

# 4.4 過学習と学習不足

17T4015S

内間けんじ

## 4.4.0 過学習と学習不足

機械学習を用いてモデルの**最適化**と**汎化**を行いたい。

**最適化**：訓練データでの性能を高める

**汎化**：新しいデータに対する性能を高める

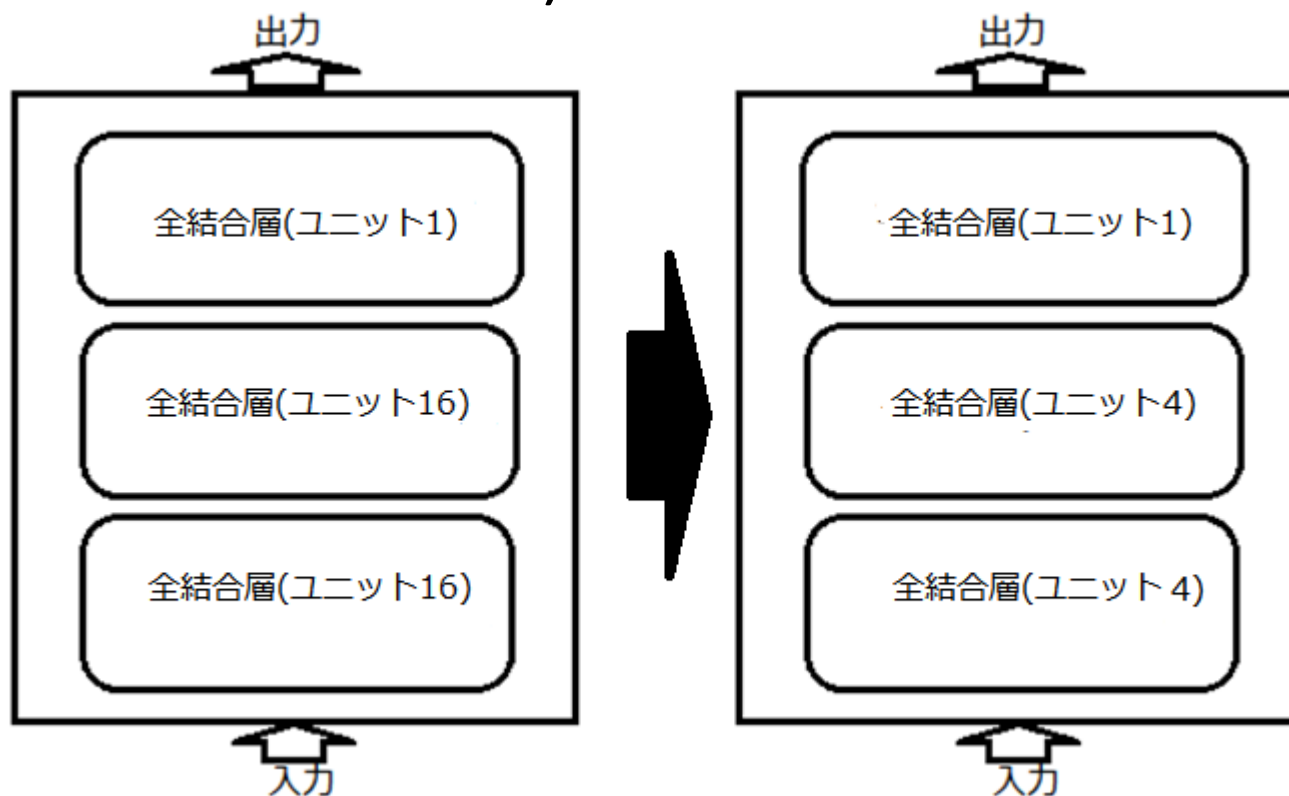
訓練中に、訓練データでの損失値とテストデータの損失値をまだ小さくできる状態を**学習不足**という。

しかし、訓練で汎化性能がそれ以上改善されなかった状態を**過学習**といい、訓練データに特化したパターンを学習するようになる。

→モデルを正しく学習させ過学習を克服したい = **正則化**

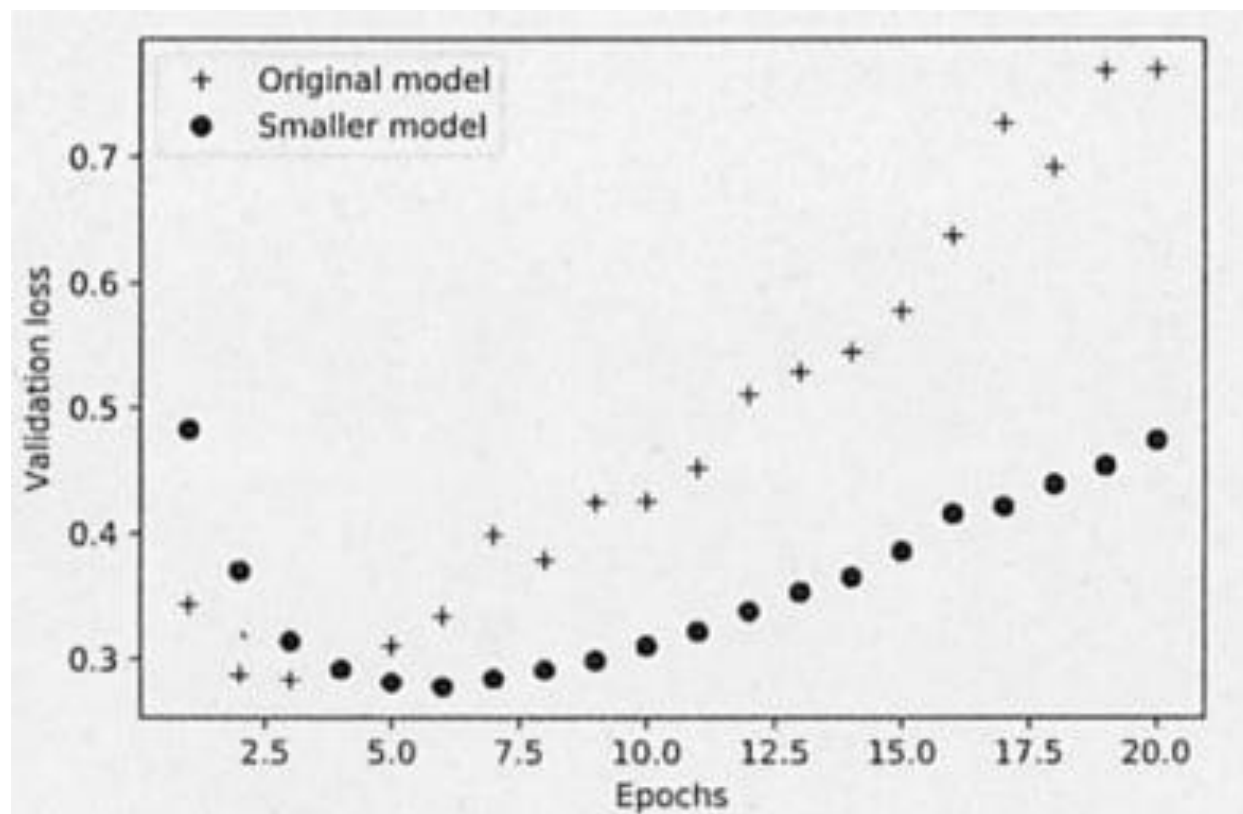
## 4.4.1 ネットワークのサイズを削減

モデルのサイズを小さくする(モデルの学習可能パラメータを小さくする)ことで過学習を回避できる。



2クラス分類のモデルの変更

## 4.4.1 ネットワークのサイズを削減



ネットワークのサイズを削減した結果

## 4.4.2 重みの正則化

**オッカムの剃刀**：何かに対する説明が複数ある場合、最も正しい説明は最も単純なものである。

→ある訓練データに対するモデルが複数ある場合、単純なモデルであるほど複雑なものより過学習を起こしにくい。

**単純なモデル**：パラメータの値に関するエントロピーが小さいモデル

→ネットワークの重みに小さい値だけが設定されるようにすることでネットワークの複雑さに歯止めをかける**重みの正則化**を行う

## 4.4.2 重みの正則化

重みを正則化するために、大きな重みを使用する場合のコストを損失関数に追加する

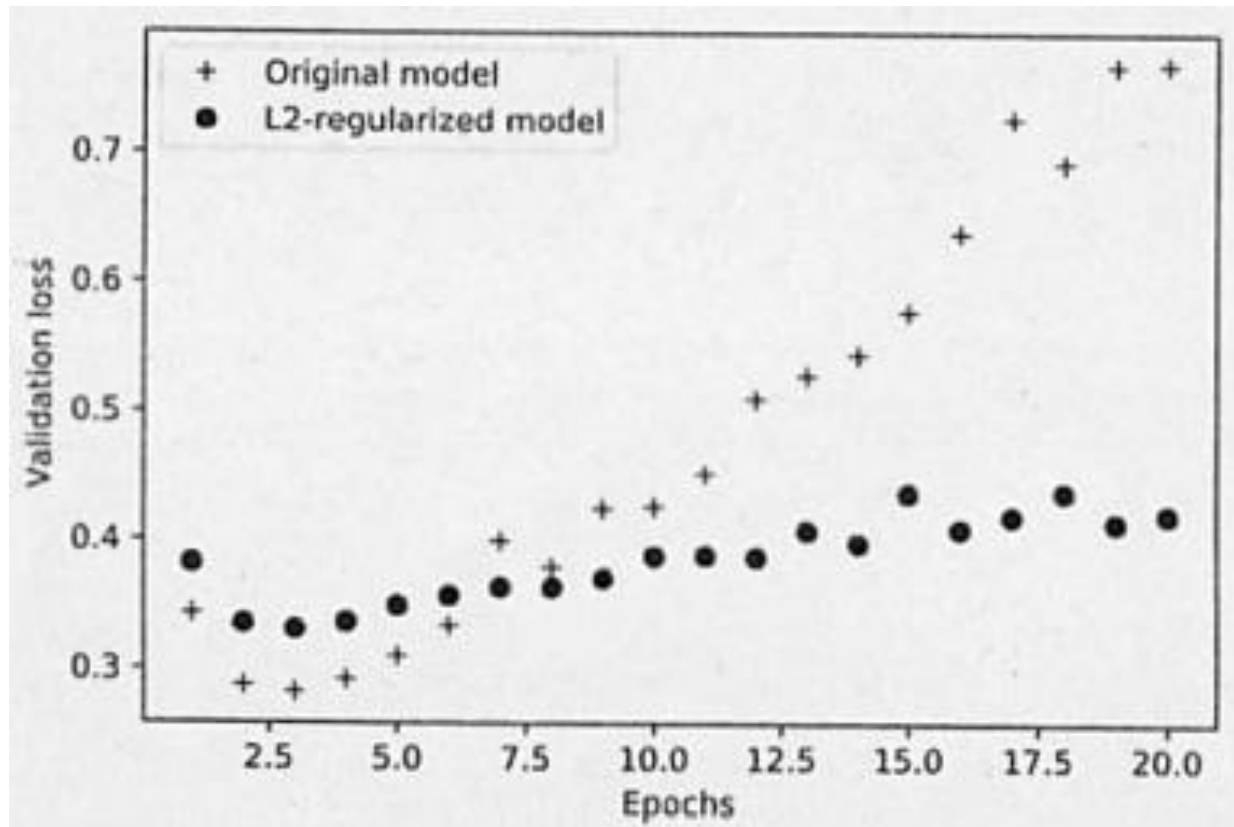
### **L1正則化**

追加されるコストは重み係数の絶対値に比例

### **L2正則化**

追加されるコストは重み係数の値の二乗に比例

## 4.4.2 重みの正則化



訓練時にネットワークの全損失に  
 $0.001 * \text{weight\_coefficient\_value}$ を加えた結果

## 4.4.3 ドロップアウトの追加

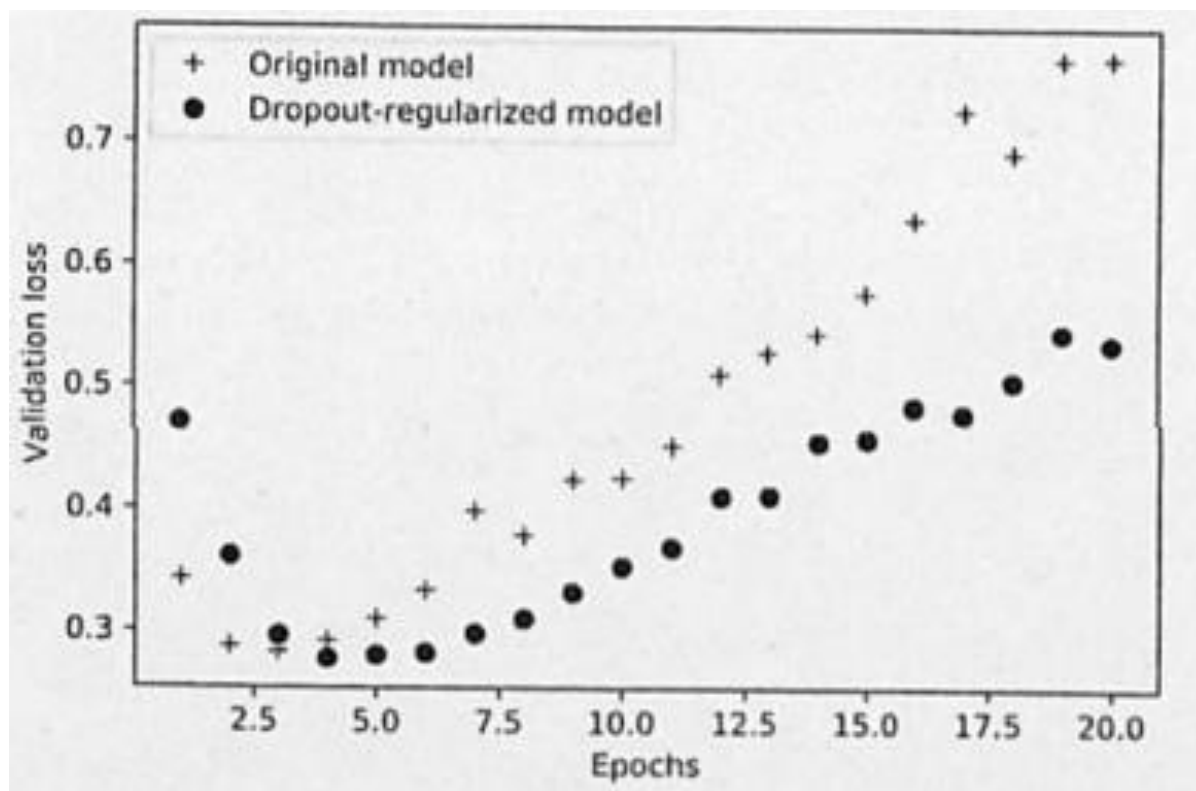
**ドロップアウト**：訓練中にある層の出力の特徴量の一部を**ランダム**に0に置きかえる

例[0.2, 0.5, 1.3, 0.8, 1.1]→ [ 0, 0.5, 1.3, 0, 1.1]

通常テスト時には訓練時よりも多くのユニットが活性化することを踏まえ、テスト時にはドロップアウト率に基づいで出力をスケールダウンする。

(訓練時にドロップアウトとともにスケールアップをすることでテスト時のスケールダウンを省くことができる。)

## 4.4.3 ドロップアウトの追加



ドロップアウトを適用した結果

# まとめ

## ー過学習を防ぐための方法ー

- ・ 訓練データを増やす
- ・ ネットワークのキャパシティを減らす
- ・ 重みを正則化する
- ・ ドロップアウトを追加する

\* 古典的なNNに対する過学習の対応であり、DLにはまた別の対策もある