

# Regularização L1 e L2

# Lasso (L1)

```
1 l1_lambda = 0.001
2 loss_fn = nn.MSELoss()
3
4 for data, target in dataloader:
5     optimizer.zero_grad()
6     output = model(data)
7     loss = loss_fn(output, target)
8
9     # Termo L1 manual: soma dos valores absolutos dos pesos
10    l1_norm = sum(p.abs().sum() for p in model.parameters())
11    loss = loss + l1_lambda * l1_norm
12
13    loss.backward()
14    optimizer.step()
```

$$J(\theta) = \text{Loss}(\theta) + \lambda \sum_i |w_i|$$

- Penaliza a soma dos valores absolutos dos pesos.
- Efeito: força muitos pesos a se tornarem exatamente zero → gera esparsidade.

# Ridge (L2)

```
1 import torch
2 import torch.nn as nn
3 import torch.optim as optim
4
5 model = nn.Linear(10, 1) # exemplo simples
6
7 # Otimizador com regularização L2 (weight_decay = λ)
8 optimizer = optim.SGD(model.parameters(), lr=0.01, weight_decay=0.001)
```

$$J(\theta) = \text{Loss}(\theta) + \lambda \sum_i w_i^2$$

- Penaliza a soma dos quadrados dos pesos.
- Efeito: reduz todos os pesos, mas raramente zera → promove suavidade.