

scikit-learn、Keras、TensorFlow による
実践機械学習 第 2 版
第 10 章 前半
データ工学研究室 輪読会

AL18036 片岡 凪

芝浦工業大学 工学部 情報工学科 4 年

Aipril 08, 2021

発表者紹介

- ▶ 片岡 凪
- ▶ 千葉県 浦安市
- ▶ 芝浦工業大学 工学部 情報工学科 4 年
- ▶ データ工学研究室（木村昌臣研究室）
- ▶ 関心：画像，XAI，自動化，効率化
- ▶ Twitter @calm_IRL
- ▶ Github KataokaNagi



目次

- ① 第 10 章 人工ニューラルネットワークと Keras の初歩
- ② 10.1 生物学的なニューロンから人工ニューロンへ
 - 10.1.1 生物学的ニューロン
 - 10.1.2 ニューロンによる論理演算
 - 10.1.3 パーセプトロン
 - 10.1.4 MLP とバックプロパゲーション
 - 10.1.5 回帰 MLP
 - 10.1.6 分類 MLP
- ③ 10.3 NN のハイパラの微調整
 - 10.3.1 隠れ層の数
 - 10.3.2 隠れ層あたりのニューロン数
 - 10.3.3 学習率、バッチサイズ、その他のハイパラ

目次

- ① 第 10 章 人工ニューラルネットワークと Keras の初歩
- ② 10.1 生物学的なニューロンから人工ニューロンへ
 - 10.1.1 生物学的ニューロン
 - 10.1.2 ニューロンによる論理演算
 - 10.1.3 パーセプトロン
 - 10.1.4 MLP とバックプロパゲーション
 - 10.1.5 回帰 MLP
 - 10.1.6 分類 MLP
- ③ 10.3 NN のハイパラの微調整
 - 10.3.1 隠れ層の数
 - 10.3.2 隠れ層あたりのニューロン数
 - 10.3.3 学習率、バッチサイズ、その他のハイパラ

第 10 章 概要

- ▶ ANN (Artificial NN)
- ▶ MLP (Multi-layer Perceptron)
- ▶ Keras API
- ▶ ※ ユニット = ニューロン
 - 生物学的な意味に制限しないための用語

目次

- ① 第 10 章 人工ニューラルネットワークと Keras の初歩
- ② 10.1 生物学的なニューロンから人工ニューロンへ
 - 10.1.1 生物学的ニューロン
 - 10.1.2 ニューロンによる論理演算
 - 10.1.3 パーセプトロン
 - 10.1.4 MLP とバックプロパゲーション
 - 10.1.5 回帰 MLP
 - 10.1.6 分類 MLP
- ③ 10.3 NN のハイパラの微調整
 - 10.3.1 隠れ層の数
 - 10.3.2 隠れ層あたりのニューロン数
 - 10.3.3 学習率、バッチサイズ、その他のハイパラ

ANN の歴史 1

- ▶ 1943
 - ANN
 - 命題論理
- ▶ 1960s
 - コネクショニズム (NN の研究)
- ▶ 1990
 - SVM

ANN の歴史 2

▶ 近年

– ANN 再燃？

- 膨大なデータ
- ムーアの法則
- アルゴリズムの改良
- 局所的な最適値は意外と無害
- 流行りと資金

目次

- ① 第 10 章 人工ニューラルネットワークと Keras の初歩
- ② 10.1 生物学的なニューロンから人工ニューロンへ
 - 10.1.1 生物学的ニューロン
 - 10.1.2 ニューロンによる論理演算
 - 10.1.3 パーセプトロン
 - 10.1.4 MLP とバックプロパゲーション
 - 10.1.5 回帰 MLP
 - 10.1.6 分類 MLP
- ③ 10.3 NN のハイパラの微調整
 - 10.3.1 隠れ層の数
 - 10.3.2 隠れ層あたりのニューロン数
 - 10.3.3 学習率、バッチサイズ、その他のハイパラ

BNN の構造

- ▶ 細胞体
- ▶ 樹状突起
- ▶ 軸索
 - 細胞体の数倍から数万倍の長さ
- ▶ 終末分岐
- ▶ シナプス終端

BNN の挙動

- ▶ 活動電位（AP）か信号
- ▶ シナプスからの神経伝達物質
- ▶ 発火

目次

- ① 第 10 章 人工ニューラルネットワークと Keras の初歩
- ② 10.1 生物学的なニューロンから人工ニューロンへ
 - 10.1.1 生物学的ニューロン
 - 10.1.2 ニューロンによる論理演算
 - 10.1.3 パーセプトロン
 - 10.1.4 MLP とバックプロパゲーション
 - 10.1.5 回帰 MLP
 - 10.1.6 分類 MLP
- ③ 10.3 NN のハイパラの微調整
 - 10.3.1 隠れ層の数
 - 10.3.2 隠れ層あたりのニューロン数
 - 10.3.3 学習率、バッチサイズ、その他のハイパラ

ニューロン

- ▶ 複数のバイナリ入力
 - 2つ以上の1が入力されると1となる
- ▶ 1つのバイナリ出力
 - 分岐も可能だが両方同じ値
- ▶ 任意の論理命題に対応
 - 恒等写像
 - AND
 - OR
 - NOT (以下を組み合わせる)
 - 1によって出力を禁止する機構
 - 常に1であるニューロン

目次

- ① 第 10 章 人工ニューラルネットワークと Keras の初歩
- ② 10.1 生物学的なニューロンから人工ニューロンへ
 - 10.1.1 生物学的ニューロン
 - 10.1.2 ニューロンによる論理演算
 - 10.1.3 パーセプトロン
 - 10.1.4 MLP とバックプロパゲーション
 - 10.1.5 回帰 MLP
 - 10.1.6 分類 MLP
- ③ 10.3 NN のハイパラの微調整
 - 10.3.1 隠れ層の数
 - 10.3.2 隠れ層あたりのニューロン数
 - 10.3.3 学習率、バッチサイズ、その他のハイパラ

パーセプトロン

- ▶ ANN アーキテクチャ
- ▶ TLU (Threshold Logic Unit : 閾値論理素子) の NW
 - 別名 LTU (Linear Threshold Unit : 線形閾値素子)
 - 人工ニューロン
 - 入力の加重総和 $x^T w$ をステップ関数 (後述) に代入
 - 単体でも NN のように機能
- ▶ 接続重み

ステップ関数

▶ ヘヴィサイドステップ関数

- $\text{heaviside}(z)$
- 0 以上で 1

▶ 符号関数

- $\text{sgn}(z)$
- 0 は 0
- 他は符号付きの 1

全結合層（密層）

- ▶ 前の層の全てのニューロンが次の全てのニューロンに接続
- ▶ パーセプトロンも全結合層
 - 入力層には恒等写像の入力ニューロン
 - 1 を出力するバイアスニューロン
- ▶ 出力 $h_{\{w, b\}}(x) = \phi(xw + b)$
 - x : インスタンスごとに 1 行、特徴量ごとに 1 列
 - w : 入力ニューロンごとに 1 行、直近の層のニューロンあたりに 1 列
 - b : バイアスベクトル
 - ϕ : ステップ関数などの活性化関数

パーセプトロンの訓練

- ▶ ヘップの法則（ヘップ学習）
 - 同時に発火するニューロン同士の重みを強化
- ▶ 誤った予測をしたニューロンの重みを上げる
 - $w_{i,j}^{(next\ step)} = w_{i,j} + \eta(y_j - \hat{y}_j)x_i$
- ▶ パーセプトロンの収束定理
 - 層の境界が線形なので複雑なパターンには向かないが、
訓練インスタンスが**線形分離可能**なら解が収束

単一 TLU のパーセプトロンの実装

- ▶ scikit-learn の Perceptron クラス
- ▶ 確率的勾配降下法の特別なケースと等価
- ▶ ハード投票分類をするだけ
 - 排他的 OR のような簡単な問題も解けない
 - 確率を出力するロジスティック回帰の方が優れている
 - MLP (Multi-layer Perceptron: 多層パーセプトロン) で解決

```
[2] import numpy as np
    from sklearn.datasets import load_iris
    from sklearn.linear_model import Perceptron

    iris = load_iris()
    X = iris.data[:, (2, 3)] # petal length, petal width
    y = (iris.target == 0).astype(np.int)

    per_clf = Perceptron()
    per_clf.fit(X, y)

    y_pred = per_clf.predict([[2, 0.5]])
```

▶ y_pred

array([0])

MLP の XOR

- ▶ 図で確認

目次

- ① 第 10 章 人工ニューラルネットワークと Keras の初歩
- ② 10.1 生物学的なニューロンから人工ニューロンへ
 - 10.1.1 生物学的ニューロン
 - 10.1.2 ニューロンによる論理演算
 - 10.1.3 パーセプトロン
 - 10.1.4 MLP とバックプロパケーション
 - 10.1.5 回帰 MLP
 - 10.1.6 分類 MLP
- ③ 10.3 NN のハイパラの微調整
 - 10.3.1 隠れ層の数
 - 10.3.2 隠れ層あたりのニューロン数
 - 10.3.3 学習率、バッチサイズ、その他のハイパラ

MLP

- ▶ 入力層、複数の TLU 層、出力層
- ▶ 出力層に近いほど上位層、遠いほど下位層と呼ばれる
 - 一方通行なアーキテクチャを FNN (Feedforward NN: 順伝搬型 NN)
- ▶ 出力層以外にバイアスニューロン
 - 通常は省略される

DNN

- ▶ ANN が深い隠れ層をもったもの
 - 層の数は曖昧になってきている
- ▶ 深層学習という分野で研究されるもの
- ▶ 1986 **バックプロパケーション (誤差逆電波法)** で再燃
 - 勾配降下法の自動化 (リバースモードの自動微分)
 - 高速
 - 正確
 - 微分対象が様々な変数を持ち、出力が少ない場合に最適
 - 往復 1 回で重みの誤差の勾配がわかる

バックプロパケーション

- ▶ 1 度に 1 個のミニバッチ（複数インスタンス）を処理（1 エポック）
- ▶ 手順
 - 全ての接続重みを無作為に初期化
 - 対称性のある初期化をすると、ニューロンが少ない NN のように表現力が下がる
 - 前進パス
 - 処理結果を記録しつつ下位層から順に処理
 - 後退パス
 - 出力誤差をコスト関数で評価
 - 連鎖律で個々の出力接続部の誤差を計算
 - 勾配降下ステップ
 - 誤差が小さくなるように接続重みを調節

バックプロパゲーションのための改良

▶ 活性化関数の改良

- ステップ関数 (カクカク) -> ロジスティック関数 (滑らか)
 - 連鎖律のために勾配が取れるように
 - 生物学的ニューロンに近い
- 双曲線正接
 - $\tanh(z) = 2\sigma(2z) - 1$
 - $-1 < \tanh(z) < 1$ のため、出力が 0 を中心として散らばり収束が早まる
- ReLU 関数
 - $\text{ReLU}(z) = \max(0, z)$
 - 勾配が $z = 0$ で発散し、 $z < 0$ で 0 となるが、高速で最も利用される
 - 出力の最大値がないため、勾配消失問題を解決する (11 章で詳解)
- そもそも活性化関数は、線形関数の組み合わせで
線形問題しか解けなかった問題を解消するための非線形関数

目次

- ① 第 10 章 人工ニューラルネットワークと Keras の初歩
- ② 10.1 生物学的なニューロンから人工ニューロンへ
 - 10.1.1 生物学的ニューロン
 - 10.1.2 ニューロンによる論理演算
 - 10.1.3 パーセプトロン
 - 10.1.4 MLP とバックプロパゲーション
 - 10.1.5 回帰 MLP
 - 10.1.6 分類 MLP
- ③ 10.3 NN のハイパラの微調整
 - 10.3.1 隠れ層の数
 - 10.3.2 隠れ層あたりのニューロン数
 - 10.3.3 学習率、バッチサイズ、その他のハイパラ

MLP による回帰タスク

- ▶ 予測の数 = 出力ニューロンの数
- ▶ 出力ニューロンには活性化関数はいないことが多い
 - 回帰では任意の値を出力させるため
- ▶ 正の数に限定するのであれば
ReLU 関数や softplus 関数を用いることも
 - $\text{softplus}(z) = \ln(1 + \exp(z))$
- ▶ 予測値の範囲を制限するには
ロジスティック関数や双極正接関数に通してスケーリング

回帰 MLP のコスト関数

▶ 平均二乗誤差 $\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2$

- デファクトスタンダード

▶ 平均絶対誤差 $\frac{1}{n} \sum_{i=1}^n |y_i - \bar{y}|$

- 2 乗しないので外れ値に強い

▶ フーバー損失関数 $L_\delta(y, f(x))$

- 誤差が閾値（一般に 1）以下のとき
 - 平均二乗誤差に近い、**高速で正確な式**
- 誤差が閾値より大きいとき
 - 平均絶対誤差に近い、**外れ値に強い式**
- Qiita (2020) 「損失関数のまとめ」

$$|y - f(x)| \leq \delta$$

$$\frac{1}{2}(y - f(x))^2$$

$$|y - f(x)| > \delta$$

$$\delta|y - f(x)| - \frac{1}{2}\delta^2$$

目次

- ① 第 10 章 人工ニューラルネットワークと Keras の初歩
- ② 10.1 生物学的なニューロンから人工ニューロンへ
 - 10.1.1 生物学的ニューロン
 - 10.1.2 ニューロンによる論理演算
 - 10.1.3 パーセプトロン
 - 10.1.4 MLP とバックプロパゲーション
 - 10.1.5 回帰 MLP
 - 10.1.6 分類 MLP
- ③ 10.3 NN のハイパラの微調整
 - 10.3.1 隠れ層の数
 - 10.3.2 隠れ層あたりのニューロン数
 - 10.3.3 学習率、バッチサイズ、その他のハイパラ

Table: MLP による分類タスク

ハイパー パラメータ	二項分類	多ラベル二項分類	多クラス分類
入力 Neuron 数	特徴量ごとに 1 つ	左に同じ	左に同じ
隠れ層の数	一般に 1 ~ 5	左に同じ	左に同じ
隠れ層ごとの ニューロン数	一般に 10 ~ 100	左に同じ	左に同じ
出力 Neuron 数	1	ラベルごとに 1	クラスごとに 1
出力層の 活性化関数	ロジスティック	ロジスティック	ソフトマックス
損失関数	交差エントロピー	左に同じ	左に同じ

目次

- ① 第 10 章 人工ニューラルネットワークと Keras の初歩
- ② 10.1 生物学的なニューロンから人工ニューロンへ
 - 10.1.1 生物学的ニューロン
 - 10.1.2 ニューロンによる論理演算
 - 10.1.3 パーセプトロン
 - 10.1.4 MLP とバックプロパゲーション
 - 10.1.5 回帰 MLP
 - 10.1.6 分類 MLP
- ③ 10.3 NN のハイパラの微調整
 - 10.3.1 隠れ層の数
 - 10.3.2 隠れ層あたりのニューロン数
 - 10.3.3 学習率、バッチサイズ、その他のハイパラ

単純な MLP の調整要素

- ▶ 層の数
- ▶ 層ごとのニューロン
- ▶ 活性化関数
- ▶ 重みの初期化ロジック
- ▶ etc.

ANN の調整の方法 1

- ▶ 手動で調節して交差検証で良いものを選択
 - scikit-learn 風に使うためにラップ
 - 損失ではなくスコアを用いることに注意
 - ハイパーパラメータが多いのでグリッドサーチではなくランダムサーチで一部を評価
 - コード
- ▶ 広範囲のハイパーパラメータ → 最良のハイパーパラメータ付近
 - 時間がかかる
- ▶ ある領域が良いとわかったらズームイン

ズームインするライブラリ 1

- ▶ Hyperopt
 - あらゆるタイプの複雑な探索空間に対応する広く使われるライブラリ
- ▶ Hyperas
 - Keras の最適化ライブラリ
- ▶ kopt
 - Keras の最適化ライブラリ
- ▶ Talos
 - Keras の最適化ライブラリ
- ▶ Keras Tuner
 - Keras の最適化ライブラリ
 - 可視化と解析も可能

ズームインするライブラリ 2

- ▶ Scikit-Optimize
 - GridSearchCV クラスと似たベイズ最適化が可能
- ▶ Spearmint
 - ベイズ最適化ライブラリ
- ▶ Hyperband
 - 高速
- ▶ Sklearn
 - 進化的アルゴリズム
 - GridSearchCV 風

ズームインするツール

- ▶ Google Cloud API のハイパーパラメータ調整サービス
- ▶ Arimo
- ▶ SigOpt
- ▶ CallDesk の Oscar
- ▶ DeepMind (2017) “Population Based Training of Neural Networks”
- ▶ Google の AutoML スイート

目次

- ① 第 10 章 人工ニューラルネットワークと Keras の初歩
- ② 10.1 生物学的なニューロンから人工ニューロンへ
 - 10.1.1 生物学的ニューロン
 - 10.1.2 ニューロンによる論理演算
 - 10.1.3 パーセプトロン
 - 10.1.4 MLP とバックプロパゲーション
 - 10.1.5 回帰 MLP
 - 10.1.6 分類 MLP
- ③ 10.3 NN のハイパラの微調整
 - 10.3.1 隠れ層の数
 - 10.3.2 隠れ層あたりのニューロン数
 - 10.3.3 学習率、バッチサイズ、その他のハイパラ

隠れ層の数

▶ 浅い NN

- 複雑でなければ隠れ層は 1, 2 個で十分なことが多い
- MNIST + 隠れ層 1 個で 97%

▶ 深い NN

- 森を描くときに枝葉や木をコピペできるようなイメージ
- 指数的に少ないニューロンで済む
- 収束が早い
- 汎化能力が高い
 - 髪の実験に頭部の分類器の下位層を使いまわせる
 - **転移学習**という
- 過学習の手前まで層を増やす

目次

- ① 第 10 章 人工ニューラルネットワークと Keras の初歩
- ② 10.1 生物学的なニューロンから人工ニューロンへ
 - 10.1.1 生物学的ニューロン
 - 10.1.2 ニューロンによる論理演算
 - 10.1.3 パーセプトロン
 - 10.1.4 MLP とバックプロパゲーション
 - 10.1.5 回帰 MLP
 - 10.1.6 分類 MLP
- ③ 10.3 NN のハイパラの微調整
 - 10.3.1 隠れ層の数
 - 10.3.2 隠れ層あたりのニューロン数
 - 10.3.3 学習率、バッチサイズ、その他のハイパラ

層ごとのニューロン数

- ▶ 上位層に向けて萎むパターン
 - 従来手法
- ▶ 全ての層で同じ数にするパターン
 - 多くの場合、性能は同じか向上
 - ハイパーパラメータが n 層分から 1 つに統一可能
- ▶ 過学習の手前まで数を増やすパターン
- ▶ ストレッチパンツアプローチ
 - 必要以上に多く取り、早期打ち切りや正則化で過学習対策
 - ボトルネックとなる層が発生しない
- ▶ ニューロン数より層の数を増やした方が効果は得やすい

目次

- ① 第 10 章 人工ニューラルネットワークと Keras の初歩
- ② 10.1 生物学的なニューロンから人工ニューロンへ
 - 10.1.1 生物学的ニューロン
 - 10.1.2 ニューロンによる論理演算
 - 10.1.3 パーセプトロン
 - 10.1.4 MLP とバックプロパゲーション
 - 10.1.5 回帰 MLP
 - 10.1.6 分類 MLP
- ③ 10.3 NN のハイパラの微調整
 - 10.3.1 隠れ層の数
 - 10.3.2 隠れ層あたりのニューロン数
 - 10.3.3 学習率、バッチサイズ、その他のハイパラ

MLP の重要なハイパーパラメータ 1

▶ 学習率

- 一般に、出力値が発散し始める値の半分が最適
- 実装例： 10^{-5} から数百回イテレートして 10 へ
- 対数軸を取って損失をグラフ化する
- 一般に、折り返しの $\frac{1}{10}$ 手前が最適
 - 過学習を恐れて？

▶ オプティマイザ

- ミニバッチ勾配降下法の代替法
- 11 章で詳解

MLP の重要なハイパーパラメータ 2

▶ バッチサイズ

- RAM の制限を受ける
- 大きいほど GPU が同時に沢山処理
- 大きすぎると訓練初期に不安定 & 汎化しない
 - 2~32 が推奨されたりされなかったり
 - 段々と大きくしていく手法も存在

▶ 活性化関数

- 出力層以外は ReLU 関数でよい

▶ イテレーション数

- ほぼ操作することはないが、早期打ち切りで意識することがある

NN のハイパーパラメータ調整のベストプラクティス

- ▶ Leslie N. Smith (2018) "Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour"