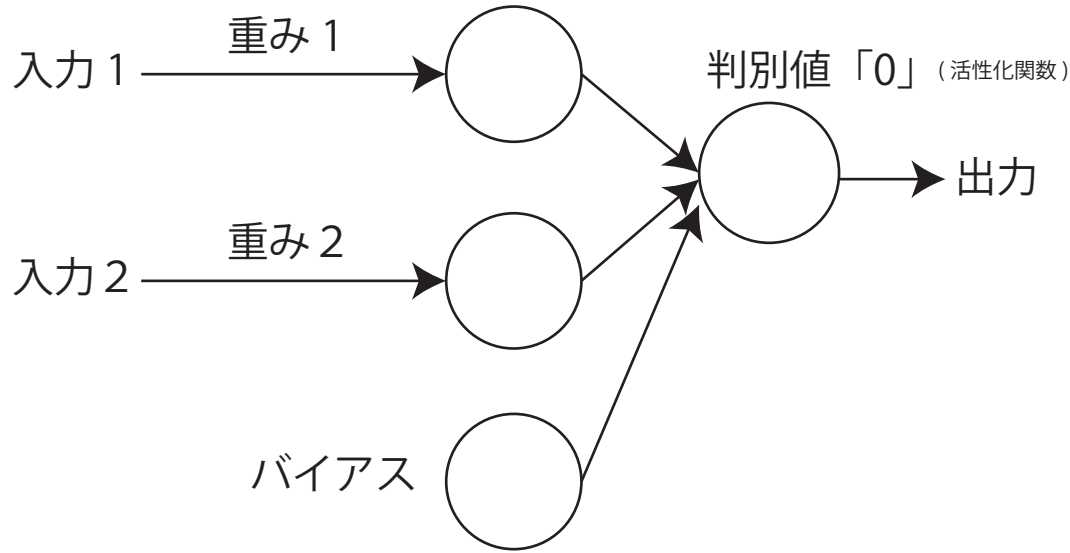


紙と鉛筆とサイコロで学ぶ単層パーセプトロン (バイアス版)

バイアス版は初期値によっては繰り返しが少し大変 (人にとっては)
取りあえずサイコロを振らずに [3, 4, 1] から始めるといいかも
epoch = 2 くらいで収束する。 [2, 2, -3]

(Deep Learning(AI) プログラミング基礎技術の体験)



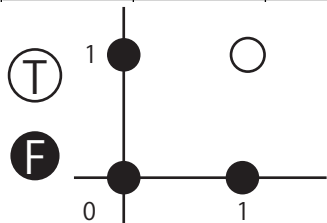
◆手順◆

- サイコロで重みとバイアスを決める (初期値)
- 入力 1 と入力 2 の入力パターンを決める (回路ごとの表を見て)
- 入力と重みを乗じた値とバイアスの合計が「0」を超えていれば 1 を出力, そうでなければ 0 を出力
- 出力が誤っている場合は次の操作をする。
A: 出力が 0 で正解が 1 の時 (合計が小さい)
バイアスと入力が 1 だった重みに 1 を足す
B: 出力が 1 で正解が 0 の時 (合計が大きい)
バイアスと入力が 1 だった重みから 1 を減じる
→コードで言えば学習係数が 1 ということ
- 3)4) を繰り返す
- 全入力パターンに正解するまで 2)-5) を繰り返す
- 完成したら重みとバイアスをメモし違う初期値で試す
(重みと判別値が一意に決まらない≡AI の個性の seed ! ?)
- 他も作る
NAND は AND の逆転
単層では作れないものもある。プロット参考+複層で作ってみる

作る回路の整理と単層では作れないもの探し (線形分離できないもの探し)

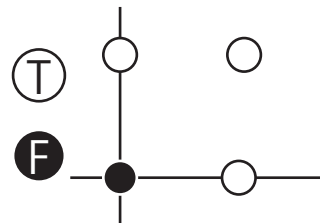
AND 回路

入力 1	入力 2	正解
0	0	0(F)
0	1	0
1	0	0
1	1	1(T)



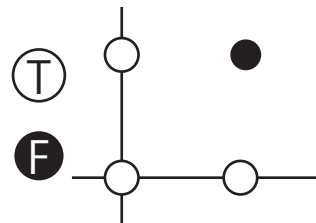
OR 回路

入力 1	入力 2	正解
0	0	0
0	1	1
1	0	1
1	1	1



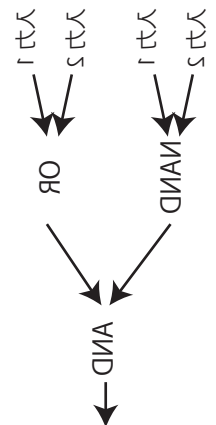
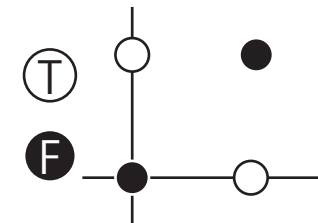
NAND 回路

入力 1	入力 2	正解
0	0	1
0	1	1
1	0	1
1	1	0



XOR 回路

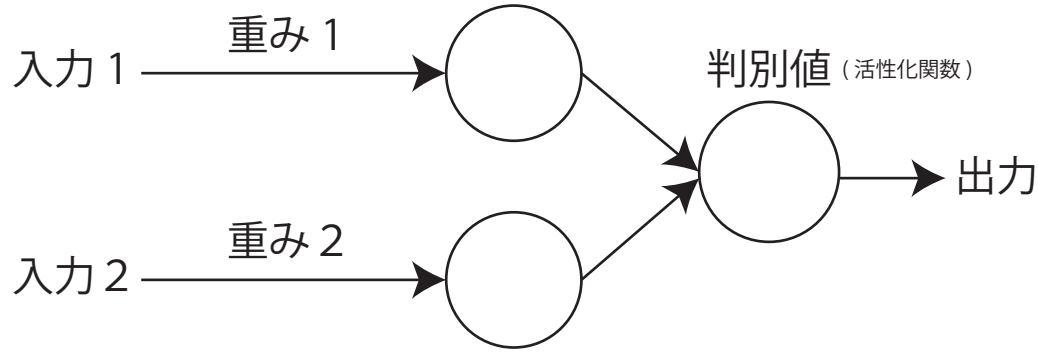
入力 1	入力 2	正解
0	0	0
0	1	1
1	0	1
1	1	0



そのうち重みとバイアスのルールが分かるようになるのも面白いところ> 深層学習をするとその人自身がドメインに詳しくなる?

紙と鉛筆とサイコロで学ぶ単層パーセプトロン (少し早く収束する版)

(Deep Learning(AI) プログラミング基礎技術の体験)



◆手順◆

- 1) サイコロで重みと判別値を決める (初期値)
- 2) 入力1と入力2の入力パターンを決める (回路ごとの表を見て)
- 3) 重みを乗じた合計が判別値を超えていれば1を出力, そうでなければ0を出力
- 4) 出力が誤っている場合は次の操作をする。
 A: 出力が0で正解が1の時 (出力が小さい (判別が大きい))
 判別値の値を1引いて, 入力が1だった重みに1足す
 B: 出力が1で正解が0の時 (出力が大きい (判別が小さい))
 重みのうち入力が1だった方の数字を1引いて, 判別値に1足す
- 5) 3)4) を繰り返す
- 6) 全入力パターンに正解するまで 2)-5) を繰り返す
- 7) 完成したら重みと判別値をメモし違う初期値で試す
 (重みと判別値が一意に決まらない≡AIの個性のseed!?)
- 8) 他も作る

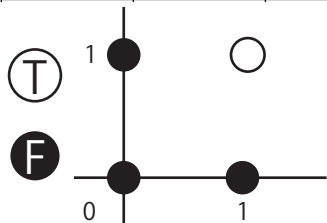
NANDはANDの逆転

単層では作れないものもある。プロット参考+複層で作ってみる

作る回路の整理と単層では作れないもの探し (線形分離できないもの探し)

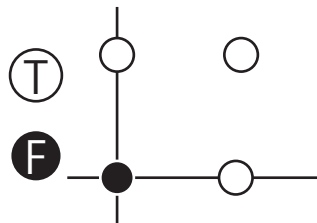
AND 回路

入力1	入力2	正解
0	0	0(F)
0	1	0
1	0	0
1	1	1(T)



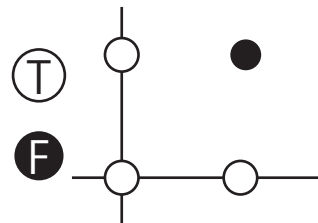
OR 回路

入力1	入力2	正解
0	0	0
0	1	1
1	0	1
1	1	1



NAND 回路

入力1	入力2	正解
0	0	1
0	1	1
1	0	1
1	1	0



XOR 回路

入力1	入力2	正解
0	0	0
0	1	1
1	0	1
1	1	0

