

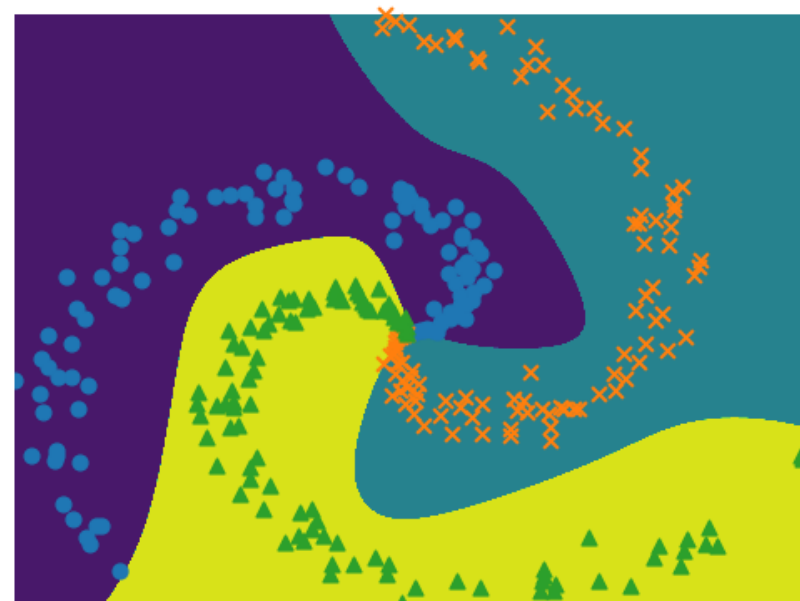
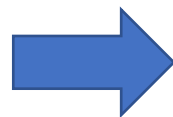
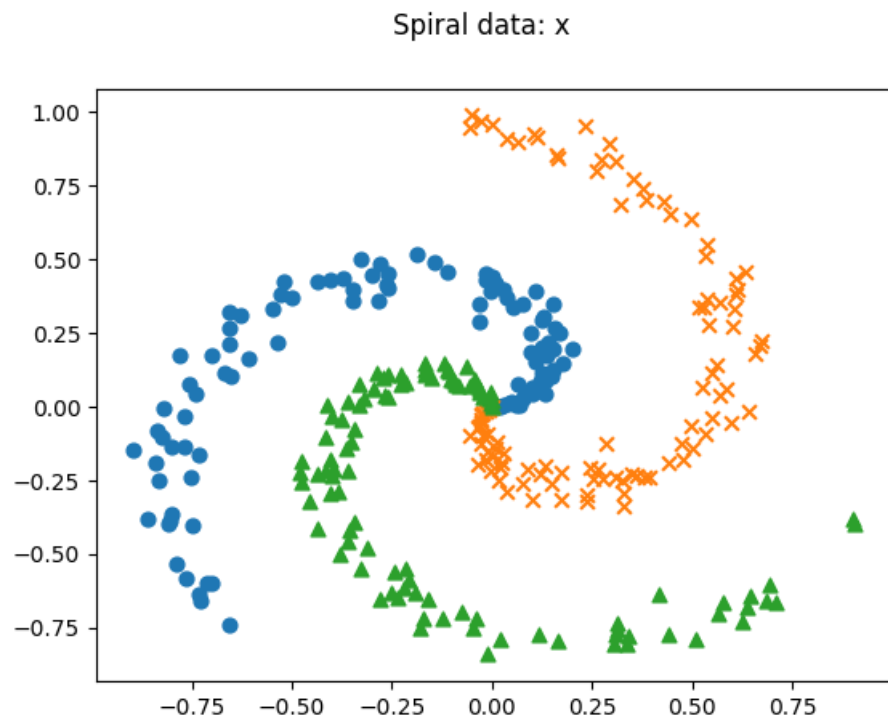
# 数値解析 2

## 第 3 回

静岡理科大学  
コンピュータシステム学科  
幸谷智紀

# 当面の目標

- スパイラルデータに基づいて平面領域を色分けする



- spiral\_dataset\_show.py : spiral.py生成のデータを表示する
- dataset/spiral.py: スパイラル点生成

# DeZeroによる点列の分類

- 斎藤康毅「ゼロから作るDeep Learning」による実装
- NumPyを使ったスクラッチ実装
- 深層学習の概略を，Pythonコードを作りながら理解しやすい

→点列の分類をDeZeroで実行する

# スパイラルデータを分類する: train\_custom\_loop.py

```
1  # train_custom_loop.py: 学習用コード
2  import sys
3  sys.path.append('..')
4  import numpy as np
5  import matplotlib.pyplot as plt
6
7  # dataset/spiral.py
8  from dataset import spiral
9  # common/optimizer.py
10 from common.optimizer import SGD
11 # two_layer_net.py
12 from two_layer_net import TwoLayerNet
13
14 # (1) Hyper parametersの設定
15 max_epoch = 300 # epochは学習の単位: 「1 epoch = 全てのデータを1
    回学習」
16 batch_size = 30
17 hidden_size = 10
18 learning_rate = 1.0
19
20 # (2) データの読み込み、モデルとオプティマイザの設定
21 x, t = spiral.load_data()
22 model = TwoLayerNet(input_size = 2, hidden_size =
    hidden_size, output_size = 3)
23 optimizer = SGD(lr = learning_rate)
```

- train\_custom\_loop.py . . . メインルーチン
- dataset/spiral.py . . . 学習データ生成
- common/optimizer.py . . . SGD最適化
- two\_layer\_net.py . . . 2層NN
  - common/layers.py . . . 各種変換クラス
  - common/functions.py . . . 各種関数定義

# スパイラルデータ生成:dataset/spiral.py

- クラス数(CLS)
- サンプル数(N)
- スパイラルデータの分布を変えて何通りか作ってみる。

|         | CLS = 3 | CLS = 5 |
|---------|---------|---------|
| N = 100 |         |         |
| N = 200 |         |         |

```
1  # spiral.py: スパイラル点の分離
2  import numpy as np
3
4  # データの生成と読み込み
5  def load_data(seed = 20201027):
6      np.random.seed(seed) # 乱数の種をセット
7      N = 100 # クラスごとのサンプル数
8      DIM = 2 # データの次元数
9      CLS_NUM = 3 # クラス数
10
11     x = np.zeros((N * CLS_NUM, DIM))
12     t = np.zeros((N * CLS_NUM, CLS_NUM), dtype = np.int)
13
14     for j in range(CLS_NUM):
15         for i in range(N): # N * j, N * (j + 1)
16             rate = i / N
17             radius = 1.0 * rate
18             theta = j * 4.0 + 4.0 * rate + np.random.randn()
19                 * 0.2
20
21             ix = N * j + i
22             x[ix] = np.array([
23                 radius * np.cos(theta),
24                 radius * np.sin(theta)
25             ]).flatten()
26             t[ix, j] = 1
27
28     return x, t
```

# メインルーチン: spiral\_dataset\_show.py

```
1  # spiral_dataset_show.py: spiral.py生成のデータを表示する
2  import sys # システム設定用
3  sys.path.append('..') # 親ディレクトリにパスを通す
4  from dataset import spiral # dataset/spiral.pyを読み込む
5  import matplotlib.pyplot as plt # グラフ描画用
6
7  # spiralデータを読み込み、xとtのデータ形式を確認
8  x, t = spiral.load_data()
9  print('x -> ', x.shape, x.dtype) # (300, 2) float64... 2次元
   300個のdouble型データ
10 print('t -> ', t.shape, t.dtype) # (300, 3) int64... 3次元300
   個のlong型データ
11 #print('x = \n', x[:,0], x[:,1])
12
13 # グラフ表示
14 fig, ax = plt.subplots()
15
16 # 一色
17 #ax.scatter(x[:, 0], x[:, 1]) # 散布図
```

```
25 # クラスごと塗分け
26 #CLS_NUM, N = 3, 100
27 markers = ['o', 'x', '^', '+', '*'] # 5クラスまで対応
28 for i in range(CLS_NUM): # CLS_NUM
29     ax.scatter(x[i * N: (i + 1)*N, 0], x[i * N: (i + 1) * N, 1], s = 40, marker =
        markers[i])
30
31 fig.suptitle('Spiral data: x')
32 plt.show() # グラフの画面表示
33 fig.savefig('spiral' + str(CLS_NUM) + '_' + str(N) + '.png')
```

# 本日の課題

4種類のスパイラルデータを生成し，図示せよ。

|         | CLS = 3 | CLS = 5 |
|---------|---------|---------|
| N = 100 |         |         |
| N = 200 |         |         |