

# ヘッブの法則

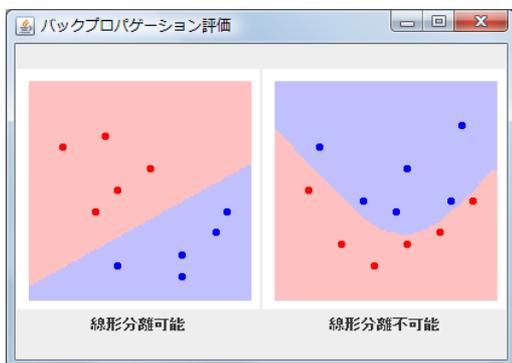
2016年4月29日

# ニューラルネットのアルゴリズム

様々なものが存在するが、4つを紹介予定

## 誤差逆伝播法

- 多層ネットワーク



<http://d.hatena.ne.jp/nowokay/20080701/1214915017>

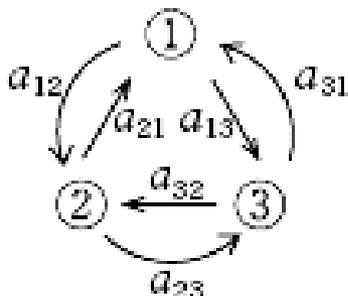
## 自己組織化写像

- 自動分類



## ヘップの法則

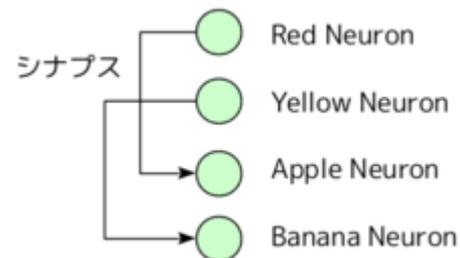
- 人間の脳活動



[http://gaya.jp/spiking\\_neuron/matrix.htm](http://gaya.jp/spiking_neuron/matrix.htm)

## ホップフィールド

- 連想記憶



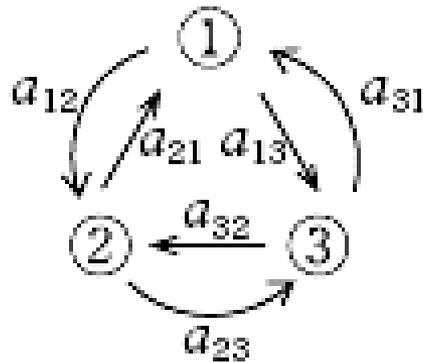
<http://enakai00.hatenablog.com/entry/20150108/1420719651>

# ニューラルネットのアルゴリズム

- 今回はヘップの法則。
- アルゴリズムの動きについての参照、引用:  
[http://gaya.jp/spiking\\_neuron/matrix.htm](http://gaya.jp/spiking_neuron/matrix.htm)

## ヘップの法則

- 人間の脳活動



# ヘッブの法則

- ヘッブの法則(ヘッブのほうそく)は、脳のシナプス可塑性についての法則である。ヘッブ則、ヘブ則とも呼ばれる。心理学者のドナルド・ヘッブによって提唱された。ニューロン間の接合部であるシナプスにおいて、シナプス前ニューロンの繰り返し発火によってシナプス後ニューロンに発火が起こると、そのシナプスの伝達効率が增強されるというものである。

出典: フリー百科事典『ウィキペディア (Wikipedia)』

- 何度も同じことを反復学習することで、定着する。望んでいるか望んでいないかにも関わらず、繰り返した行動が習慣化する。

# 脳の可塑性

- 脳の可塑性とは「発達段階の神経系が環境に応じて最適の処理システムを作り上げるために、よく使われるニューロンの回路の処理効率を高め、使われない回路の効率を下げるという現象」

引用:

<http://www.cscd.Osaka-u.ac.jp/user/rosaldo/110505PezBrain.html>

# ヘッブの法則(神経の表記)

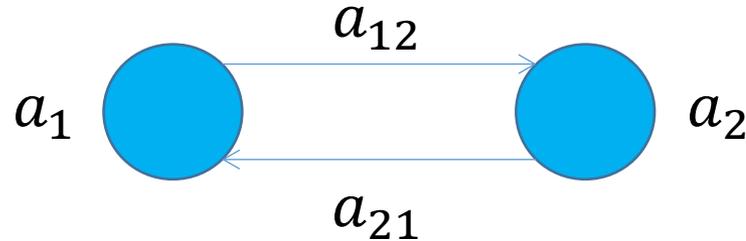
- $a_1$ と $a_2$ という神経があるとする。 $a_1$ と $a_2$ の神経の状態を
$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

と表す。ある刺激を受けて活動があった神経は1、なかった神経は-1と表す。

- $a_1$ と $a_2$ がともに刺激で活動したならば
$$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

と表す。

# ヘップの法則(脳内状態)



- さらに、神経が2つの場合は、 $a_1$ から $a_2$ へ伸びているシナプス $a_{12}$ と、 $a_2$ から $a_1$ へ伸びているシナプス $a_{21}$ が存在する。
- この時の脳内の状態は

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

と表す。

自分から自分へシナプスは伸びないので、 $a_{11}$ と $a_{22}$ は常に0とする。

# ヘッブの法則（脳内状態の変化）

神経と神経の間の結びつきの強さの変化の法則は神経が活動した際に、

- ① 両方の神経が同時に活動したならば、結びつきを強める。
- ② 両方の神経が同時に活動しなかったならば、結びつきを弱める。

のどちらかに従う動きになる。

# ヘップの法則(神経2つでの学習状態1)

- $\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$  の  $a_1$  と  $a_2$  がともに1(どちらも活動した)ならば、①より、 $a_1$  と  $a_2$  の間のシナプス  $a_{12}$  と  $a_{21}$  の結びつきを強める。

値の動きとしては、先程の行列内の  $a_{12}$  と  $a_{21}$  の値に+1する。

$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$  というものを学習させた場合の  $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$  は初期状態

$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$  から変化して

$$\begin{pmatrix} 0 & 0+1 \\ 0+1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

となる。

# ヘッブの法則(神経2つでの学習状態2)

- $\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$  の  $a_1$  が1、 $a_2$  が-1 ( $a_1$  のみ活動した) ならば、②より、 $a_1$  と  $a_2$  の間のシナプス  $a_{12}$  と  $a_{21}$  の結びつきを弱める。

値の動きとしては、先程の行列内の  $a_{12}$  と  $a_{21}$  の値に-1する。

よって、

$\begin{pmatrix} 1 \\ -1 \end{pmatrix}$  というものを学習させた場合の  $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$  は

$$\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$$

$a_1$  が-1、 $a_2$  が1の場合も②に従い同じ行列となる。

## ヘッブの法則(神経2つでの学習状態3)

- $\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$  の  $a_1$  と  $a_2$  がどちらも -1 (どちらも活動しない) ならば、①にも②にも従わず、 $a_1$  と  $a_2$  の間のシナプス  $a_{12}$  と  $a_{21}$  の結びつきは変化しない。

よって、

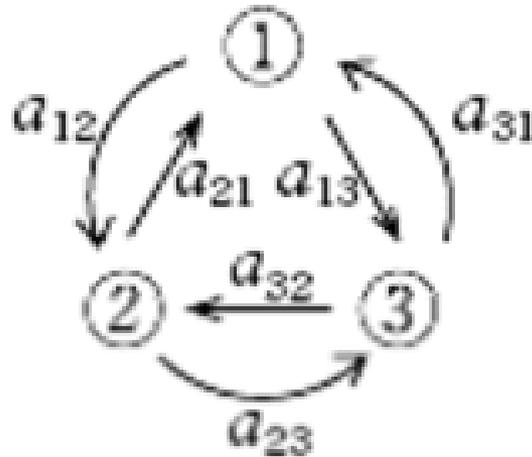
$\begin{pmatrix} -1 \\ -1 \end{pmatrix}$  というものを学習させた場合の  $\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$  は

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

となる。

# ヘップの法則

- 神経が3つの場合はどうなるか。

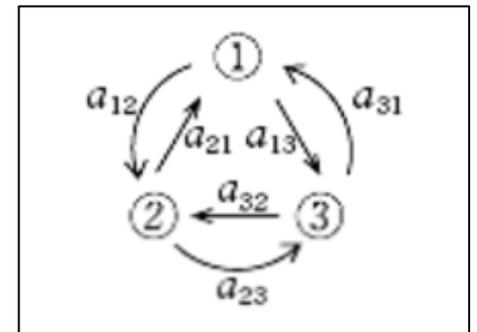


# ヘップの法則（神経3つでの脳内状態）

- 神経が3つの場合はどうなるか。
- $\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$  という3つの神経があった場合、脳内の状態は

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

と表される。



# ヘップの法則（神経3つでの学習例）

- 神経が3つの場合はどうなるか。

全てが活動状態で  $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$  ならば、

脳内の状態は、初期状態  $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$  から変化して、

$$\begin{pmatrix} 0 & 0 + 1 & 0 + 1 \\ 0 + 1 & 0 & 0 + 1 \\ 0 + 1 & 0 + 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

となる。

# ヘッブの法則 (hebb.py)

- 先週のプログラムの学習を再現する。
- 神経は3つだった。

```
... |> c:\code\admin\hebb\hebb.py
[1, 1, -1]を覚えました。
[1, 1, -1]を覚えました。
[1, 1, -1]を覚えました。
[1, 1, -1]から、[ 1  1 -1]を思い出しました！
[1, 1, 0]から、[ 1  1 -1]を思い出しました！
[1, -1, 1]を覚えました。
[1, -1, 1]を覚えました。
[1, 1, -1]から、[ 1  1 -1]を思い出しました！
[1, -1, 1]から、[-1 -1  1]を思い出しました！
```

# ヘップの法則(神経3つでの学習状態1)

- 1回目の学習(1,1,-1)

神経の活動状態は  $\begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$  なので、脳内の学習状態は、

$$\begin{pmatrix} 0 & 0+1 & 0-1 \\ 0+1 & 0 & 0-1 \\ 0-1 & 0-1 & 0 \end{pmatrix} \text{ であり、}$$

$$\begin{pmatrix} 0 & 1 & -1 \\ 1 & 0 & -1 \\ -1 & -1 & 0 \end{pmatrix}$$

となる。

# ヘップの法則(神経3つでの学習状態2)

- 2回目の学習(1,1,-1)

神経の活動状態は  $\begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$  なので、脳内の学習状態は、

$$\begin{pmatrix} 0 & 1+1 & -1-1 \\ 1+1 & 0 & -1-1 \\ -1-1 & -1-1 & 0 \end{pmatrix} \text{ であり、}$$

$$\begin{pmatrix} 0 & 2 & -2 \\ 2 & 0 & -2 \\ -2 & -2 & 0 \end{pmatrix}$$

となる。

# ヘップの法則(神経3つでの学習状態3)

- 3回目の学習(1,1,-1)

神経の活動状態は  $\begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$  なので、脳内の学習状態は、

$$\begin{pmatrix} 0 & 2+1 & -2-1 \\ 2+1 & 0 & -2-1 \\ -2-1 & -2-1 & 0 \end{pmatrix} \text{ であり、}$$

$$\begin{pmatrix} 0 & 3 & -3 \\ 3 & 0 & -3 \\ -3 & -3 & 0 \end{pmatrix}$$

となる。

# ヘップの法則(神経3つでの思い出し作業1)

- 現在の学習状態から(1,1,-1)を見るとどうなるか？  
**学習状態と、見せるものを掛け合わせる。**

$$\begin{pmatrix} 0 & 3 & -3 \\ 3 & 0 & -3 \\ -3 & -3 & 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 6 \\ 6 \\ -6 \end{pmatrix}$$

神経は1と-1でしか表さないため、計算して現れた数字が、正の数なら1、負の数なら-1とする。  
つまり、

$$\begin{pmatrix} 6 \\ 6 \\ -6 \end{pmatrix} \simeq \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$$

となり、今回は元のものを思い出せたという結果になる。

# ヘッブの法則(神経3つでの全体補完)

- 一部の情報からの推測:  
(1,1,-1)を思い出したいが、(1,1,?)と最後の部分だけわからない時、  
わからない部分を0とする。

$$\begin{pmatrix} 0 & 3 & -3 \\ 3 & 0 & -3 \\ -3 & -3 & 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \\ -6 \end{pmatrix}$$

となる。つまり、

$$\begin{pmatrix} 3 \\ 3 \\ -6 \end{pmatrix} \simeq \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$$

となり、一部の情報から全体を補完できた。

# ヘップの法則(神経3つでの学習状態4)

- 4回目の学習(1,-1,1)

神経の活動状態は  $\begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$  なので、脳内の学習状態は、

$$\begin{pmatrix} 0 & 3-1 & -3+1 \\ 3-1 & 0 & -3-1 \\ -3+1 & -3-1 & 0 \end{pmatrix} \text{ であり、}$$

$$\begin{pmatrix} 0 & 2 & -2 \\ 2 & 0 & -4 \\ -2 & -4 & 0 \end{pmatrix}$$

となる。

# ヘップの法則(神経3つでの学習状態5)

- 5回目の学習(1,-1,1)

神経の活動状態は  $\begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$  なので、脳内の学習状態は、

$$\begin{pmatrix} 0 & 2-1 & -2+1 \\ 2-1 & 0 & -4-1 \\ -2+1 & -4-1 & 0 \end{pmatrix} \text{ であり、}$$

$$\begin{pmatrix} 0 & 1 & -1 \\ 1 & 0 & -5 \\ -1 & -5 & 0 \end{pmatrix}$$

となる。

## ヘッブの法則(神経3つでの思い出し作業2)

- 現在の学習状態から(1,1,-1)を見るとどうなるか？

$$\begin{pmatrix} 0 & 1 & -1 \\ 1 & 0 & -5 \\ -1 & -5 & 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \\ -6 \end{pmatrix}$$

となる。つまり、

$$\begin{pmatrix} 2 \\ 6 \\ -6 \end{pmatrix} \simeq \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$$

となり、うまく思い出せた。

# ヘッブの法則(神経3つでの思い出し作業3)

- 現在の学習状態から(1,-1,1)を見るとどうなるか？

$$\begin{pmatrix} 0 & 1 & -1 \\ 1 & 0 & -5 \\ -1 & -5 & 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} -2 \\ -4 \\ 4 \end{pmatrix}$$

となる。つまり、

$$\begin{pmatrix} -2 \\ -4 \\ 4 \end{pmatrix} \simeq \begin{pmatrix} -1 \\ -1 \\ 1 \end{pmatrix}$$

となり、うまく思い出せなかった。

# ヘッブの法則 (hebb.py)

- サンプルプログラムのコードを読んで、理解する。
- 28行目に `print (self.memory)` というコードを入れると、学習の様子が段階ごとにわかる。

```
18     for y in range(self.memory_size):
19         for x in range(self.memory_size):
20             if x == y:
21                 self.memory[x,y] = 0
22             elif neural_activity[x] == 1 and neural_acti
23                 self.memory[x,y] += 1
24             elif neural_activity[x] == -1 and neural_act
25                 pass
26             else:
27                 self.memory[x,y] -= 1
28     print (self.memory)
29     return True
30
```

```
[[ 0  1 -1]
 [ 1  0 -1]
 [-1 -1  0]]
[1, 1, -1]を覚えました。
[[ 0  2 -2]
 [ 2  0 -2]
 [-2 -2  0]]
[1, 1, -1]を覚えました。
[[ 0  3 -3]
 [ 3  0 -3]
 [-3 -3  0]]
[1, 1, -1]を覚えました。
[1, 1, -1]から、[ 1  1 -1]を思い出しました！
[1, 1, 0]から、[ 1  1 -1]を思い出しました！
[[ 0  2 -2]
 [ 2  0 -4]
 [-2 -4  0]]
[1, -1, 1]を覚えました。
[[ 0  1 -1]
 [ 1  0 -5]
 [-1 -5  0]]
[1, -1, 1]を覚えました。
[1, 1, -1]から、[ 1  1 -1]を思い出しました！
[1, -1, 1]から、[-1 -1  1]を思い出しました！
```

# 演習

## 1. Read

[http://gaya.jp/spiking\\_neuron/matrix.htm](http://gaya.jp/spiking_neuron/matrix.htm)

<http://www.cscd.Osaka-u.ac.jp/user/rosaldo/110505PezBrain.html>

## 2. Understand source code

<https://www.python.org/>

<https://docs.python.org/3/tutorial/index.html>

わからない所があれば、調べてください。

# 次回

- ヘッブの法則を応用する作業を行う予定。