

# パターン認識 2017年度後学期

## レポートⅡ

氏名 伊藤源太 クラス 3EP4 クラス番号 6 貢献度 50 %

氏名 河村天哉 クラス 3EP4 クラス番号 14 貢献度 50 %

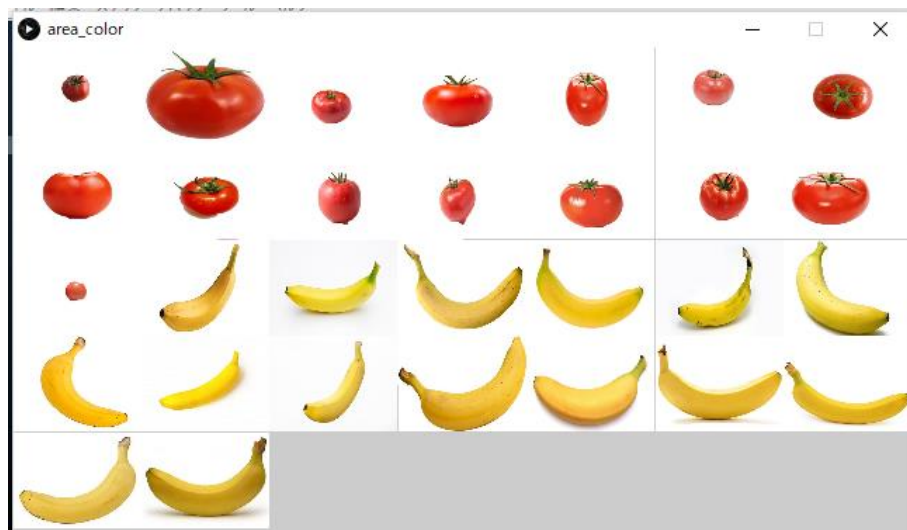
- (1) 分類する2種類のものを決める。

種類 A トマト 種類 B バナナ

- (2) 以上の2種類のものを分類するとき用いる2つの特徴量を決める。

特徴量1 面積 特徴量2 色

- (3) 三種類のものの画像それぞれ15枚用意する、全部で30枚の画像を用意する。(三人グループの場合、それぞれ20枚用意する、全部で40枚を用意する)
- (4) PhotoShop を利用して、画像の前処理を行い、最終的にすべての画像のサイズが320x240になるように処理する。すべての画像を pic01.jpg~pic60.jpg のように統一したファイル名を付け、jpeg フォーマットで保存する。
- (5) 画像を番号順に読み込み、表示するプログラムを作成する。



- (6) それぞれの画像に対して、特徴量1と特徴量2を計算する。結果をファイルに保存する。

area.txt - メモ帳	area2.txt - メモ帳
<pre> 3.506 5.7871537 4.7934 8.280876 0.3168 8.080114 1.187 9.887001 0.59 10.324035 0.0278 4.8043165 1.4466 7.939658 1.4176 9.690236 1.4126 8.779584 0.7854 8.058289 0.1512 8.139484 0.2186 7.2734213 0.9076 8.394755 1.055 10.285687 0.0348 8.329597 </pre>	<pre> 2.554 0.7274007 0.176 0.79295456 0.424 1.0357078 0.34600002 1.0380347 0.966 0.63180125 1.5780001 0.95564 0.252 1.0490478 0.0019999999 0.61 0.146 0.8436987 1.008 0.99150795 0.14 0.91057146 0.51600003 1.0131783 0.172 0.91639537 0.308 0.99902594 1.692 1.0630615 </pre>

トマトの面積と色

バナナの面積と色

ここまで 8 点

(7) 特徴量 1 と特徴量 2 に基づいて、パーセプトロンを求め、2 つのグループに分類するプログラムを作成する。

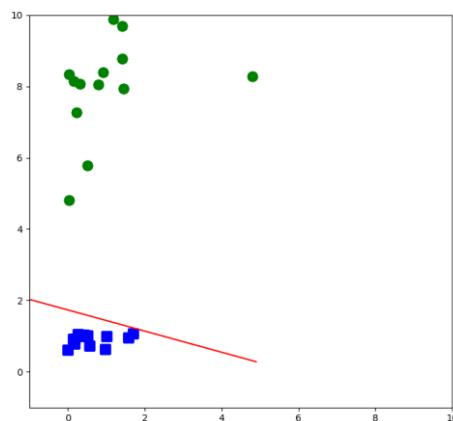
```

2 import numpy as np
3 import matplotlib.pyplot as plt
4 import matplotlib.animation as animation
5
6 # 非活性化関数 g(x) のメソッド
7 def predict(w, x):
8     out = np.dot(w, x)
9     if out >= 0:
10         o = 1.0
11     else:
12         o = -1.0
13     return o
14
15 # プロットするメソッド
16 def plot(wvec, x1, x2):
17     x_fig = np.arange(-2, 5, 0.1)
18     fig = plt.figure(figsize=(8, 8), dpi=100)
19     ims = []
20     plt.xlim(-1, 10.0)
21     plt.ylim(-1, 10.0)
22
23     # プロットする
24     for w in wvec:
25         y_fig = [-(w[0] / w[1]) * x1 - (w[2] / w[1]) for x1 in x_fig]
26         plt.scatter(x1[:, 0], x1[:, 1], marker='o', color='g', s=100)
27         plt.scatter(x2[:, 0], x2[:, 1], marker='s', color='b', s=100)
28         ims.append(plt.plot(x_fig, y_fig, "r"))
29
30     for i in range(10):
31         ims.append(plt.plot(x_fig, y_fig, "r"))
32     ani = animation.ArtistAnimation(fig, ims, interval=1000)
33     plt.show()
34
35 # 出力を求める
36 o = []
37 while t != o:
38     o = [] # 初期化
39     # 学習ループ
40     for i in range(class_x.shape[0]):
41         out = predict(wvec[-1], class_x[i, :])
42         o.append(out)
43         if t[i] * out < 0: # 出力と教師ラベルが異なるとき
44             wvectmp = mu * class_x[i, :] * t[i] # w を変化させる量
45             wvec.append(wvec[-1] + wvectmp) # 重みの更新
46             plot(wvec, x1, x2)

```

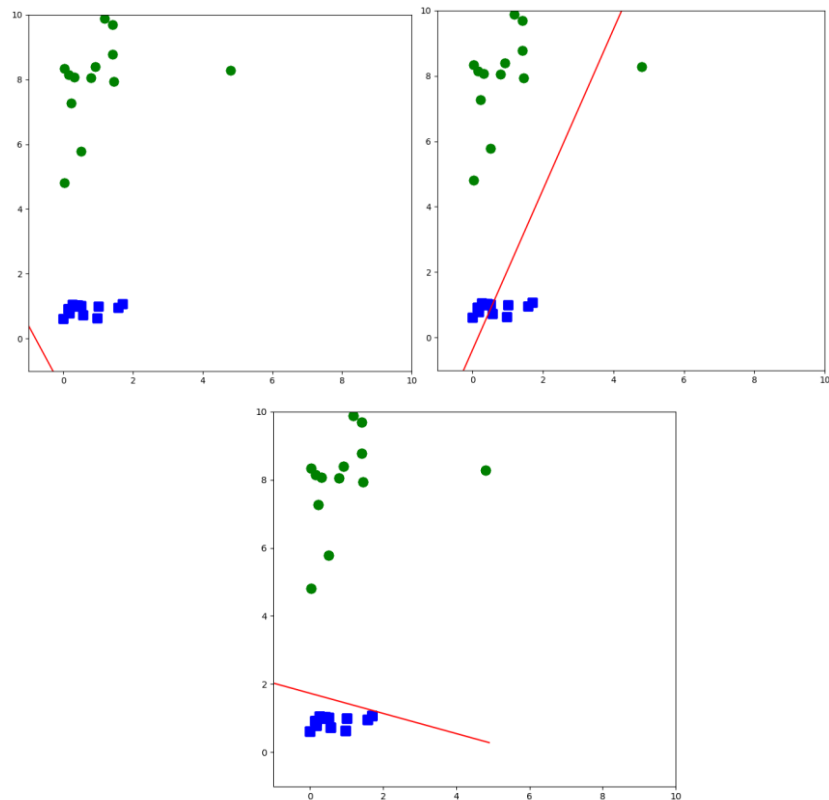
ここまで 16 点

(8) (7) の結果を可視化する



ここまで 18 点

(9) (7) の処理の流れを、CG アニメーションを用いて可視化する。



——— ここまで20点

- レポートには、できるだけ多くの画像を入れるようにしてください。
- レポートには、作成者各自の貢献度を必ず入れてください。
- プログラムもレポートも締切日までポータルを利用して提出してください。締切日は  
1月16日 16:00まで