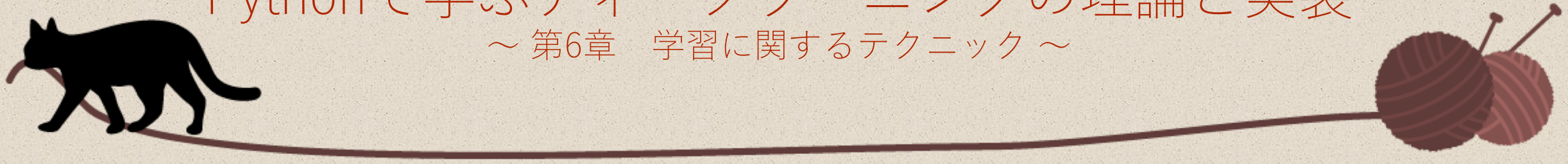
ゼロから作るDeep Learning
Pythonで学ぶディープラーニングの理論と実装
～ 第6章　学習に関するテクニック ～



浅井香奈江
(お茶の水女子大学)

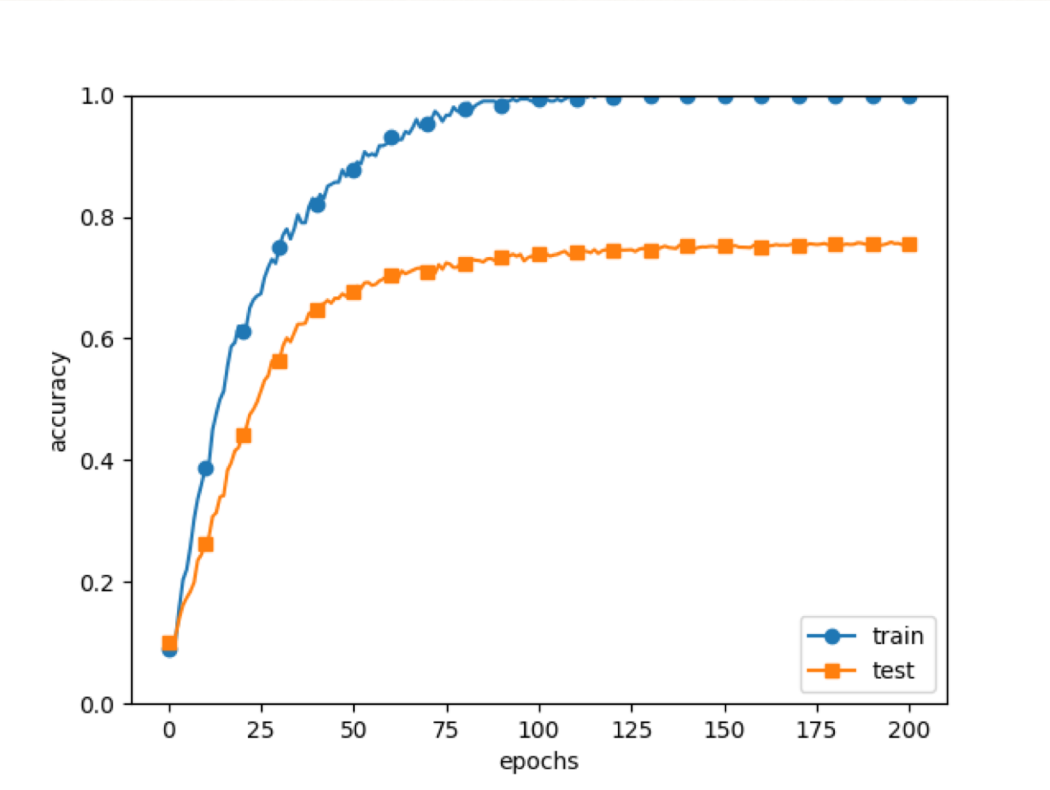
内容

1節　パラメータの更新
2節　重みの初期値
3節　Batch Normalization
4節　正則化
5節　ハイパーパラメータの検証

} 今日の話

過学習

過学習：訓練データだけに適応しすぎてしまい、訓練データに含まれない他のデータに対応できない状態
過学習が起きる原因：
- パラメータを大量に持ち、表現力の高いモデルである事（あまりに厳密な予測モデルになってしまう）
- 訓練データが少ない事（幅広いパターンに対応できない）



過学習を再現するために、訓練データを削減

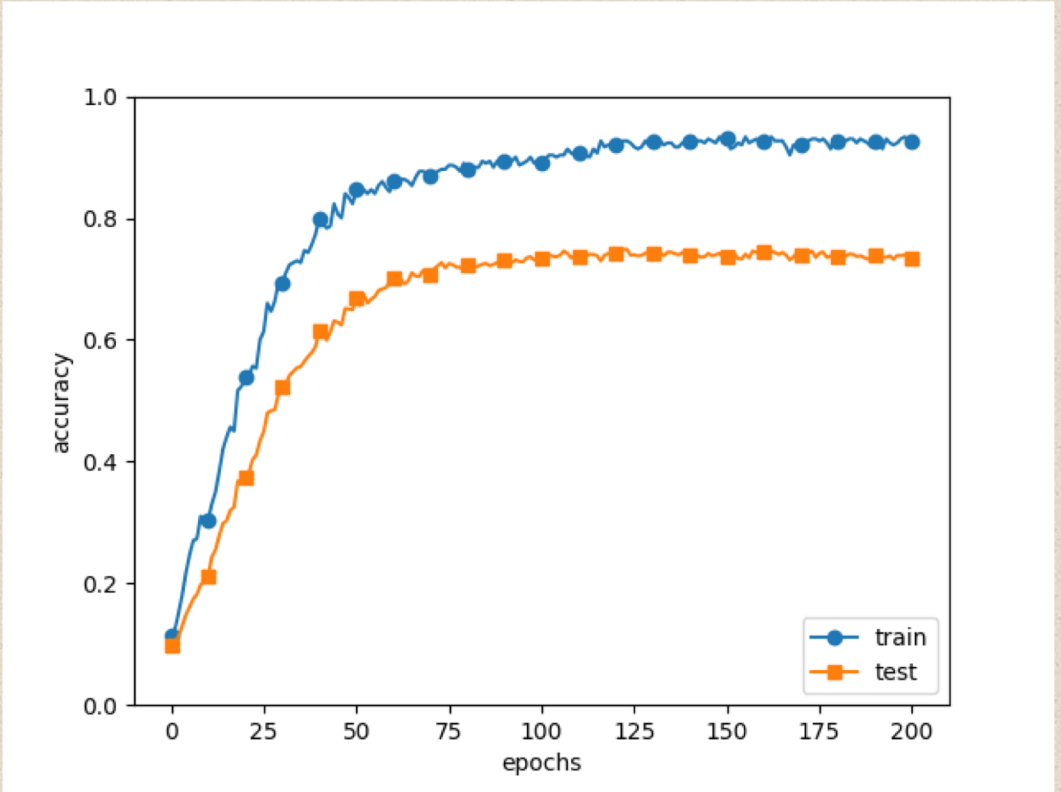
- 訓練データを用いて計測した認識精度
  100エポックを過ぎたあたりから、ほぼ100%

- テストデータを用いて計測した認識精度
  100%の認識精度からは大きな隔たり

→ 訓練データだけに適応しすぎてしまった結果

Weight Decay (荷重減衰)

ディープラーニングではレイヤーが多層になるほど
・モデルの表現能力が増す
・過学習のリスクも高くなる
→ モデルの表現能力を維持したまま、パラメータの自由度に制限を与えることで過学習のリスクを減らす
その手法の一つがweight decay (荷重減衰)



過学習は重みが大きな値をもつことによって発生することが多いため、学習過程で重みが大きくならないようにペナルティを課す

L2正規化：

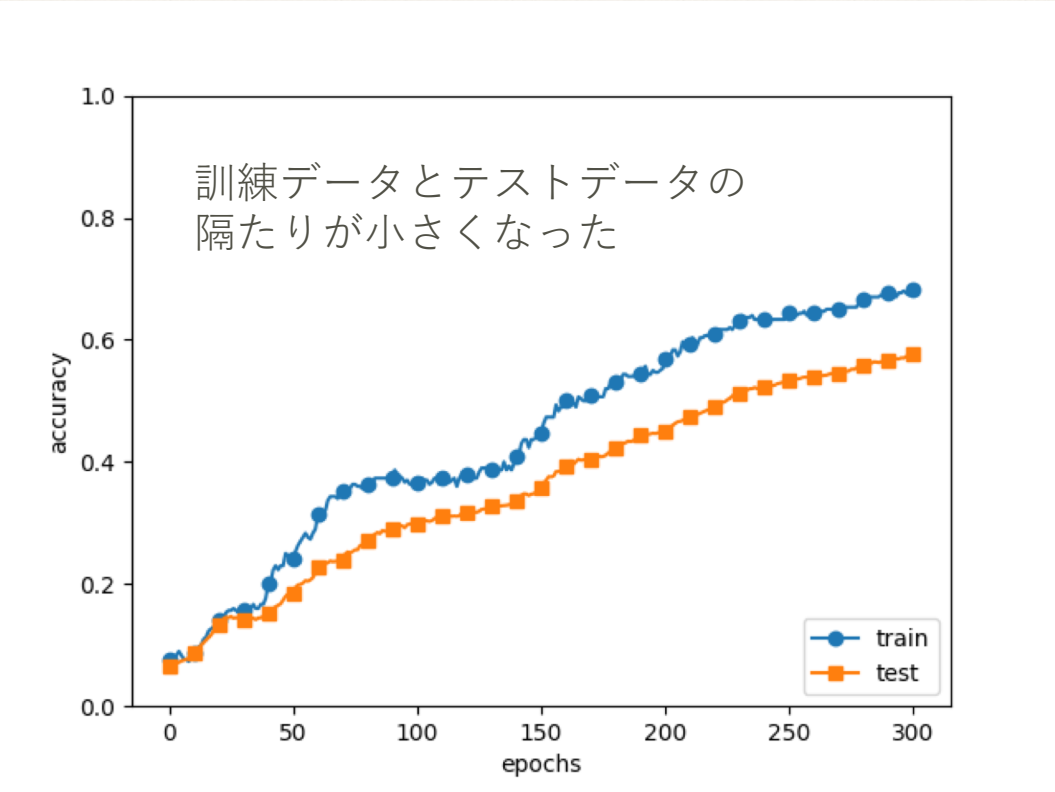
$$w \leftarrow w - \eta\left(\frac{\partial C(w)}{\partial w} - \lambda w\right), \tilde{C}(w) = C(w) + \frac{\lambda}{2}\|w\|^2$$

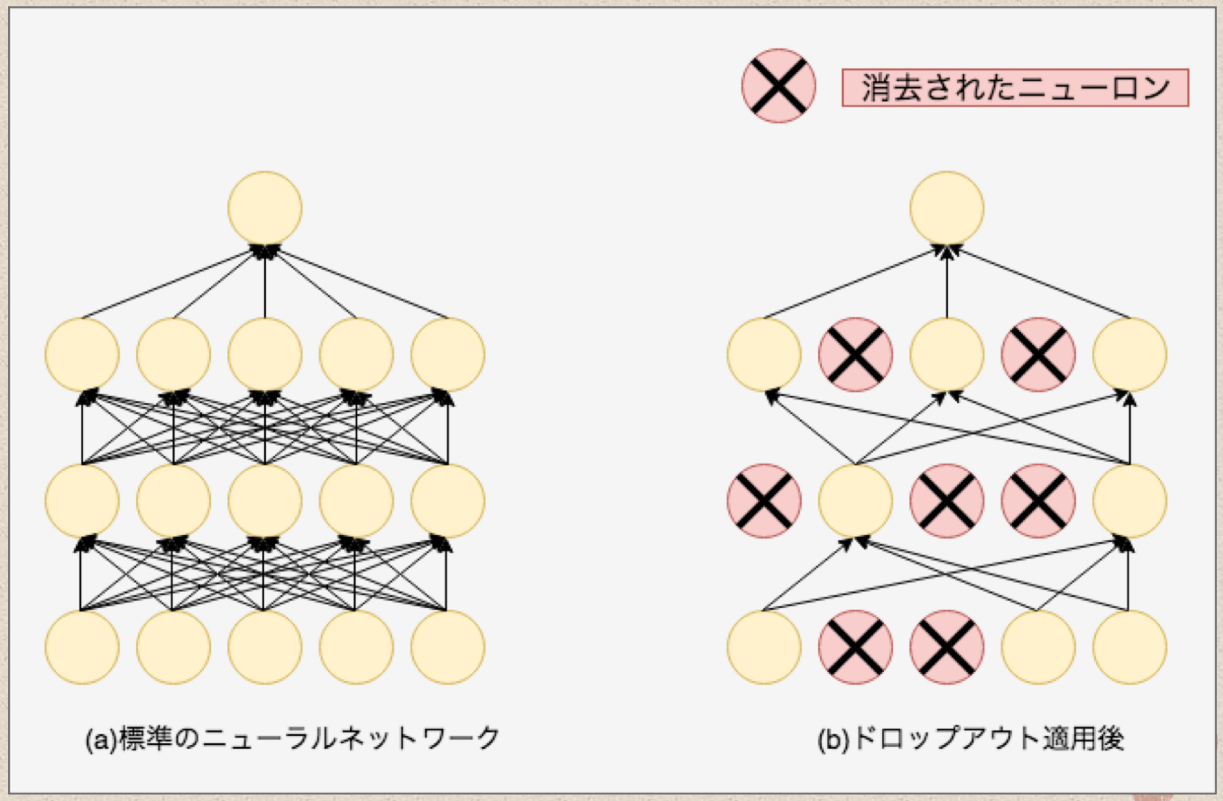
・weight decayなし(前のページ)と比較すると、隔たりが小さい
→ 過学習が抑制されてテストデータの精度が高まった

・訓練データの認識精度が100%に到達していない
→ 過剰な表現力が和らいだ

Dropout

ランダムにニューロンを選び、そのニューロンを消去することで、その先の信号の伝達をストップするしながら学習する手法 (過学習を抑制)
→ コード上では生成した乱数がある閾値以下であればfalseを返し、信号をmaskしている





ハイパーパラメータの検証

ハイパーパラメータの例：
- 各層のニューロンの数
- バッチサイズ
- 学習係数
- weight decay

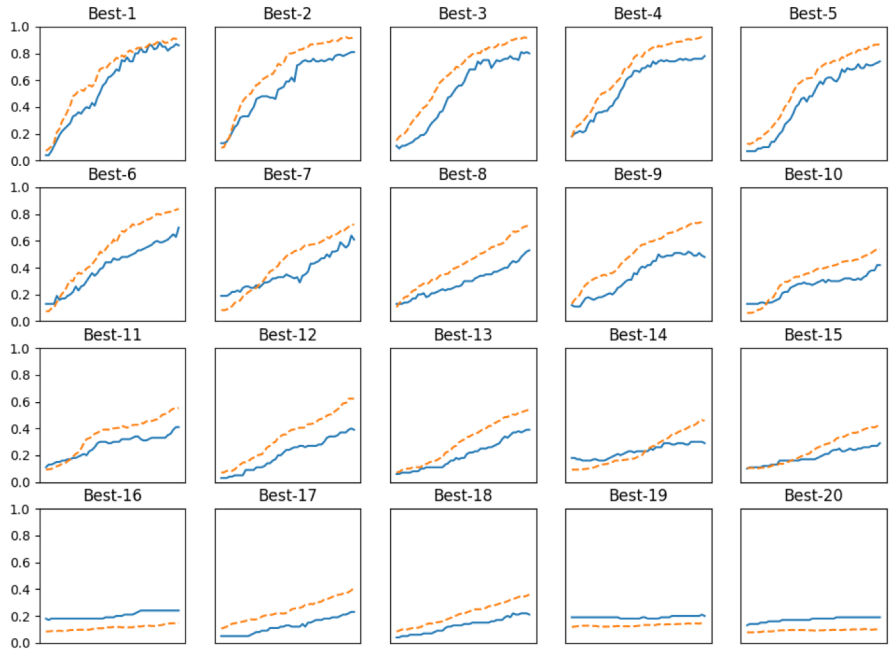
→ 最適なハイパーパラメータを決定するために、検証データを用いてハイパーパラメータの良さを評価

検証データ：
- ハイパーパラメータはテストデータで性能を評価してはいけない
  → 過学習を起こす原因になるため
- ハイパーパラメータ専用の確認データ(検証データ)を用意
  → データの内容によってはユーザの手で作る必要がある
  (コード上では、シャッフルした訓練データから20%程度を検証データとして先に分離して使用している)

各データの使用用途：
- 訓練データ：パラメータ(重みやバイアス)の学習
- 検証データ：ハイパーパラメータの性能評価
- テストデータ：ニューラルネットワークの汎化性能 (過学習をしていないかの) 評価
  → 理想的には一度だけ

ハイパーパラメータの最適化

実線は検証データの認識精度、点線は訓練データの認識精度



うまく学習が進んでいるのは、学習係数が 0.001～0.01、weight decay係数が10^{-8}～10^{-7}あたり

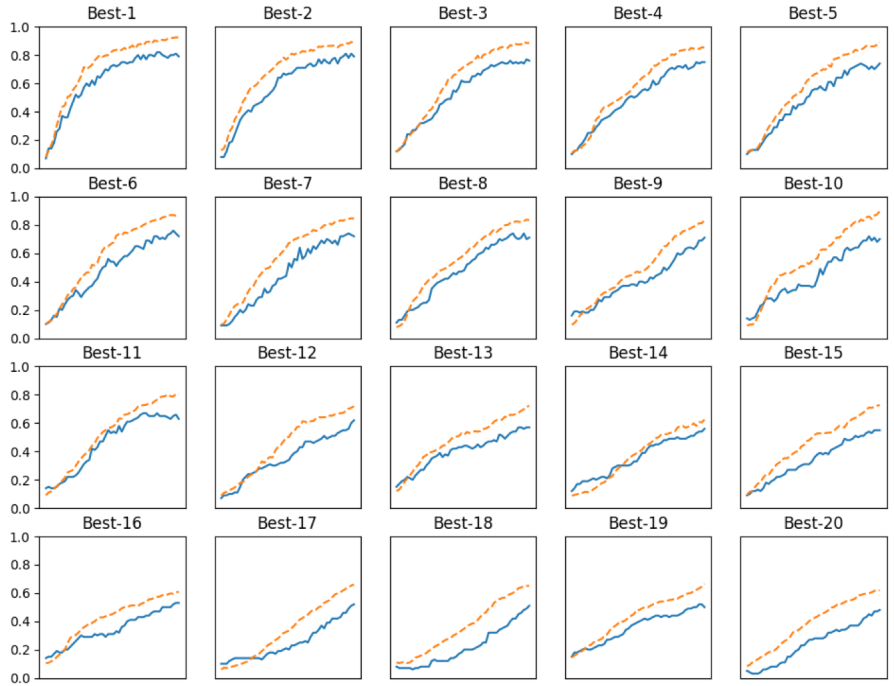
ハイパーパラメータの最適化には次のステップを繰り返す：

STEP0：ハイパーパラメータの範囲を指定
STEP1：設定されたハイパーパラメータの範囲からランダムにサンプリング
STEP2：STEP1でサンプリングされたハイパーパラメータの値を使用して学習を行い、検証データの認識精度を評価
STEP3：ある回数（100回ほど）繰り返し、それらの認識精度の結果からハイパーパラメータの範囲を狭める

```
Best-1(val acc:0.86) | lr:0.009212978164927667, weight
decay:4.585885333273752e-08
Best-2(val acc:0.81) | lr:0.009064674205941432, weight
decay:4.871642962158545e-08
Best-3(val acc:0.8) | lr:0.009416787047186598, weight
decay:3.304592785640497e-07
Best-4(val acc:0.78) | lr:0.009533842347981046, weight
decay:1.775250600714783e-07
Best-5(val acc:0.74) | lr:0.007904064415820061, weight
decay:1.5873180730810917e-07
Best-6(val acc:0.7) | lr:0.006062677697889875, weight
decay:9.98104429626135e-07
```

ハイパーパラメータの決定

実線は検証データの認識精度、点線は訓練データの認識精度




```
Best-1(val acc:0.79) | lr:0.009895533547453095, weight
decay:1.0582731552876836e-08
Best-2(val acc:0.79) | lr:0.007548490792399015, weight
decay:2.9278921862895924e-08
Best-3(val acc:0.76) | lr:0.009309779015595617, weight
decay:4.061709167553873e-08
Best-4(val acc:0.75) | lr:0.008017530712496189, weight
decay:1.611217031551 72e-08
Best-5(val acc:0.74) | lr:0.007757248850656763, weight
decay:1.7645317512188798e-08
Best-6(val acc:0.72) | lr:0.005886089022163818, weight
decay:1.0296598986663 31e-08
Best-7(val acc:0.72) | lr:0.008119405564663465, weight
decay:2.2728771546531516e-08
Best-8(val acc:0.71) | lr:0.007618907557977814, weight
decay:8.759129302751005e-08
Best-9(val acc:0.71) | lr:0.005246422641395342, weight
decay:7.427188763762583e-08
Best-10(val acc:0.7) | lr:0.009069534820041279, weight
```

過学習をしておらず (accuracyが100 %にならない)、検証データと訓練データの隔たりが小さいのはBest-8?

まとめ

4節　正則化

・過学習を抑制するための正則化の技術としてweight decayやdropoutがある

5節　ハイパーパラメータの検証

・ハイパーパラメータの探索は、良い値が存在する範囲を徐々に絞りながら進めるのが効率の良い方法