.1
  - **Val set Loss**: 0.45
  - **Minimum Loss achievable:** 0.1



Parameter Norm Penalties

$$J(\theta)_{reg} = J(\theta) + \alpha\Omega(\theta)$$

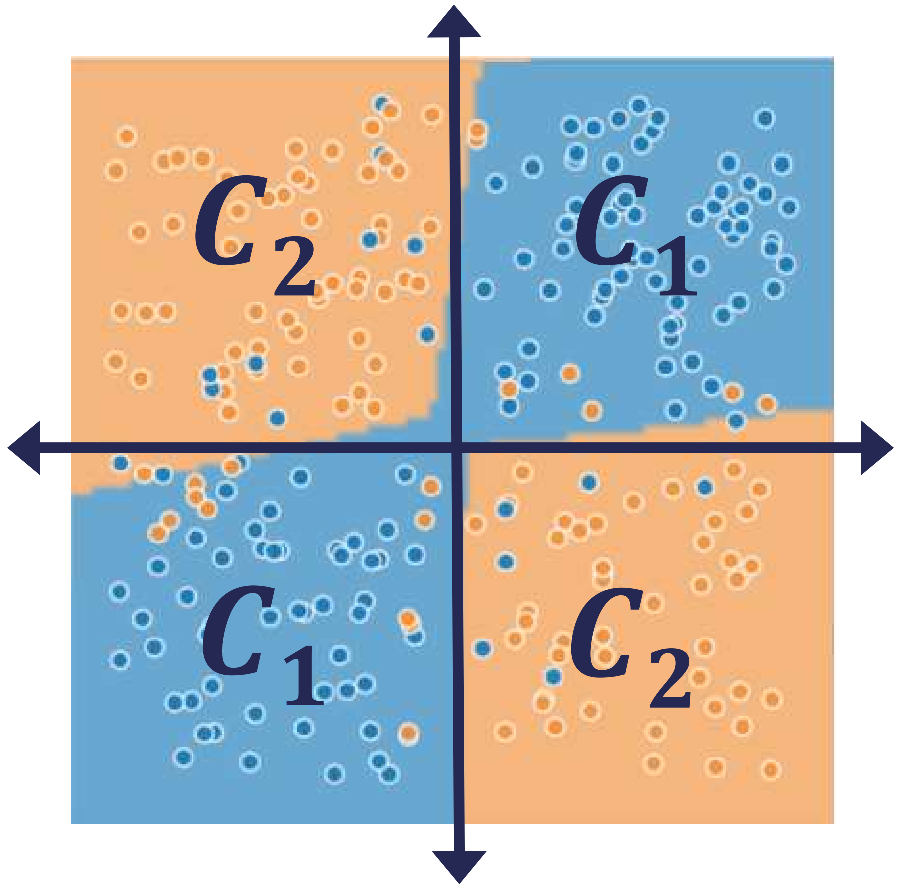
- α is a **hyperparameter** that weights the relative contribution of the norm penalty to the value of the loss function
- $\Omega(\boldsymbol{\theta})$ is a measure of how large $\boldsymbol{\theta}$'s value is, usually an **Lp Norm**
- We usually only constrain the size of **weights** and not biases

$$J(\theta)_{reg} = J(\theta) + \alpha\Omega(W)$$

L2-Norm Parameter Penalty

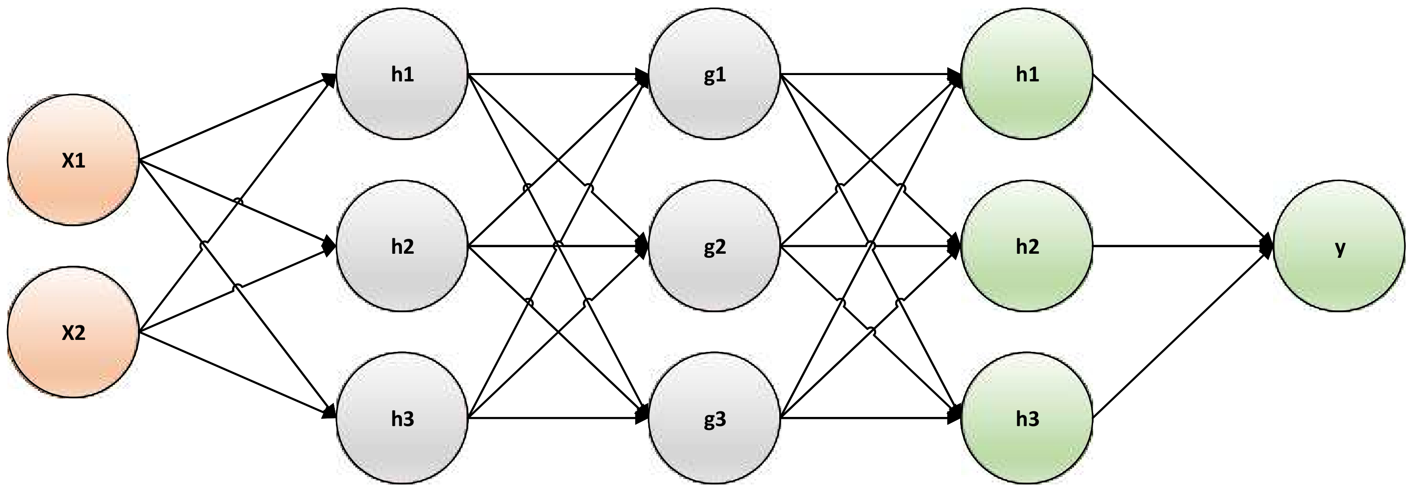
$$\Omega(W) = \frac{1}{2}W^T W = \frac{1}{2}\|W\|_2^2$$

- **New Design:**
  - 6 Layer NN, 6 Hidden Units/Layer
  - **Minimum Loss achievable:** 0.1
  - L2-Norm Penalty
  - **Train set Loss**: 0.1 0.176
  - **Val set Loss**: 0.45 0.182



Dropout

Multiply **Weights** by P_{keep} at the end of training



$$P_{keep} = 0.5$$

Dropout

- Computationally inexpensive but powerful regularization method
- Does not significantly limit the type of model or training procedure that can be used
- Dropout layers are practically implemented in all neural network libraries !

Early Stopping