

## パターン認識・課題4

### 課題4.1 ニューラルネットワークによる学習(基本編)

#### 課題4.1.1 ニューラルネットワーク学習プログラムの作成

ニューラルネットワークによる学習を実現するプログラムを作成せよ。ここでは、ニューラルネットワークは、入力層、中間層(隠れ層)、出力層からなるものとする。

このとき、特徴ベクトル(拡張特徴ベクトル)の要素数、パターン数、中間層の要素数、およびクラス数は任意の数を設定できるように作ること。

#### 課題4.1.2 ニューラルネットワーク学習プログラムの実行

完成したプログラムに、以下のデータ(課題3と同じ)を入力し、正しく学習することを確認せよ。

ただし、

$\rho=0.1$

とする。

パターン	値	クラス
パターン1	(1,1)	$\omega_1$
パターン2	(2,1)	$\omega_1$
パターン3	(1,3)	$\omega_2$
パターン4	(2,4)	$\omega_2$
パターン5	(4,3)	$\omega_3$
パターン6	(4,2)	$\omega_3$

学習したパーセプトロンに以下のパターンを入れ、どのクラスと判定されるか確認せよ。

パターン	値	クラス
パターン7	(2,2)	$\omega_1$

## ヒント

### 教師信号の表現

誤差逆伝播法でのプログラムを書く場合、正解ラベルは、クラス数に応じたOne-hot vector表現にしておくのが良い。

One-hot vector表現とは、クラス数がC個の場合、C次元のベクトルであり、正解クラスに対応する次元が1、それ以外が0であるベクトルのこと。

この表現を用いると、出力層での誤差の評価が容易になる。

```
/* 例 */
int labels[P][C];
labels[0][0] = 1; labels[0][1] = 0; labels[0][2] = 0;
labels[1][0] = 1; labels[1][1] = 0; labels[1][2] = 0;
labels[2][0] = 0; labels[2][1] = 1; labels[2][2] = 0;
...
```

Yasutomo KAWANISHI

## 誤差逆伝播法の手順

1. 中間層と出力層の初期重みを決定する。
2. 入力層から順番に、各ユニットの出力を計算していく。(中間層の出力を $g_1$ , 出力層の出力を $g_2$ とすると)
  1. 中間層のユニット数分ループしながら $g_1[j]$ の値を計算。
  2. 出力層のユニット数分ループしながら $g_2[k]$ の値を計算。
3. 教科書P.45 式(3.68)や、森先生のスライドの30ページ辺りを参考に重みを修正。(入力層 → 中間層の重みを $w_1$ , 中間層 → 出力層の重みを $w_2$ とすると)
  1.  $\epsilon[j]$ を計算.  $w_2[j][k]$ の値を修正。
  2.  $w_1[i][j]$ の値を修正。
4. 教科書P.47-48ページを参考に、収束の判定を行う。

## 課題4.2 ニューラルネットワークによる学習(応用編)

もう少し現実的なデータでの実験を行う。

ここでは、機械学習のサンプルとして良く利用される、irisデータセットを利用する。

irisデータセットとは、アヤメのがく片の長さ、幅、花びらの長さ、幅の4つの特徴量を用いて、3種類のアヤメを分類する問題のためのデータセットである。

以下に例を示す。各行の4つの小数点数は上記4つの特徴量、最後の整数値はアヤメのクラスを表す。

```
4.700000 3.200000 1.300000 0.200000 0
5.700000 2.800000 4.100000 1.300000 1
6.900000 3.200000 5.700000 2.300000 2
6.000000 2.700000 5.100000 1.600000 1
6.400000 2.900000 4.300000 1.300000 1
5.700000 2.900000 4.200000 1.300000 1
6.300000 3.400000 5.600000 2.400000 2
6.300000 2.800000 5.100000 1.500000 2
5.100000 3.500000 1.400000 0.300000 0
6.400000 2.800000 5.600000 2.200000 2
```

irisデータセット(学習用) ([http://www.murase.m.is.nagoya-u.ac.jp/~kawanishiy/lecture/patternrecognition2018//exercise/iris\\_train.dat](http://www.murase.m.is.nagoya-u.ac.jp/~kawanishiy/lecture/patternrecognition2018//exercise/iris_train.dat))をダウンロードし、ニューラルネットワークで学習せよ。

学習が済んだら、irisデータセット(評価用) ([http://www.murase.m.is.nagoya-u.ac.jp/~kawanishiy/lecture/patternrecognition2018//exercise/iris\\_test.dat](http://www.murase.m.is.nagoya-u.ac.jp/~kawanishiy/lecture/patternrecognition2018//exercise/iris_test.dat))をダウンロードし、各サンプルがどの程度正しく認識できるかを調べよ。認識率が低い場合は、中間層のユニット数を増やして結果を確認せよ。

## 課題4.3 ニューラルネットワークで遊ぶ

A Neural Network Playground (<https://playground.tensorflow.org/>)で、対象とするデータ、入力する特徴量(features)、中間層の数(hidden layers)やユニット数(neurons)をいろいろ変更し、どのような結果になるか調査せよ。

また、活性化関数(Activation)も、シグモイド関数(Sigmoid)だけでなく、いろいろあるので試してみて、どのような違いがあるのかを調べよ。この時、それぞれの関数がどのような形の関数なのかも調査せよ。

## 発展課題

### 発展課題Lv.1(オプション)

中間層の数を1層だけでなく、2層(入力層 → 中間層1 → 中間層2 → 出力層)へと拡張せよ。

### 発展課題Lv.2(オプション)

中間層の数を1層だけでなく、N層(入力層 → 中間層1 → 中間層2 → ... 中間層N → 出力層)へと拡張せよ。

ただし、Nは定数で良い。

## 提出期限

2018/6/25

レポート提出の注意は, [こちら \(../report.html\)](#)

[戻る](#)