しおん研-ゼロつくゼミ

第２回

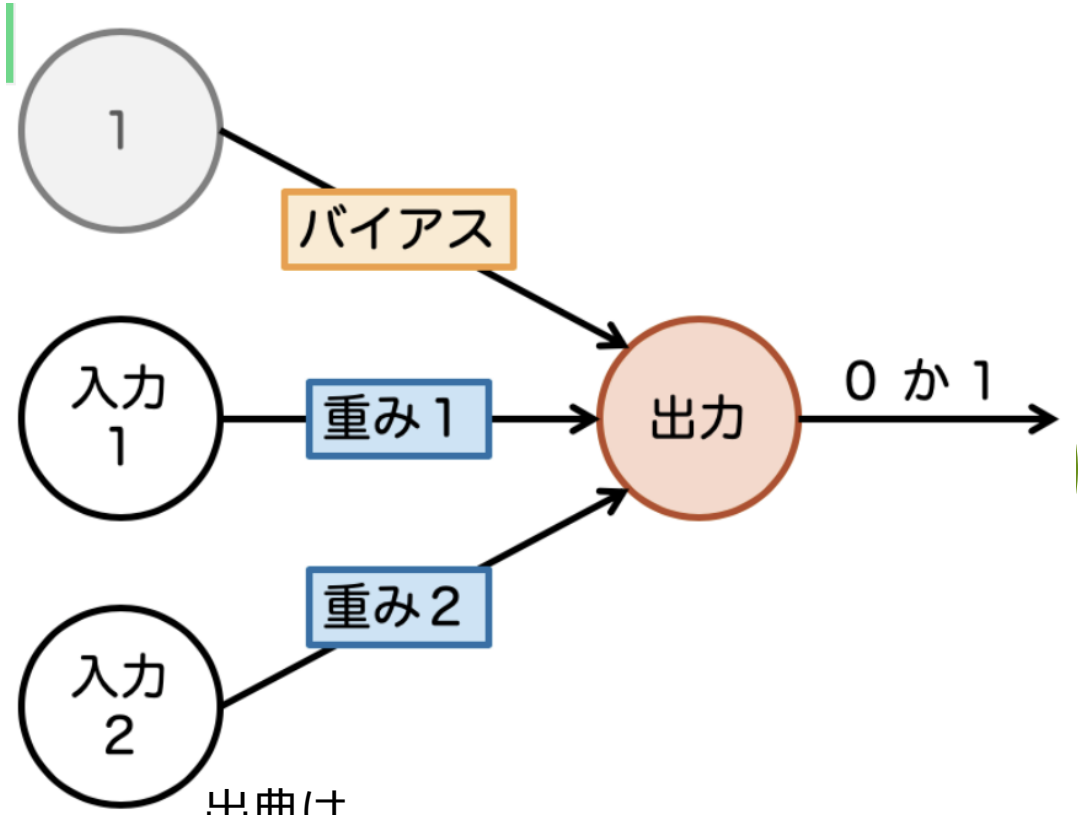
前回までにやったこと

- Deep Learningの概観
- Pythonの環境と基本文法

今回やること（参考書P.21~P.65）

- ニューラルネットワークの起源である、パーセプトロンの構造とその限界を確認する
- パーセプトロンからニューラルネットワークへの数学的拡張
- ３層ニューラルネットワーク（順伝播のみ）の実装

パーセプトロンとは：下図のようなアルゴリズム！



出典は
https://blog.apar.jp/deep-learning/11979/

入力１：x1
入力２：x2
出力：y
バイアス：b
重み１：wi
重み２：w2

と表現する。

入力１、２と出力は各々１または０を取ることとする。
（すなわち、バイアスのノードは常に１をとる入力信号であると言える）

パーセプトロンの数式的表現

$$\vec{x} = (x_1, x_2, \ldots, x_n)$$

$$f(\vec{x}) = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + w_{n+1}$$

$$f(\vec{x}) > 0 \rightarrow 1$$
$$f(\vec{x}) < 0 \rightarrow 0$$
$$f(\vec{x}) = 0 \rightarrow 1 \text{ or } 0$$

出典：
https://newtechnologylifestyle.net/一番簡単な単純パーセプトロンについて/

重み：出力に対して、当該の入力ノードがどれだけ重要か
バイアス：信号の通りやすさ（どれだけ信号の重みつき総和に下駄を履かせるか）

を意味する！

パーセプトロンを用いた単純な論理回路の実装

- AND gate（(x1,x2) = (1,1) のときのみ y=1）
- NAND gate（ANDの逆）
- OR gate（(x1,x2)の一方に1があれば y = 1）

それぞれ、パラメータは何にすれば良い？

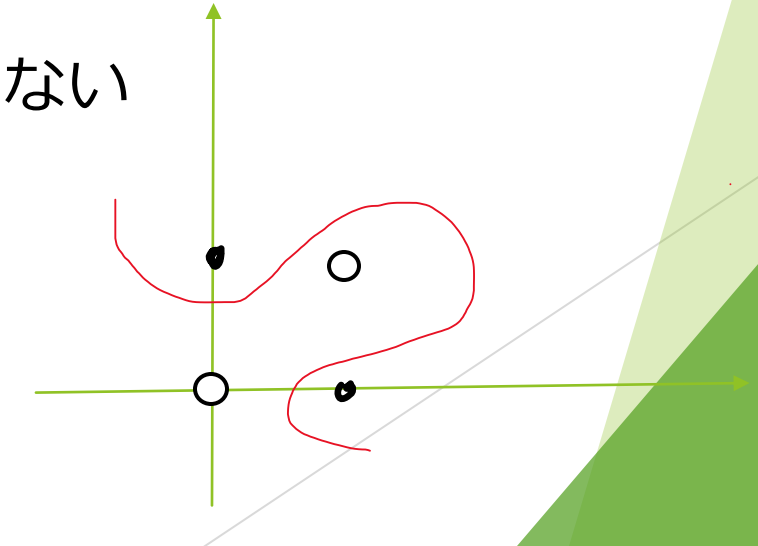
パーセプトロンでXOR回路（排他的論理和）が表せる？：（単層）パーセプトロンの限界

- XOR回路　とは

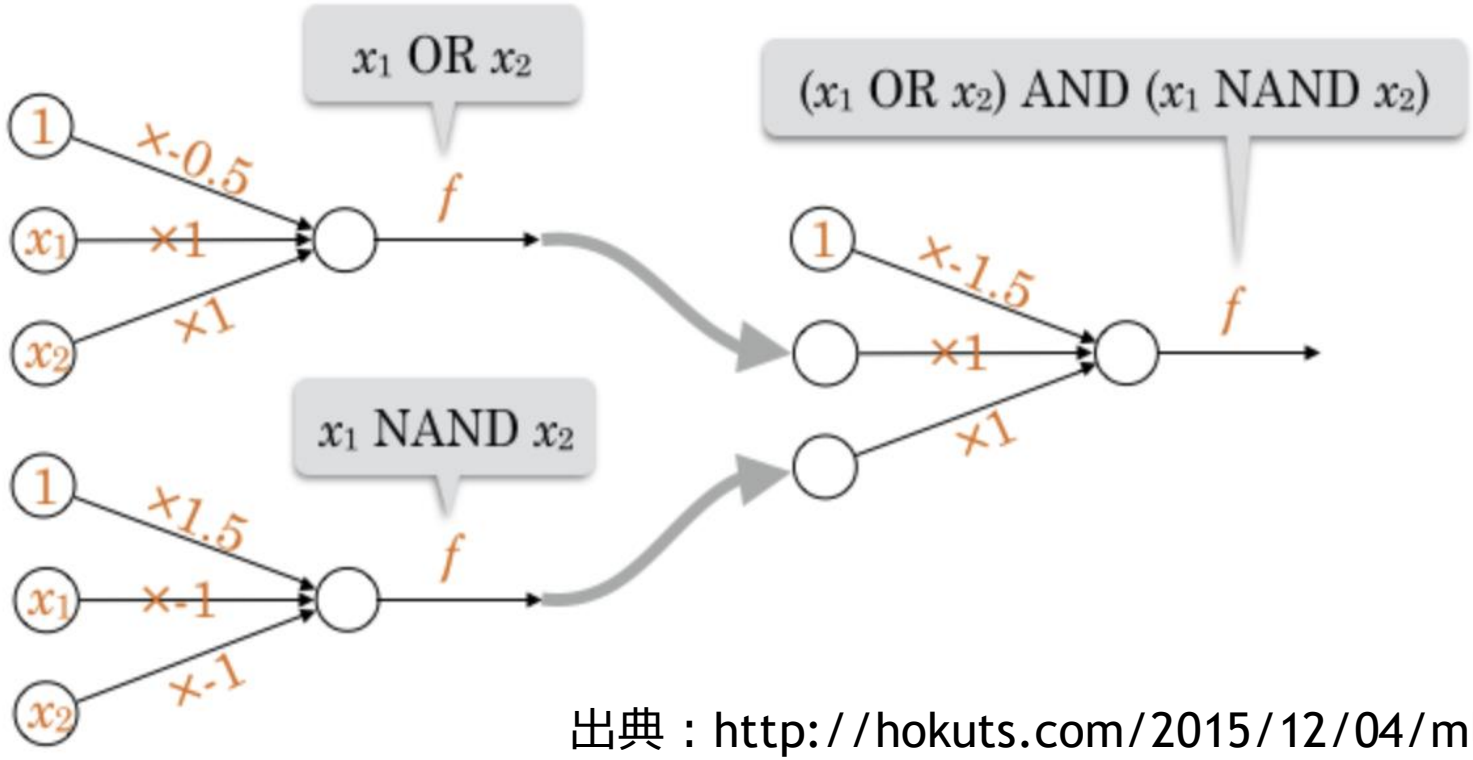
x1	x2	y
1	1	0
1	0	1
0	1	1
0	0	0

これは、１層のパーセプトロンでは表せない
（理由：境界が非線形だから！）

単層パーセプトロンでは、境界が線形のものしか扱えないという意味で限界がある。



XORをパーセプトロンの構造を使って表すには？→層を増やす！



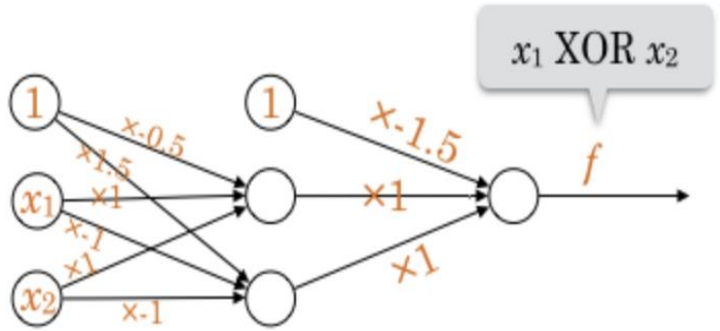
出典：http://hokuts.com/2015/12/04/ml3-mlp/

本当にそうなっているのか？

- 中間層のノードをs1,s2とすると
  うまくいってそう！
  （実装はGoogle Colaboratoryで
  行います）

つまり、層を増やすことによって、
非線形なモデルの表現も可能になる！

x1	x2	s1	s2	y
1	1	1	0	0
1	0	1	1	1
0	1	1	1	1
0	0	0	1	0



(c.f.) コンピューターは、NAND gateの組み合わせによって信号を伝達しているそうです。

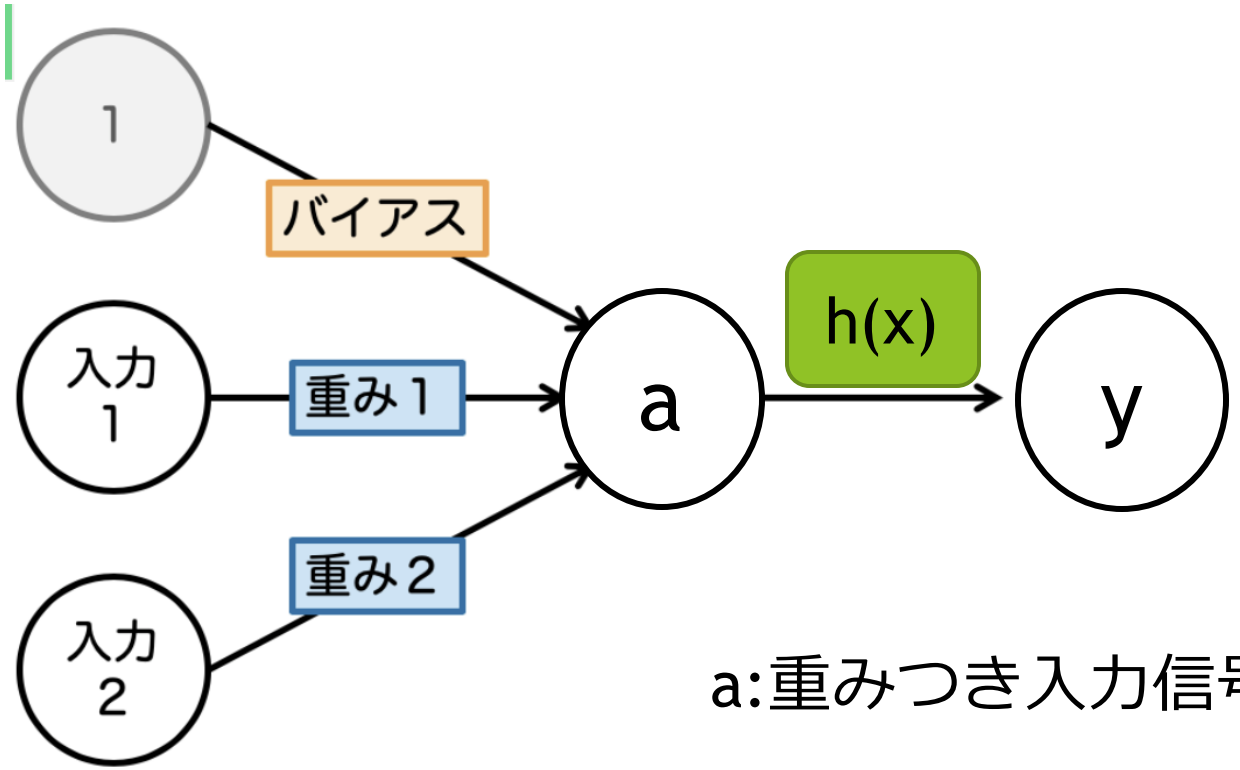
じゃあ、これだけで十分？

- 層が深くなってくると、パラメータ（重み、バイアス）の数も多くなる・・・
- それに、関数が複雑になればなるほど、適切なパラメータを人の手で決定するのは難しい・・・

→ 機械自身で適切なパラメータを決めてもらおう！（＝Deep Learning）

そのような構造を持つモデルを構築するためには、パーセプトロンをどのように拡張させたらいいんだろう？

パーセプトロンをもっと細かく分解してみる

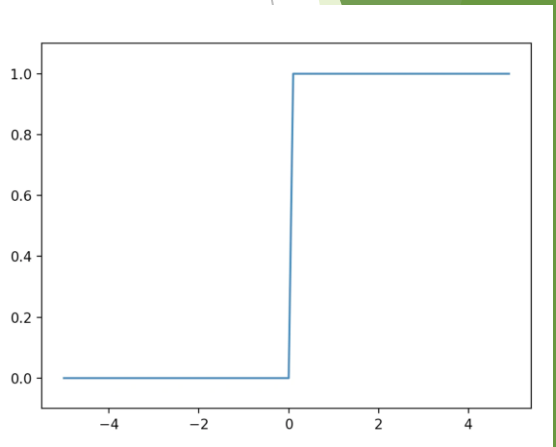


h(x) = 0 (x <= 0)
= 1 (x > 0)

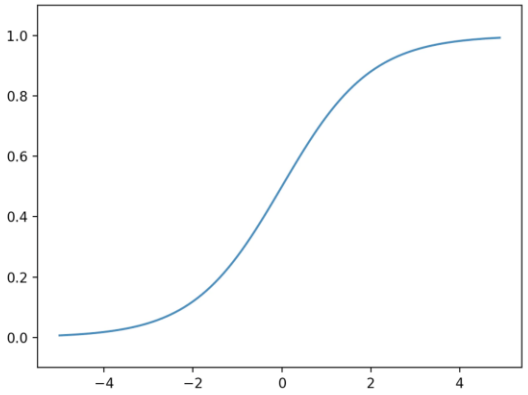
a:重みつき入力信号の総和

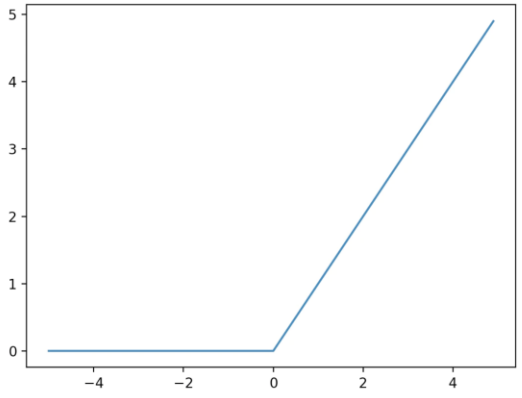
h(x)：活性化関数
（実装はGoogle Colabで）

- 活性化関数：どのように入力信号の総和を発火（=実行）させるかを決める関数
- パーセプトロンにおける活性化関数はStep関数といい、右のような不連続関数をいう。



- ニューラルネットワークでよく使う活性化関数は、Sigmoid（左）やRelu（右）といった連続関数である。



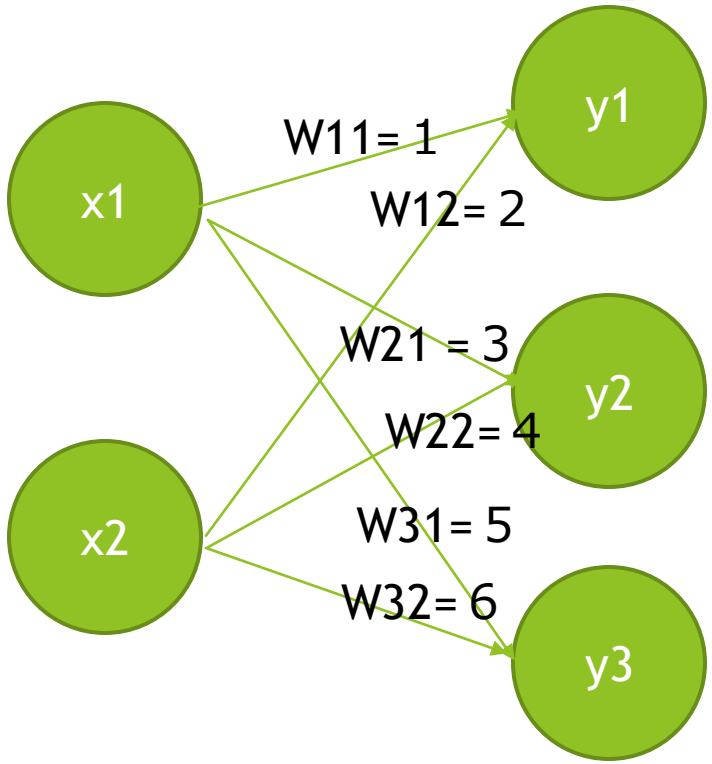


出典：
https://qiita.com/namitop/items/d3d5091c7d0ab669195f

ニューラルネットワークの実装

- 多次元配列の計算
- 実際の入力ノード、出力ノードは２個、１個よりもだいぶ多い

→１回の「重みつき入力信号の総和」の計算を行列でまとめて行いたい！！



$$[x1,x2] \times \begin{bmatrix}[1,3,5] \\ [2,4,6]\end{bmatrix} = [y1,y2,y3]$$

ニューラルネットワークの実装

- 入力のノードのベクトルをX = [x1,x2,……,xn1]とする
- 重みつき入力信号の総和をA= [a1,,a2,……,an2]とする
- 重みパラメータをW = [[w11,w21,……..,wn21]

  [w12,w22,……..,wn22]

  [w1n1,w2n,……..,wn2n1]]　とする

  バイアスを　B = [b1,b2,…….,bn2]とする

- 出力ノードのをY = [y1,y2,……..,yn2]とすると

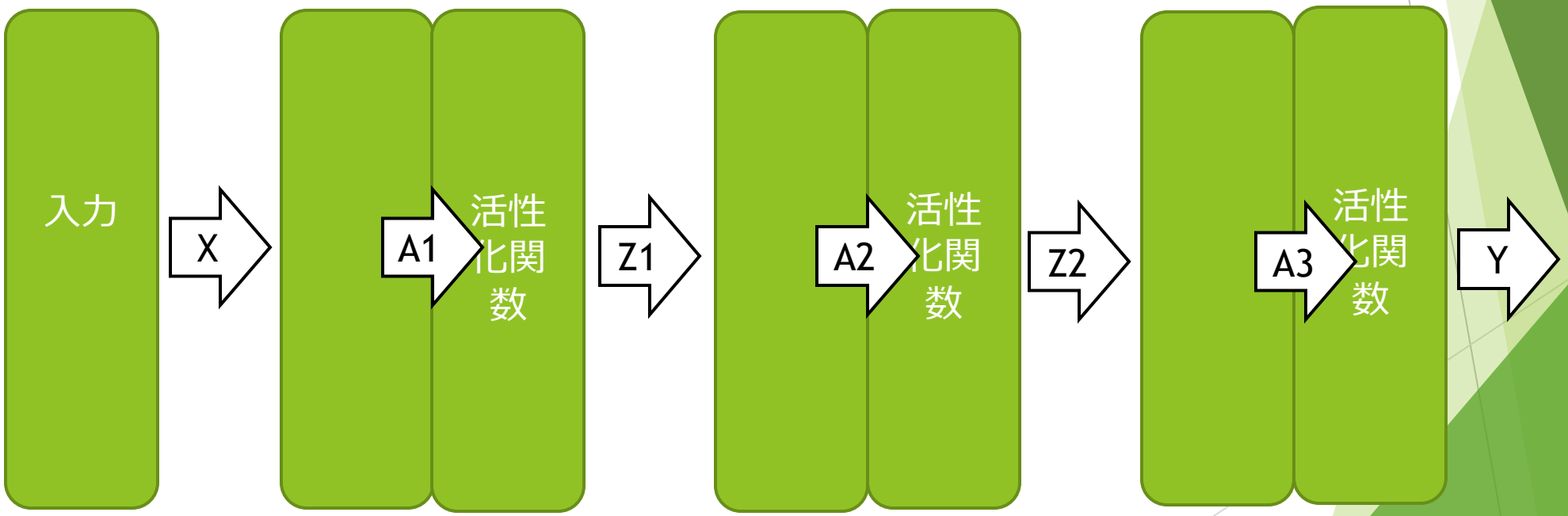
- ニューラルネットワーク１層分：

  $z = XW + B$ (A=XW)

  $Y = h(Z)$

３層のニューラルネットワークを実装してみよう

- （注意）ニューラルネットワークの層を数える時は、入力層を省くことが多い。



（おまけ）どうしてニューラルネットワークではStep関数を使わないのか？