ホップフィールドネットワークの応用

2017年5月26日

今回の内容

- ホップフィールドネットワークのプログラム解説。
- ホップフィールドネットワーク応用。
- 輪読 (3-1～3-2)。

コードについて

コードの解説は、前回のスライドと照らし合わせながら見てみると良い。（このコードにおける学習・思い出しの計算の解説は前回スライドで行っている。）

main

```
初期状態
Apple    -1
Banana   -1
Red      -1
Yellow   -1
Name: Activation, dtype: int64
output  Apple  Banana  Red  Yellow
input
Apple       0       0    0       0
Banana      0       0    0       0
Red         0       0    0       0
Yellow      0       0    0       0
```

```
names = ['Apple','Banana','Red','Yellow']
neurons = pd.Series([-1]*len(names),index=names,name='Activation')
print ("初期状態")
print (neurons)

links = pd.DataFrame(np.zeros(shape=(len(neurons),len(neurons))),index=names, columns=names)
links.index.name = 'input'
links.columns.name = 'output'
print (links)
print ("¥n")
```

- 「リンゴ」、「バナナ」、「赤」、「黄色」という神経作成。
- 4つの神経は初期状態が-1 (neurons)
- 神経の数×神経の数の脳内状態を作成 初期値は全て0 (links)
- Indexやcolumnsは表示をわかりやすくするため。

main

```
print ("赤いリンゴを見せる")
learn_event(['Red', 'Apple'], links)
print (links)
print ("¥n")

print ("黄色いバナナを見せる")
learn_event(['Yellow', 'Banana'], links)
print (links)
print ("¥n")
```

```
赤いリンゴを見せる
output   Apple  Banana  Red  Yellow
input
Apple        0      -1    1      -1
Banana      -1       0   -1       1
Red          1      -1    0      -1
Yellow      -1       1   -1       0

黄色いバナナを見せる
output   Apple  Banana  Red  Yellow
input
Apple        0      -2    2      -2
Banana      -2       0   -2       2
Red          2      -2    0      -2
Yellow      -2       2   -2       0
```

- 「赤」と「リンゴ」で学習
- 「黄色」と「バナナ」で学習
- 前回スライドp.15~17参照。

learn_event関数

```
def learn_event(event, links):
    pairs = [(x,y) for x in links.index for y in links.columns if x!= y]
    for (x,y) in pairs:
        if (x in event) & (y in event):
            links.ix[x, y] += 1
        elif (x in event) & (not y in event):
            links.ix[x, y] -= 1
        elif (x not in event) & (y in event):
            links.ix[x, y] -= 1
        else:
            links.ix[x, y] += 1
```

脳内状態

output / input	Apple	Banana	Red	Yellow
Apple	0	-2	2	-2
Banana	-2	0	-2	2
Red	2	-2	0	-2
Yellow	-2	2	-2	0

- 受け取った2つの神経の名前(“Red”,”Apple”等)から学習をして、脳内状態を更新する。
- 脳内状態において神経の名前をペアで順番に見ていき、受け取った名前同士のペアや受け取ってない名前同士のペアなら+1、受け取ったものと受け取っていないもののペアなら-1する。
  学習理論は6回スライドのP.7,8参照

main

```
print ("赤を刺激")
neurons['Red'] = 1
print (neurons)
print ("赤といえば？")
run_hopfield(neurons, links)
print (neurons)
print ("¥n")
```

- neuronsのうち、「赤」の部分を1にして刺激状態にする。
- この状態で一度neurons表示 ([-1,-1,1,-1])
- このneuronsと、現在の脳内状態(links)から何が連想されるかrun_hopfieldによって更新されたneuronsが返ってくるので再度表示。
  ([1,-1,1,-1])

run_hopfield関数

```
def run_hopfield(neurons, links):
    for index, value in links.iterrows():
        a = (value * neurons).sum()
        if a >= 0:
            neurons[index] = 1
        else:
            neurons[index] = -1
```

- 思いだし作業で、6回スライドのP.18-23の作業を行っている。（linksを1行ずつneuronsと掛け合わせてその度にneurons更新といった具合）。

main

```
print ("黄色いリンゴを見せる")
learn_event(['Yellow', 'Apple'], links)
print (links)
print ("¥n")
```

- 「黄色」と「リンゴ」で学習

main

```
print ("赤を刺激、リンゴを刺激していない状態に")
neurons['Red'] = 1; neurons['Apple'] = -1
print (neurons)
print ("赤といえば？")
run_hopfield(neurons, links)
print (neurons)
```

- 再度、赤だけを刺激して思いだしをするのだが、プログラムではneuronsが[1,-1,1,-1]になったままなので、少なくともリンゴは-1に戻す。そして赤だけを刺激状態に。詳細は次ページ。
- その後はスライド7と同様。

思い出しのプログラム上の補足

- 思い出し作業を行った後は、神経(neurons)が計算が終わった後の状態のままになってしまう。
- そのため、1回思い出しをした後に思い出しをする場合は、全ての神経を-1にしてから連想させる対象だけを1にするように記述すると間違いをせずに済むはず。(全て記述してしまうということ)
- 例:前ページのコードの場合の改訂案。

```
neurons['Red'] = 1; neurons['Apple'] = -1
```

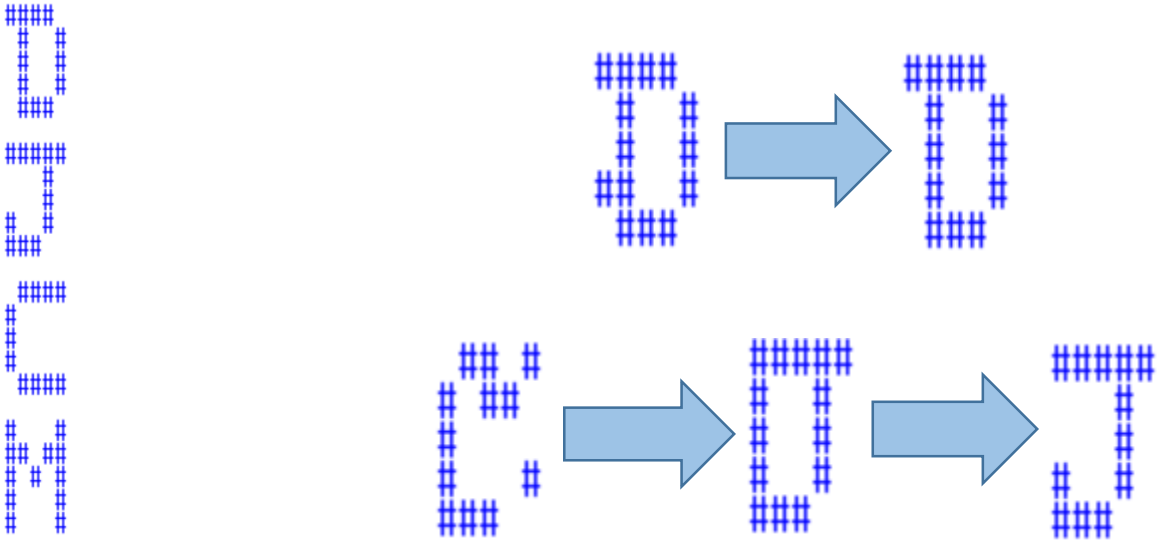
```
neurons['Red'] = 1; neurons['Apple'] = -1;neurons['Banana'] = -1; neurons['Yellow'] = -1
```

演習

- Hopfield.pyのコードを解読してみる。
  解読・理解した上で応用をしてみる。
- 神経を増やしてみる。
- 同時に3つ以上の神経を活動させた学習をさせてみる。
  ・・・等。
- 文字認識・画像認識

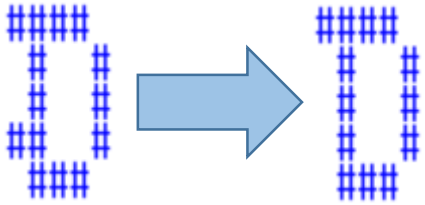
応用例（文字認識）

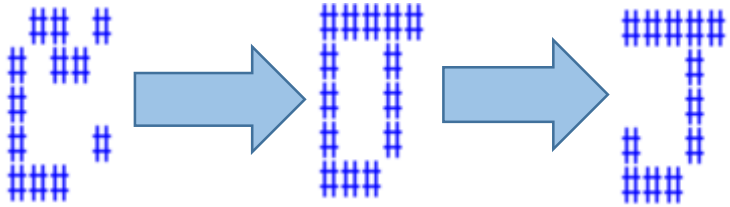
- http://enakai00.hatenablog.com/entry/20150108/1420719651
- コードはリンク先から。
- 赤、黄、リンゴ、バナナの問題以外にも、文字認識に使用する例が紹介されているので、参考に実装してみると良い。



文字認識ヒント

- SeriesはnumpyのIの機能
- そのままコードを使うと、＃の表示がおかしいバグがあるため自分で直す。
- 思いだし作業を行う回数（学習させた内、どれかに一致するまで繰り返す）は、自分で決めている。（下図の場合、たまたまDは1回、Jは2回で思いだせた）





発展応用

- 画像への応用
  http://rishida.hatenablog.com/entry/2014/03/03/174331
- 巡回セールスマン問題
  http://www-ailab.elcom.nitech.ac.jp/lecture/neuro/bz4.html
  http://www.cis.twcu.ac.jp/~asakawa/chiba2002/lect6-mutual/mutual.html

3年生輪読予定表改訂版

担当箇所	名前	発表予定日
1-1	保科	5/12
1-3	猪狩	5/12
2-1	村田 西澤	5/19
2-2 2-3	西ヶ谷 坂中	5/19
3-1	**米倉** **田之井**	**5/26**
3-2	**松田** **清水**	**5/26**

3年生輪読予定表改訂版

担当箇所	名前	発表予定日
3-3	宿野 前原 村田	6/2
4章	門木 山下 上野 數見	6/9 6/16
5章	下窪 黒川 関根 坂本	6/16

次回

- 誤差逆伝播法？
- 輪読(3-3)。
- 3年生向けの補足説明。