




正則化

複雑なモデルは多くのパラメータを持ち、柔軟に調整することができる
しかし、モデルを柔軟にしすぎると過学習を起こす可能性がある
そこで、パラメータに制限（条件）をつけることで過学習を防ぐ手法として、**正則化（regularization）**がある

ref: なっとく！機械学習 p93～97、p99

 **正則化** モデルの複雑さにペナルティを科す（モデルに格納できる情報の量を調整するか、モデルに格納できる情報の種類を制限する）ことで、最終的に過学習を防ぐ手法

複数のモデルをテストし、性能と複雑さのバランスが最もよいものを選ぶことでモデルの最適化を図る方法もあるが、正則化を用いる場合は、モデルを何種類も訓練する必要はない

正則化では、モデルを訓練するのは 1 回だけだが、モデルを訓練しながら次の 2 つの最適化も試みる

- 性能を向上させる
- 複雑さを減らす

そのために、性能の指標と複雑度の指標を数値化し、それらを組み合わせた上で最適化問題を解くという方法をとる

- **回帰誤差（regression error）**：モデルの性能（品質）の指標
- **正規化項（regularization term）**：モデルの複雑度の指標

回帰誤差として使われるのは、絶対誤差や二乗誤差などである

正規化項として使われるのは、モデルの **L1 ノルム**や **L2 ノルム**である

これらを用いて、性能がよくあまり複雑ではないモデルを見つけ出すため

に、次のように変更された誤差関数を最小化する

$$\text{誤差} = \text{回帰誤差} + \text{正則化項}$$



正則化パラメータ

正則化を用いても、モデルの性能を向上させようとするすると複雑さが増し、モデルを単純化しようとするすると性能が低下するという綱引き状態になることがある

その場合は、ハイパーパラメータを使って、性能と複雑度の間で調整を行う

ここで使われるハイパーパラメータは、**正則化パラメータ**と呼ばれるもので、その目的はモデルの訓練プロセスにおいて、性能と単純さのどちらを重視すべきかを決めることにある

正則化項に正則化パラメータ λ をかけたものに、回帰誤差を足し、その結果を使ってモデルを訓練する

$$\text{誤差} = \text{回帰誤差} + \lambda \cdot \text{正則化項}$$

λ の値を大きくすると（おそらく次数の小さい）単純なモデルになり、データセットにあまりうまく適合しないことがある

そのため、**検証データセット**を使って、モデルの性能が最も良くなる λ の値を選択することが重要となる

ref: なっとく！機械学習 p100