ニューラルネットワークの歴史と手法1

パーセプトロンまで

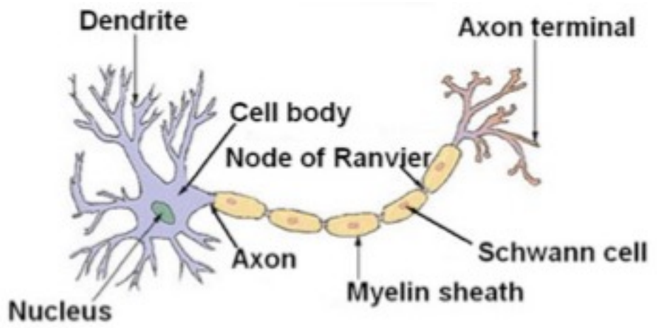
公立小松大学
藤田 一寿

かなりマニアックな部分があります.
必ず元論文をチェックしましょう！！

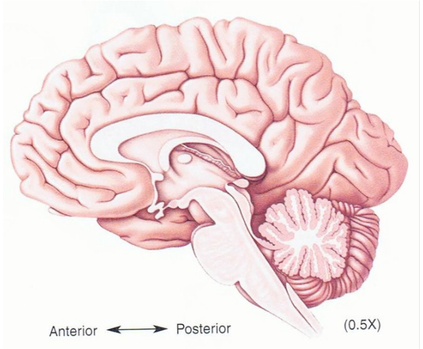
人工ニューラルネットワークのよくある説明

- 人工ニューラルネットワークは脳の構造・機能の模倣から発展した.

神経科学



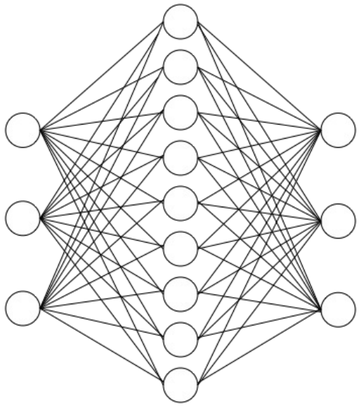
神経細胞



脳

人工ニューラルネットワーク

ネットワーク化



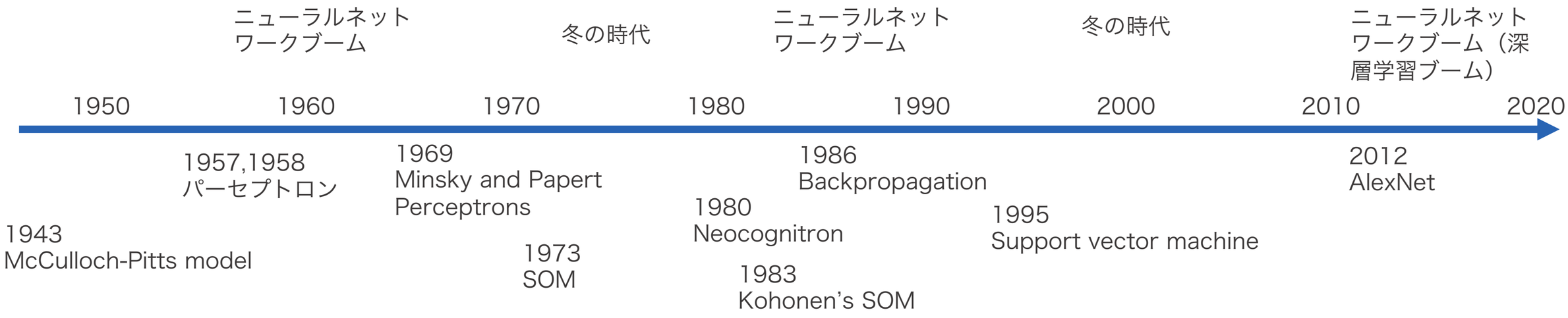
人工ニューロン
ニューロンモデル

人工ニューラルネットワーク

人工ニューロンを組み合わせることで，様々な機能を実現できる.

よく言われるニューラルネットワークの歴史

- 1943年McCullochとPittsによりニューロンモデルが提案される.
- 1957年Rosenblattがパーセプトロン（ニューラルネットワーク）を発表する.
- 1969年MinskyとPapertがパーセプトロンが線形分離不可能な問題が解けないことを示し，ニューラルネットワークの研究が下火になる.
- 1986年Backpropagationによりパーセプトロンの多層化が容易にできるようになり，線形分離不可能な問題を解けるようになる.
- 性能の頭打ち，サポートベクターマシンなどの他の手法の発展によりニューラルネットワークの研究が下火になる.
- 2012年深層ニューラルネットワークが画像識別のコンペで優勝し深層ニューラルネットワークが注目される.



ニューラルネットワークの歴史

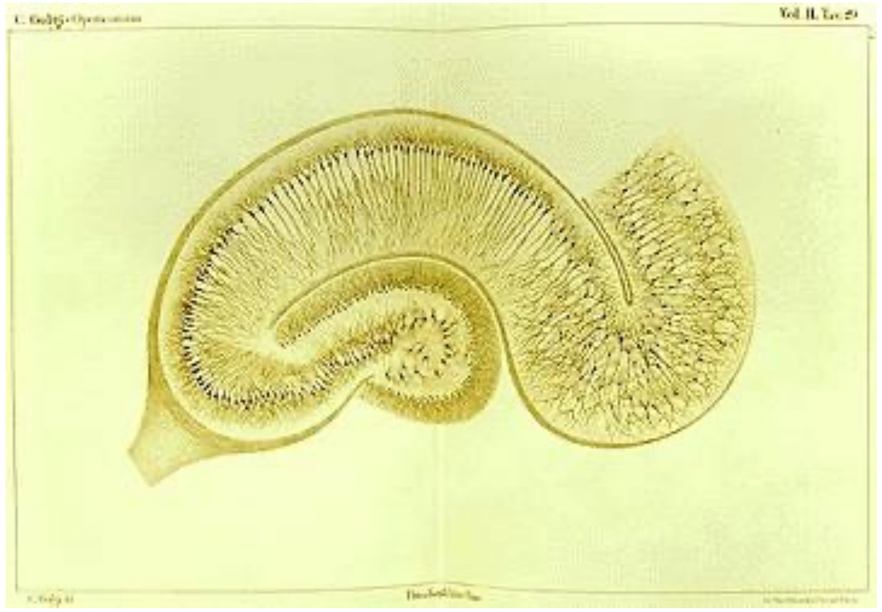
- 1873年 ゴルジ染色
- 1987年 Neuron doctrine
- 1906年 Receptive field
- 1914年 All or none law
- 1946年 McCulloch-Pitts model
- 1949年 Hebbian learning
- 1957年 Perceptron
- 1968年 Primary visual cortex
- 1969年 Rectified linear
- 1973年 Self organizing map
- 1980年 Neocognitron
- 1982年 Hopfield model (H
- 1986年 Backpropagation
- 1988年 Autoencoder
- 1989年 LeNet
- 1997年 LSTM
- 2002年 Liquid state machine
- 2006年 Deep belief network
- 2012年 Dropout
- 2012年 Alexnet
- 2015年 ResNet

詳しくはQiitaをみてね．https://qiita.com/KazuhisaFujita/items/b40f2363558c373bde55

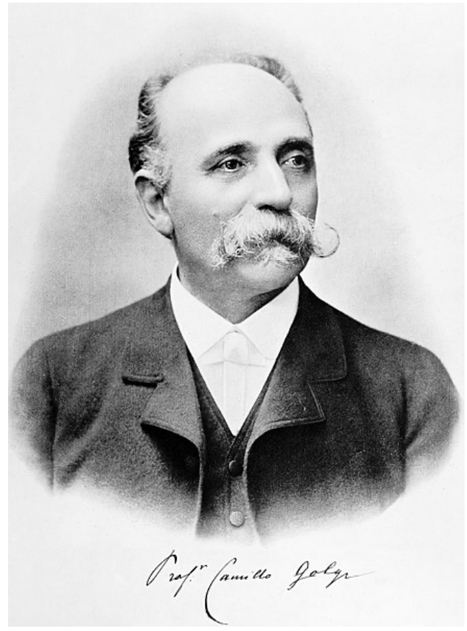
神経科学の萌芽

ゴルジ染色（黒い反応） 1873年

- ゴルジが発明した神経細胞の染色方法
- この染色法により脳の内部構造の理解が深まった.



海馬のスケッチ(Golgi, 1886)



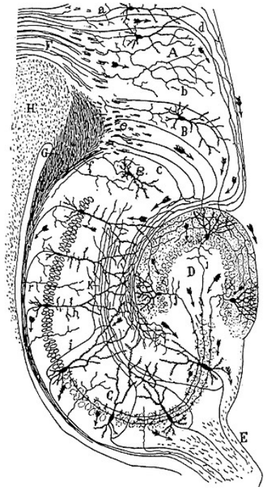
元の文献をチェックしていない.

ニューロン 1891年，シナプス 1897

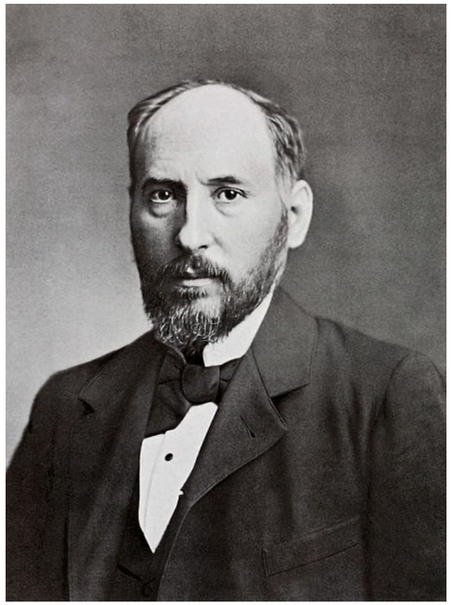
- Waldeyere-Heartzが独立した神経単位をニューロン，ニューロンとニューロンのつなぎ目をシナプスと命名した.

Neuron doctrine（ニューロン説）(1880s,1890s年)

- Cajal（カハール）らが提唱した脳の構造の考え方.
  - 脳は独立したニューロンからなる.
  - シナプス間隙が電子顕微鏡により確認され，Neuron doctrineが正しいことが分かった (Birks et al., 1960).
- 一方ゴルジはReticular theory（網状説）を提唱した.
  - 脳全体が合胞体(syncytium)であり、共通の細胞質を持つ連続した組織の塊である (Gazzaniga MS et al., 2013).



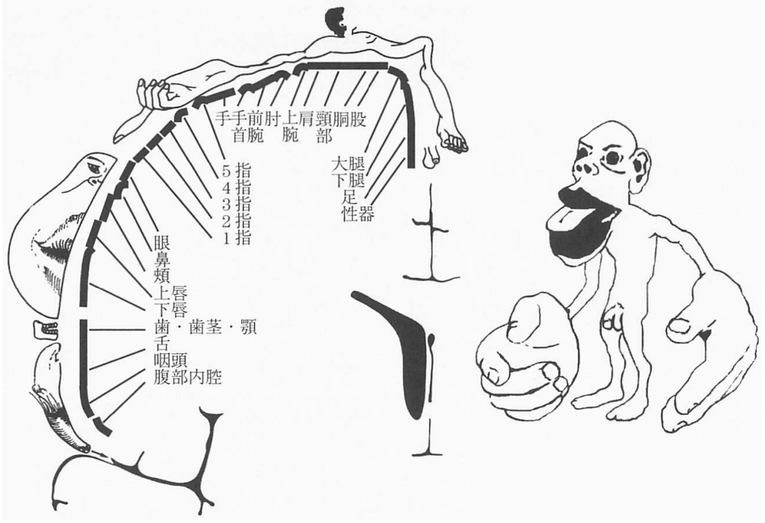
Cajalによる海馬のスケッチ



元の文献をチェックしていない.

Receptive field（受容野）(1906年)

- 一つの神経細胞が受け持つ領域のこと.
- この考え方が畳み込みニューラルネットワークニューラルネットワークの畳み込みに対応する.
  - 脳の視覚処理のRetinotopic mapに対応.
  - 畳み込みニューラルネットワークにも第1次視覚野に見られるガボールフィルタ様の受容野が現れる.



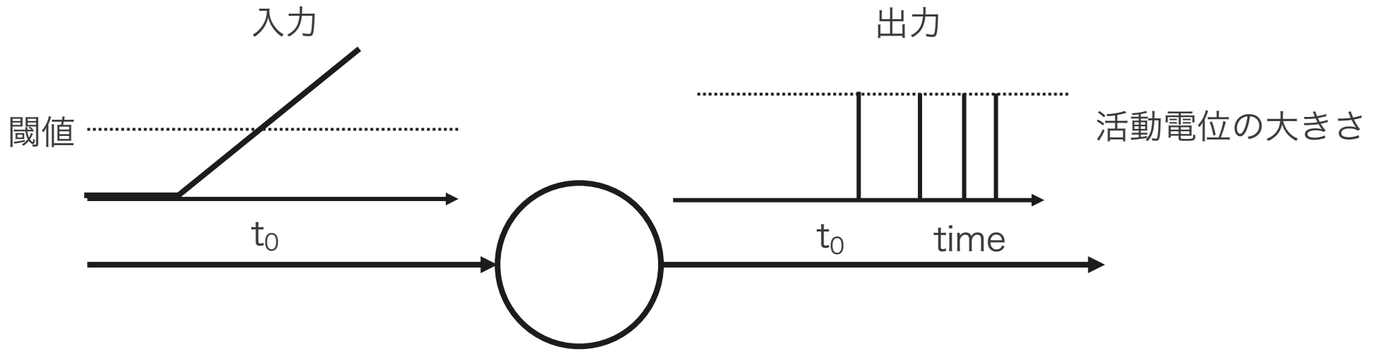
(内川恵二編，聴覚・触覚・前庭感覚)
Penfield and Rasmussen, 1950の文献の図の改変

例
体性感覚野では，体表面の特定の場所に対応したニューロンがある(Somatotopic map).
ニューロンの位置は体の場所と対応している.

Sherrington, C S (1906); Hartline (1938)
元の文献を詳しくチェックしていない

All or none law（全か無かの法則）(1914年)

- 神経細胞の応答の基本的なルール
  - 膜電位が閾値を越えると活動電位を発する（発火する）.
    - 簡単に言うと，神経細胞への入力が一定の値（閾値）より大きいと1を出力する.
  - 入力の大きさによらず，活動電位の大きさは一定である.
    - 簡単に言うと，入力の大きさによらず，神経細胞の出力は1で一定である.



入力が閾値をこえると活動電位を発する.
活動電位の大きさは一定である.

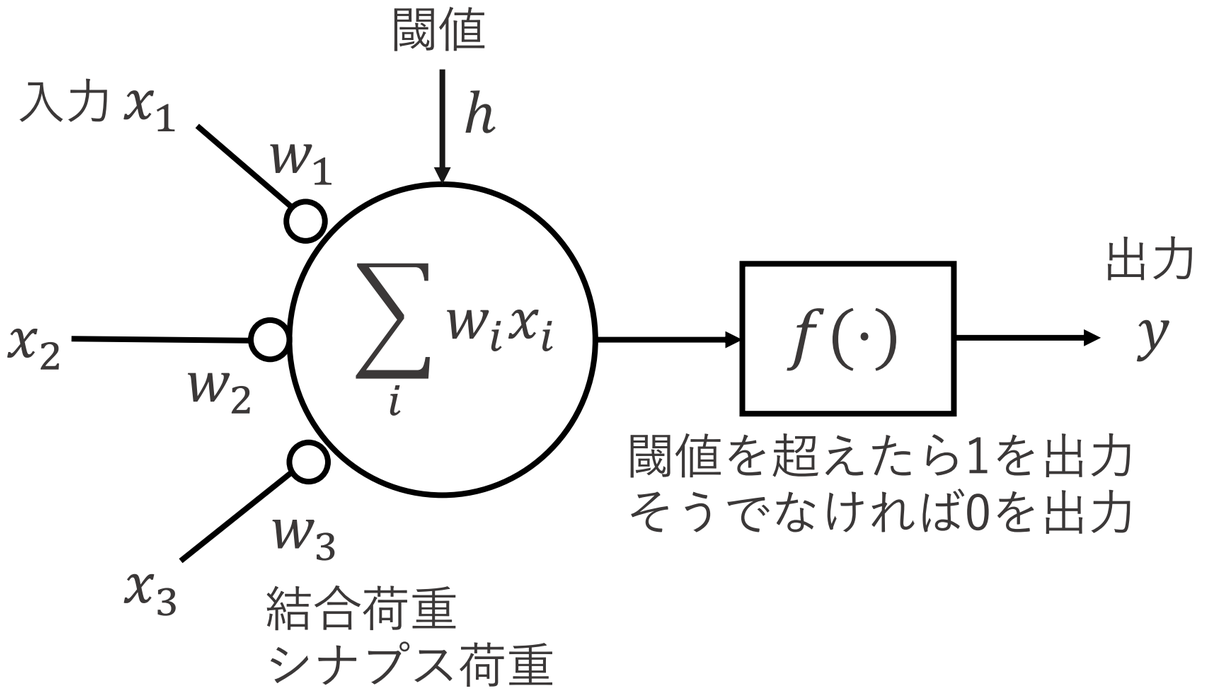
実際のニューロンはそこまで単純ではない.

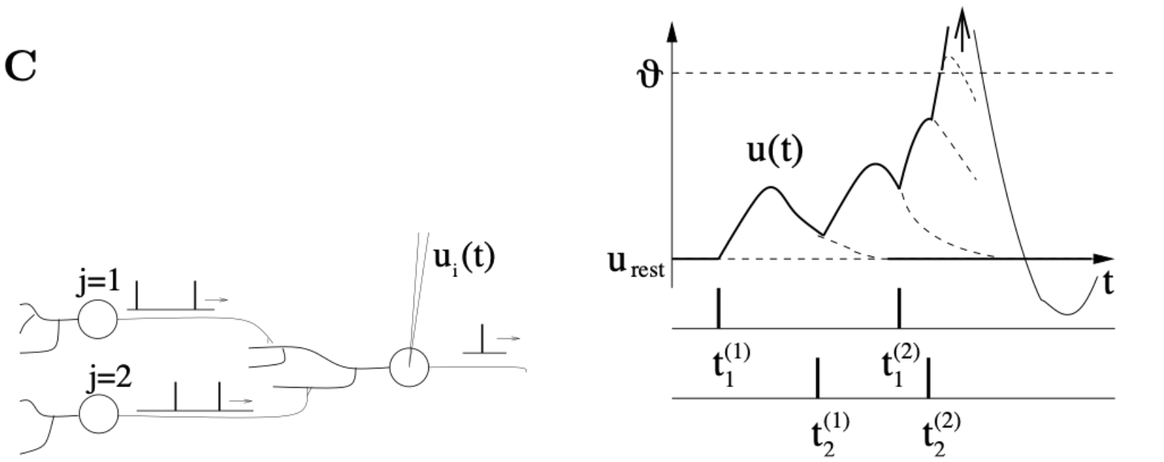
Adrian E. D. (1914). The all-or-none principle in nerve. *The Journal of physiology*, *47*(6), 460–474.
元の文献を詳しくチェックしていない.

神経科学の理論的研究の始まり

線形閾値素子

- 神経細胞の数理モデル
- 入力x_iが神経細胞に伝わる強さ(接続の強度)をシナプス荷重w_iと言う.
- 入力とシナプス荷重をかけたものの総和が閾値hを超えたら1を出力.
  - PSP (Post-synaptic potential)の総和が閾値を超えるとスパイクが発生する現象をモデル化.
  - All or None law (全か無かの法則)



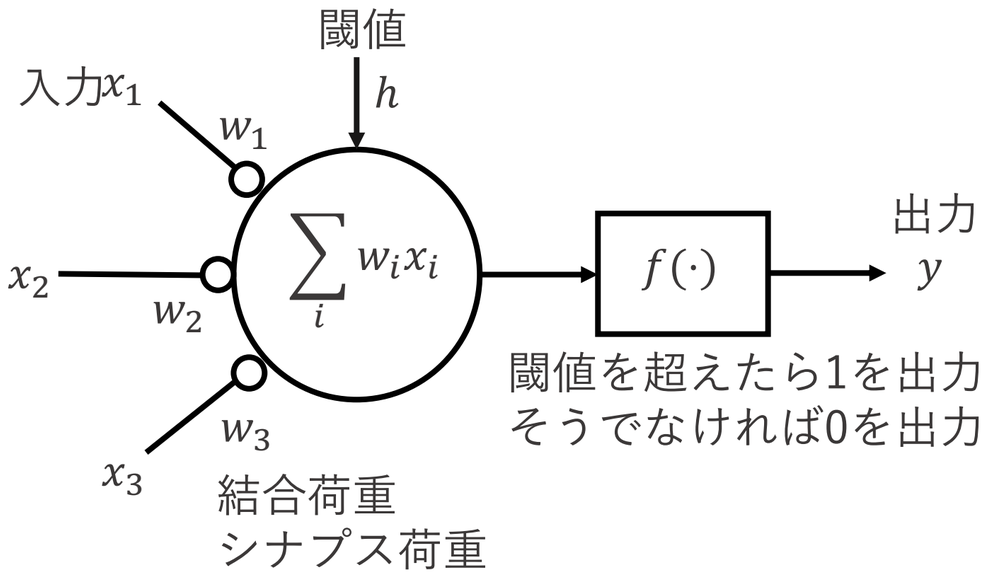


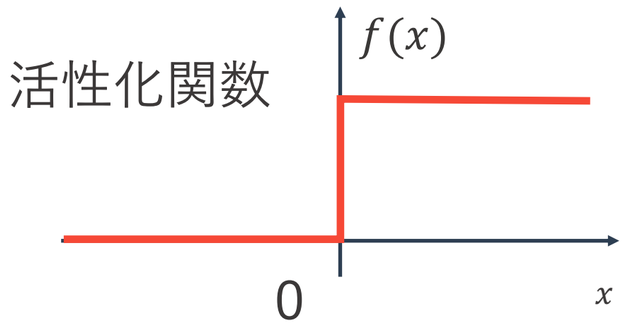
(Gerstner and Kistler, Spiking neuron models)

現代人の立場からMcCulloch-Pittsのニューロンモデルを書き直したものと言えるだろう．特にこのスライドの記述は生物よりに書いている.

線形閾値素子の数式表現

- 入力を x_i，重みを w_iとするとニューロンの出力 y は次のように書ける．
- $y = f(\sum_i w_i x_i - h)$
- $f(\cdot)$は活性化関数，hは閾値である．
- $f(\cdot)$はステップ関数である．
- $f(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$





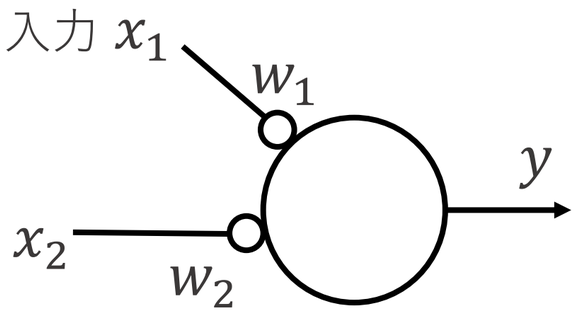
AND演算を実現する

- AND演算を実現するために図のような2入力1出力のネットワークを考える.
- $w_1 = w_2 = 1, h = 1.5$とすると，入力と出力の関係は次のように書ける.
- $y = f(x_1 + x_2 - 1.5)$
- この式はAND演算を実現している.
- 重みや閾値を人間の決め打ちではなく自動で決めたい！！

AND演算

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

ネットワーク



ネットワークの各数値

x_1	x_2	$x_1 + x_2 - 1.5$	y
0	0	-1.5	0
0	1	-0.5	0
1	0	-0.5	0
1	1	0.5	1

McCulloch-Pitts ~~neuron~~ neural network model (1946年)

- McCullochとPittsは1946年の論文でニューラルネットワークで様々な論理演算ができるといっている.

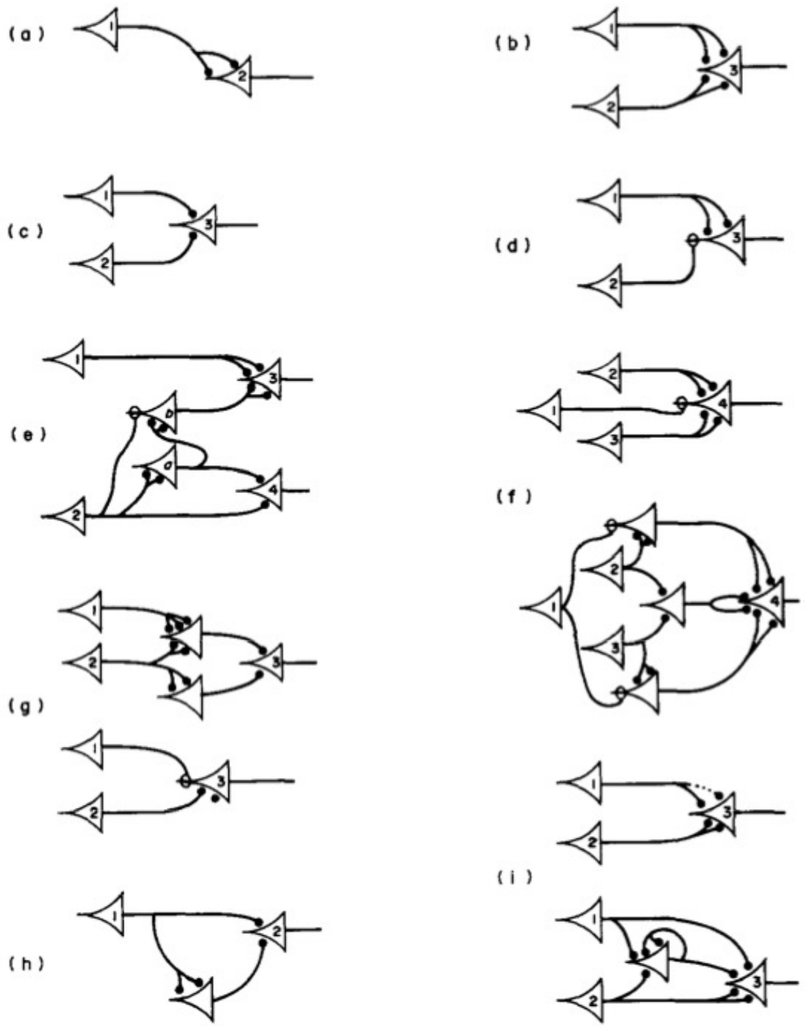
- **閾値論理素子の提案**
  - 神経細胞は論理計算を行える.
  - McCulloch-Pittsのneuron model
- **ニューラルネットワークの提案**
  - 様々な論理演算をするネットワークを構築することができる.

- McCullochとPittsの論文に記載されている前提条件
  - ニューロンの活動は全か無かの法則に従う.
  - ニューロンを興奮させるためには、ある期間内に一定の数のシナプスが興奮していなければならない. この数は以前の活動やニューロン上の位置に依存しない.
  - 神経系の中で唯一の重要な遅延はシナプスの遅延である.
  - 抑制性シナプスの活動は, その時点でのニューロンの興奮を絶対に阻止する.
  - ネットの構造は時間とともに変化しない.

(McCulloch and Pitts 1946)
分かりにくい論文

McCulloch-Pitts neural network model

McCullochとPittsによる様々なネットワーク.
これらはすべて対応する論理計算が存在する.

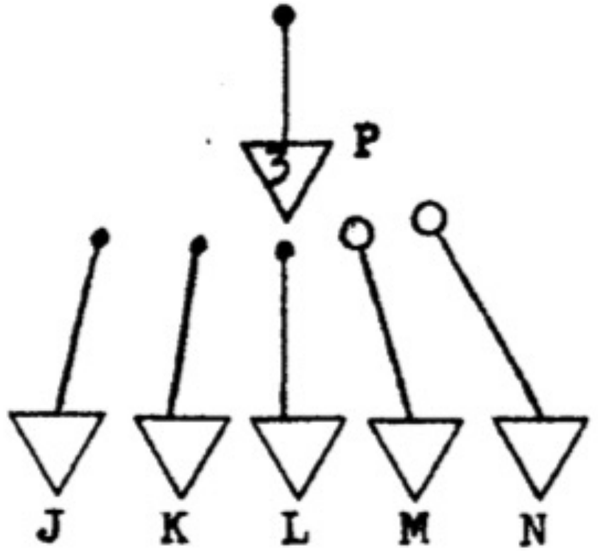


(McCulloch and Pitts, 1946)

Kleene によるニューラルネットワークによる論理計算の例.
黒丸は興奮，白丸は抑制性結合を表す.
P三角の中の数字は閾値を表す.
つまり，J, K, Lが発火し，M, Nが発火しなければPは発火するネットワークである.
これを論理式になおすと

$$P = J \cdot K \cdot L \cdot \overline{M} \cdot \overline{N}$$

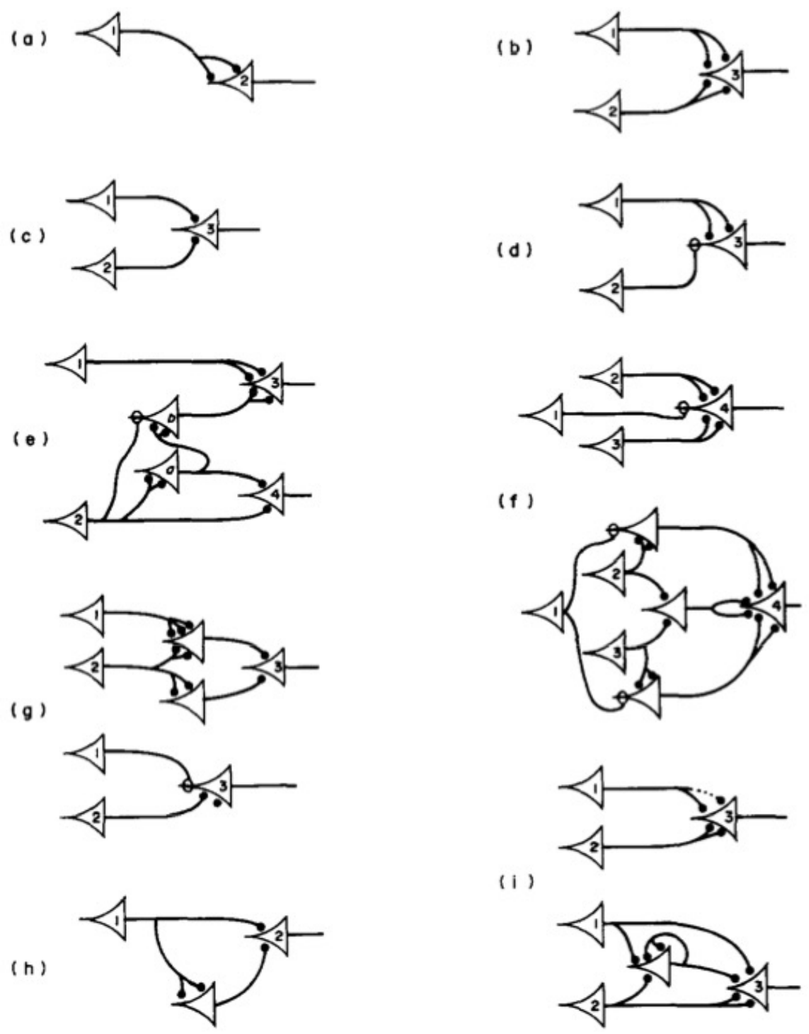
となる.



(Kleene, 1951)

McCulloch-Pitts neural network model

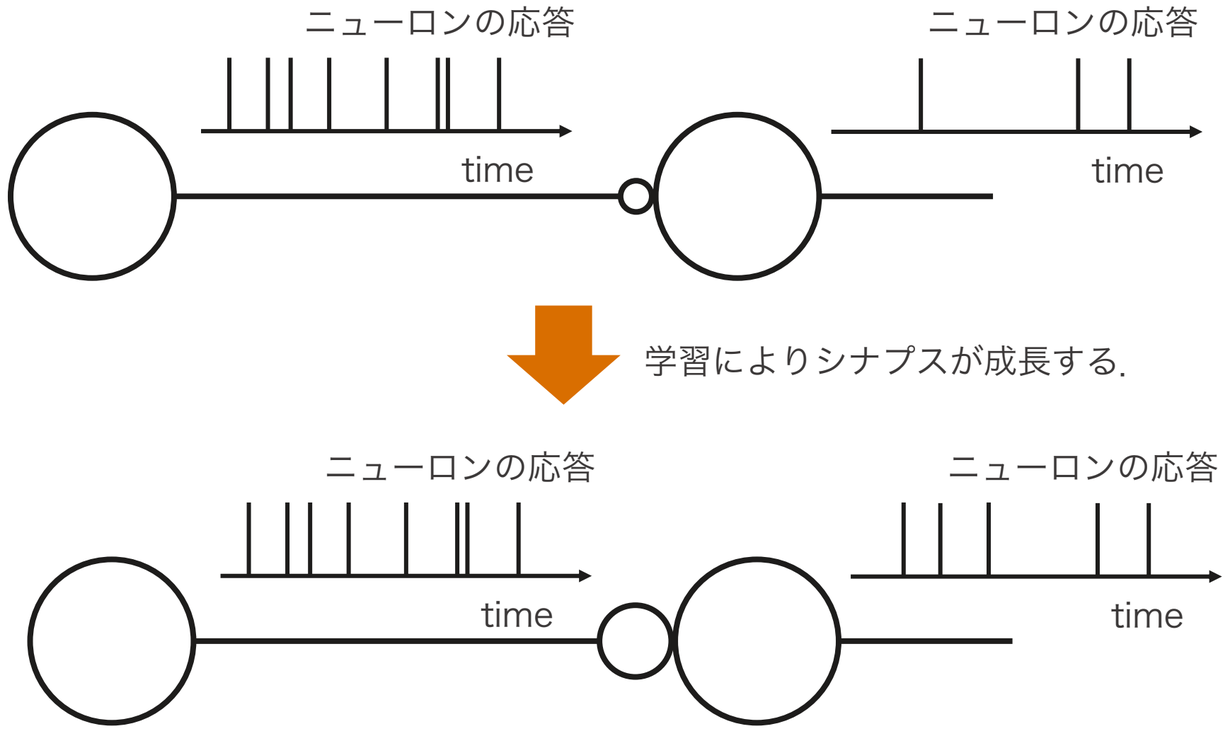
- McCullochとPittsの研究から，神経細胞が論理素子であると考えれば，論理素子で実現できるあらゆる計算を脳が行うことができると考えることもできる．
  - ニューラルネットワークでNAND回路も実装可能なので，ニューラルネットワークであらゆる論理回路が当然実現可能だろう．
- この研究の段階ですでに，時間遅れ，フィードバック接続など考えられている．この考えは後のリカレントネットワークで開花する．



(McCulloch and Pitts, 1946)

Hebbian learning（ヘブ学習）(1949年)

- Hebbが提案した脳の学習の理論
- シナプス前ニューロンが繰り返し発火し，シナプス後ニューロンの発火を助けたとき，そのシナプスは成長する.



Hebbの本では，当時おばあさん細胞説とpopulation codingが議論されていて，population codingが主流であると述べている．Hebbはおばあさん細胞説に基づき議論している．なかなか面白い.

(Hebb, 1949)

線形閾値素子を用い自動で目的の演算を実現するには

- 線形閾値素子の重みを変えることで，線形閾値素子の演算能力が変わる.
- 重みの変更（シナプスの学習）を，ネットワークにさせたい演算に合わせて行えば良い.
- しかし，どのように学習すればネットワークに意図した演算をさせることができるだろうか？

ニューラルネットワーク研究の隆盛と冬の時代

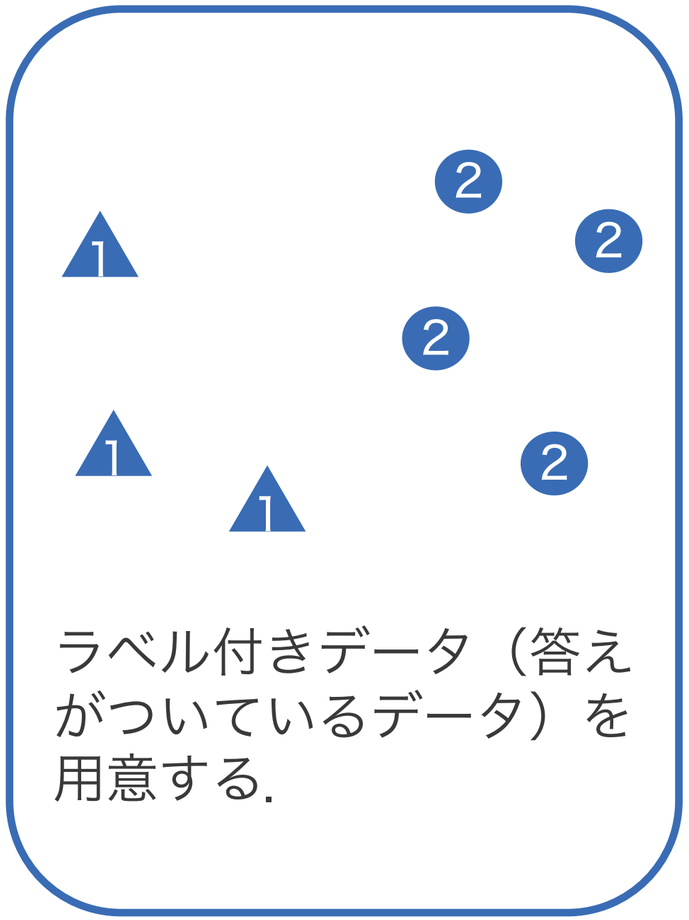
よく見るパーセプトロンの説明

パーセプトロンの説明

- 心理学者Rosenblattが開発したニューラルネットワーク（1957, 1958）
- 識別問題がとける.
- 教師あり学習（答えとネットワークの出力を比べ，それの結果から重みの計算を行う）

識別問題

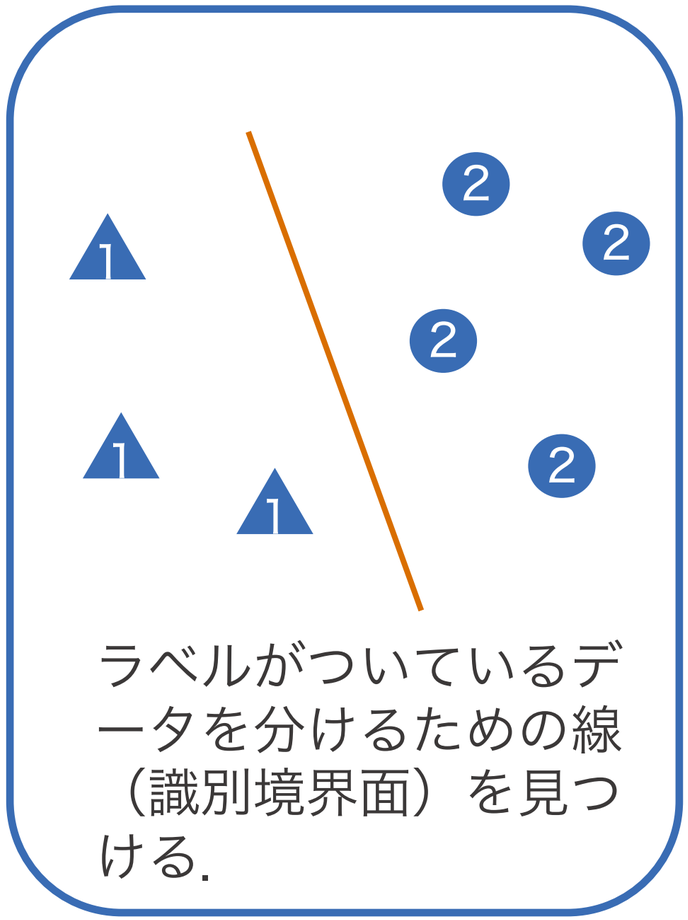
- データにラベルを付ける問題



ラベル付きデータ（答えがついているデータ）を用意する.

三角データに1，丸データに2というラベルが付いている.

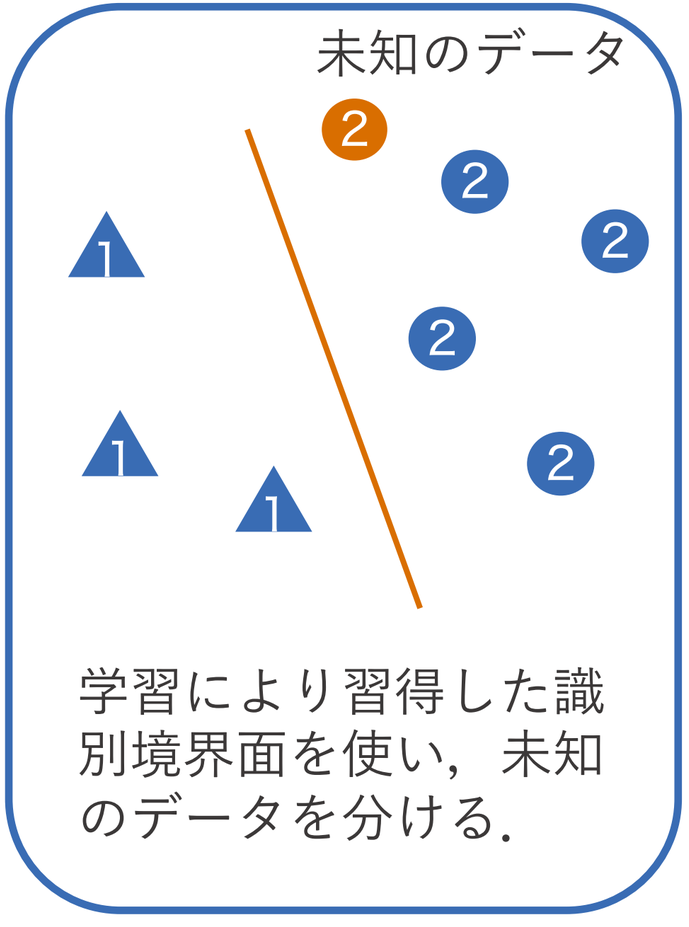
学習



ラベルがついているデータを分けるための線（識別境界面）を見つける.

用意された三角データと丸データをそれぞれ1と2のラベルを付けられる線を見つける．データの場所が線の左なら1，右なら2のらべるが付けられる.

判断

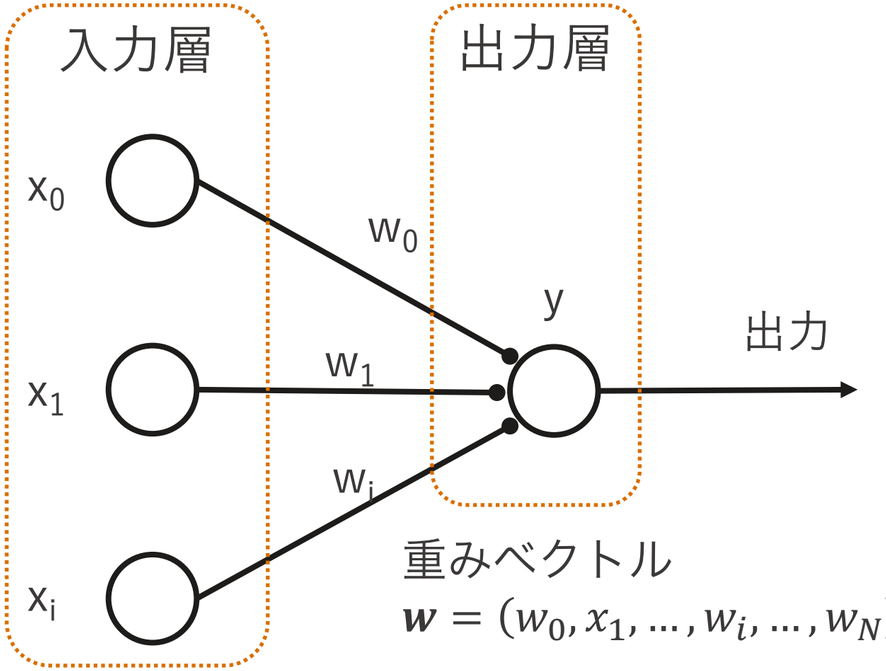


学習により習得した識別境界面を使い，未知のデータを分ける.

未知のデータを入力し，ラベルを付ける．ネットワークは未知のデータが線より左なら1，右なら2のラベルを付ける.

パーセプトロン

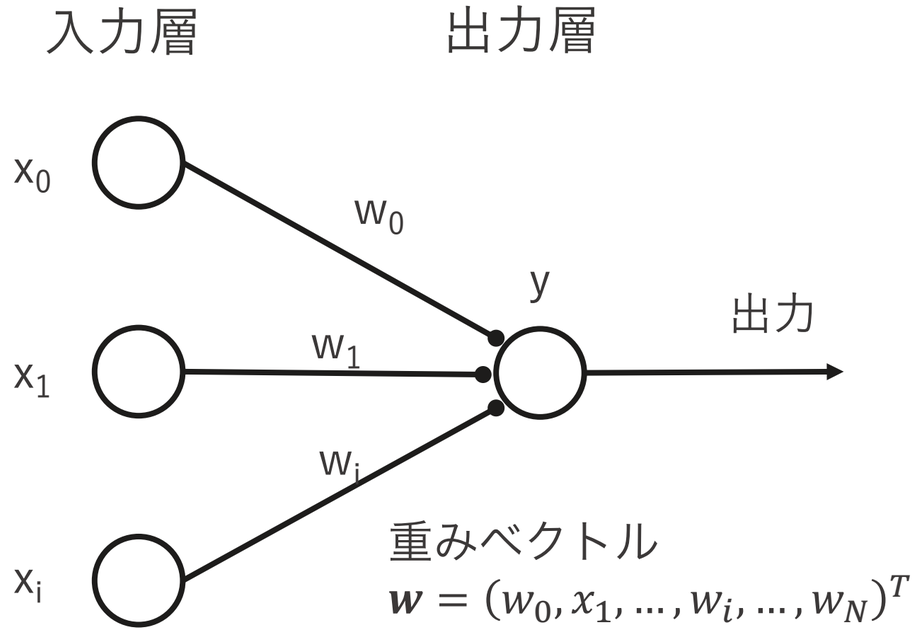
- 2クラス問題が解ける（ラベルが2種類のみ）．
- 入力層と出力層からなる．
- 入力層は入力の値そのものを出力層のニューロンに送る．
- 出力層は閾値素子である．



パーセプトロンの数式表現

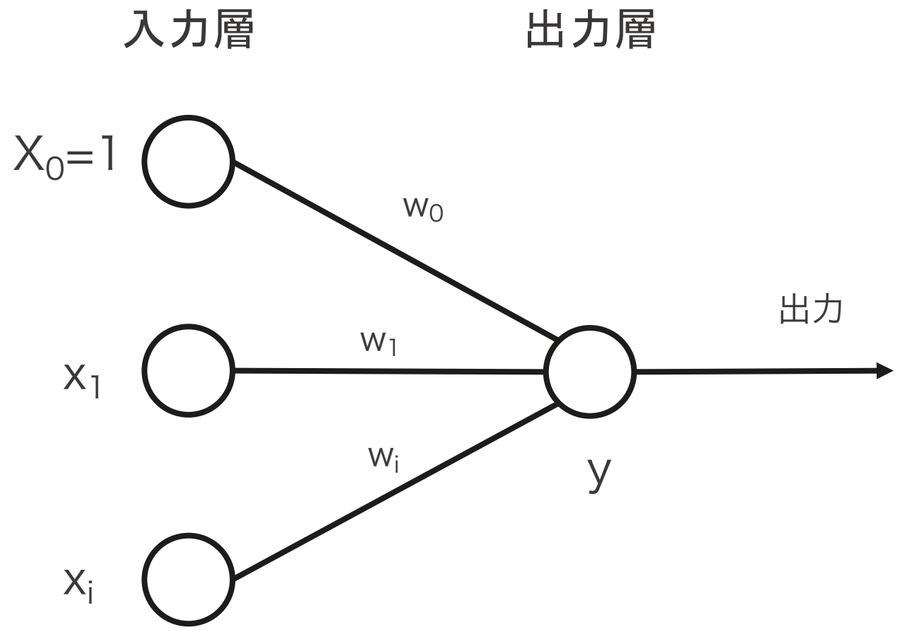
- 入力ベクトル$\boldsymbol{x} = (x_0, x_1, \ldots, x_i, \ldots, x_N)^T$
  - ただし$x_0 = 1$である．$w_0 x_0$をバイアスという．
- 重みベクトル$\boldsymbol{w} = (w_0, x_1, \ldots, w_i, \ldots, w_N)^T$
- $y = f\left(\sum_{i=0}^{N} w_i x_i\right) = f(\boldsymbol{w}^T \cdot \boldsymbol{x})$
- $f(u) = \begin{cases} 1 & \text{if } u \geq 0 \\ -1 & \text{otherwise} \end{cases}$

入力ベクトルと重みベクトルの内積（$\boldsymbol{w}^T\boldsymbol{x} = |w||x|\cos\theta$）が正か負かを基準に，入力ベクトルを分ける．言い換えれば，入力ベクトルと重みベクトルがおおよそ同じ方向を向いている（入力ベクトルが重みベクトルに対し，±90度）かどうか調べている．



パーセプトロンの学習

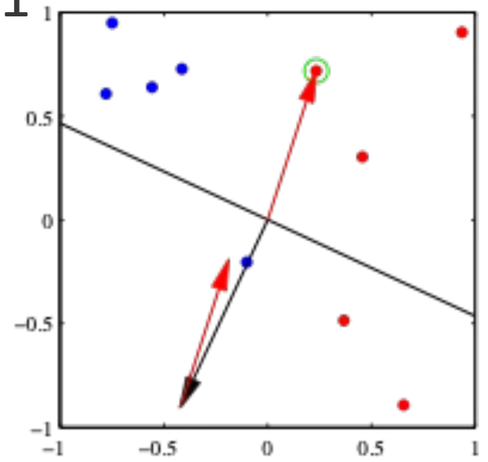
- 出力が 1 になるデータと出力が-1になるデータを用意する.
- あるデータを入れて，間違ったら次の式で重みを更新する.
- $\boldsymbol{w} \leftarrow \boldsymbol{w} + \lambda \boldsymbol{x} t$
- tは答えで1もしくは-1の値をとる． λ は学習率である.



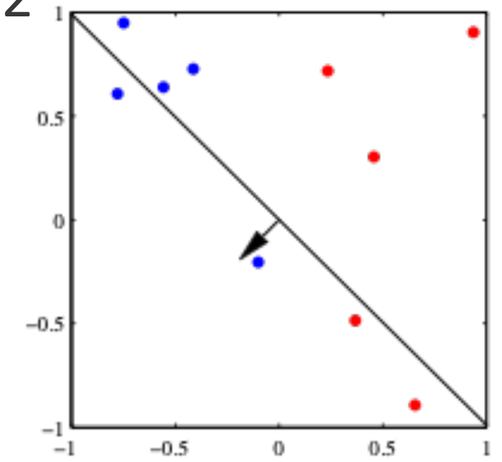
重み修正の様子

1. 出力を1にしなければならないところを-1になってしまったため，ｗに入力xを足した.
2. 識別境界面が更新された.
3. 出力を1にしなければならないところを-1になってしまったため，ｗに入力xを足した.
4. 識別境界面が更新された．その結果，赤丸と青丸が境界面で正しく区分けされた.

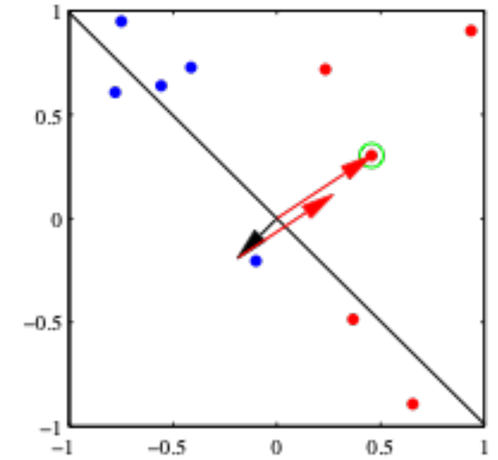
1



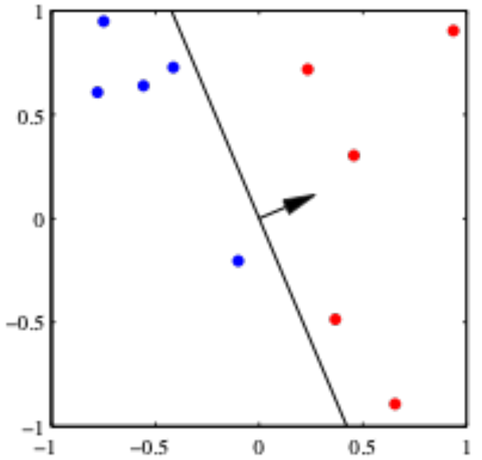
2



3



4



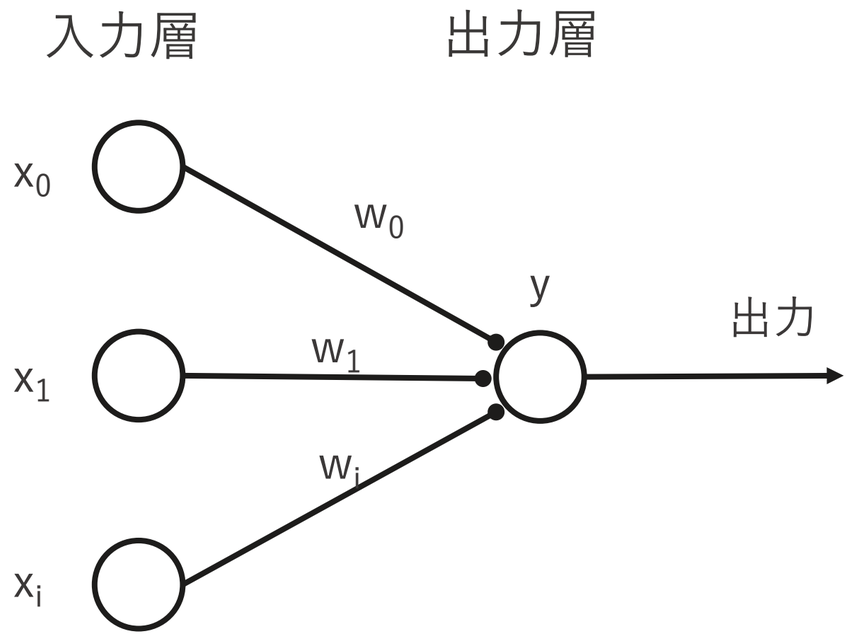
パーセプトロンの学習例

- 入力層は３つのユニット，出力層は1つのユニットで構成されるネットワークを考える．
- このネットワークでAND演算を実現してみよう．

ネットワークに覚えさせる入出力の関係（AND演算）

x0	x1	x2	t
1	0	0	-1
1	0	1	-1
1	1	0	-1
1	1	1	1

ここではTrueを1，Falseを-1としている．



パーセプトロンの学習例

- 初期値：$w_0 = 0, w_1 = 1, w_2 = 1, \lambda = 0.5$とする．
- このとき，出力は$y = f(x_1 + x_2)$と書ける．

- ネットワークにそれぞれの入力を代入してみる．
- $x_0 = 1, x_1 = 0, x_2 = 0$を入力すると，$y = 1$となり不正解
- $\boldsymbol{w} + \lambda \boldsymbol{x} t = (0,1,1) + 0.5 \times (1,0,0) \times (-1) = (-0.5, 1, 1)$
- この学習により，出力は次のようになる．
- $y = f(-0.5x_0 + x_1 + x_2)$

パーセプトロンの学習例

- $y = f(-0.5x_0 + x_1 + x_2)$
- $x_0 = 1, x_1 = 0, x_2 = 1$を入力すると，$y = 1$となり不正解なので学習する．
- $\boldsymbol{w} + \lambda \boldsymbol{x} t = (-0.5,1,1) + 0.5 \times (1,0,1) \times (-1) = (-1, 1, 0.5)$
- この学習により，出力は次のようになる．
- $y = f(-x_0 + x_1 + 0.5x_2)$

パーセプトロンの学習例

- $y = f(-x_0 + x_1 + 0.5x_2)$
- $x_0 = 1, x_1 = 1, x_2 = 0$を入力すると，$y = 1$となり不正解なので学習する．
- $\boldsymbol{w} + \lambda \boldsymbol{x} t = (-1,1,0.5) + 0.5 \times (1,1,0) \times (-1) = (-1.5, 0.5, 0.5)$
- この学習により，出力は次のようになる．
- $y = f(-1.5x_0 + 0.5x_1 + 0.5x_2)$

パーセプトロンの学習例

- $y = f(-1.5x_0 + 0.5x_1 + 0.5x_2)$
- $x_0 = 1, x_1 = 1, x_2 = 1$を入力すると，$y = -1$となり不正解なので学習する.
- $\boldsymbol{w} + \lambda \boldsymbol{x} t = (-1.5,0.5,0.5) + 0.5 \times (1,1,1) \times (1) = (-1, 1, 1)$
- この学習により，出力は次のようになる.
- $y = f(-x_0 + x_1 + x_2)$

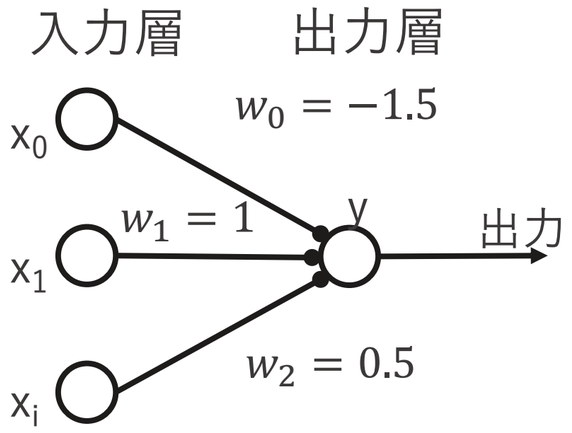
パーセプトロンの学習例

- $y = f(-x_0 + x_1 + x_2)$
- $x_0 = 1, x_1 = 0, x_2 = 0$を入力すると，$y = -1$となり正解
- $x_0 = 1, x_1 = 0, x_2 = 1$を入力すると，$y = 1$となり不正解なので学習する．
- $\boldsymbol{w} + \lambda \boldsymbol{x} t = (-1,1,1) + 0.5 \times (1,0,1) \times (-1) = (-1.5, 1, 0.5)$
- この学習により，出力は次のようになる．
- $y = f(-1.5x_0 + x_1 + 0.5x_2)$

パーセプトロンの学習例

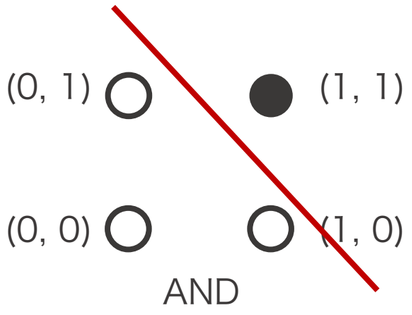
- $y = f(-1.5x_0 + x_1 + 0.5x_2)$
- $x_0 = 1, x_1 = 1, x_2 = 0$を入力すると，$y = -1$となり正解
- $x_0 = 1, x_1 = 1, x_2 = 1$を入力すると，$y = 1$となり正解
- $x_0 = 1, x_1 = 0, x_2 = 0$を入力すると，$y = -1$となり正解
- $x_0 = 1, x_1 = 0, x_2 = 1$を入力すると，$y = -1$となり正解
- よって，すべての入力に対し正解したので学習を終了する.

AND演算ができる
ニューラルネット
ワーク

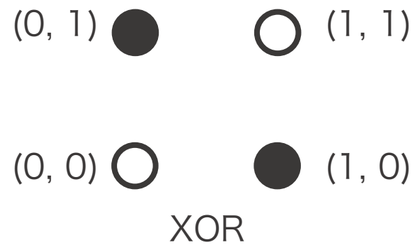


よく言われるパーセプトロンの欠点

- 線形分離不可能な問題は解けない
  - 例：XOR問題が解けない
  - これは2層のパーセプトロンの問題である.
    - 多層化が難しく，Activation functionの連続関数化とBackpropagationにより多層化問題が解消したと言われる.
- MinskyとPapertによる指摘によりニューラルネットワークブームが終わったと言われることが多い.
  - どうでも良いが，RosenblattとMinskyは同じ高校出身(１年違い？)であった.



ANDの場合，直線で分けられる（線形分離可能）.



XORの場合，直線で分けられない（線形分離不可能）.

MinskyとPapertのPerceptronsではx=yを判別することができないことを示している.

Rosenblattのパーセプトロン

Rosenblatt （1958） のパーセプトロン

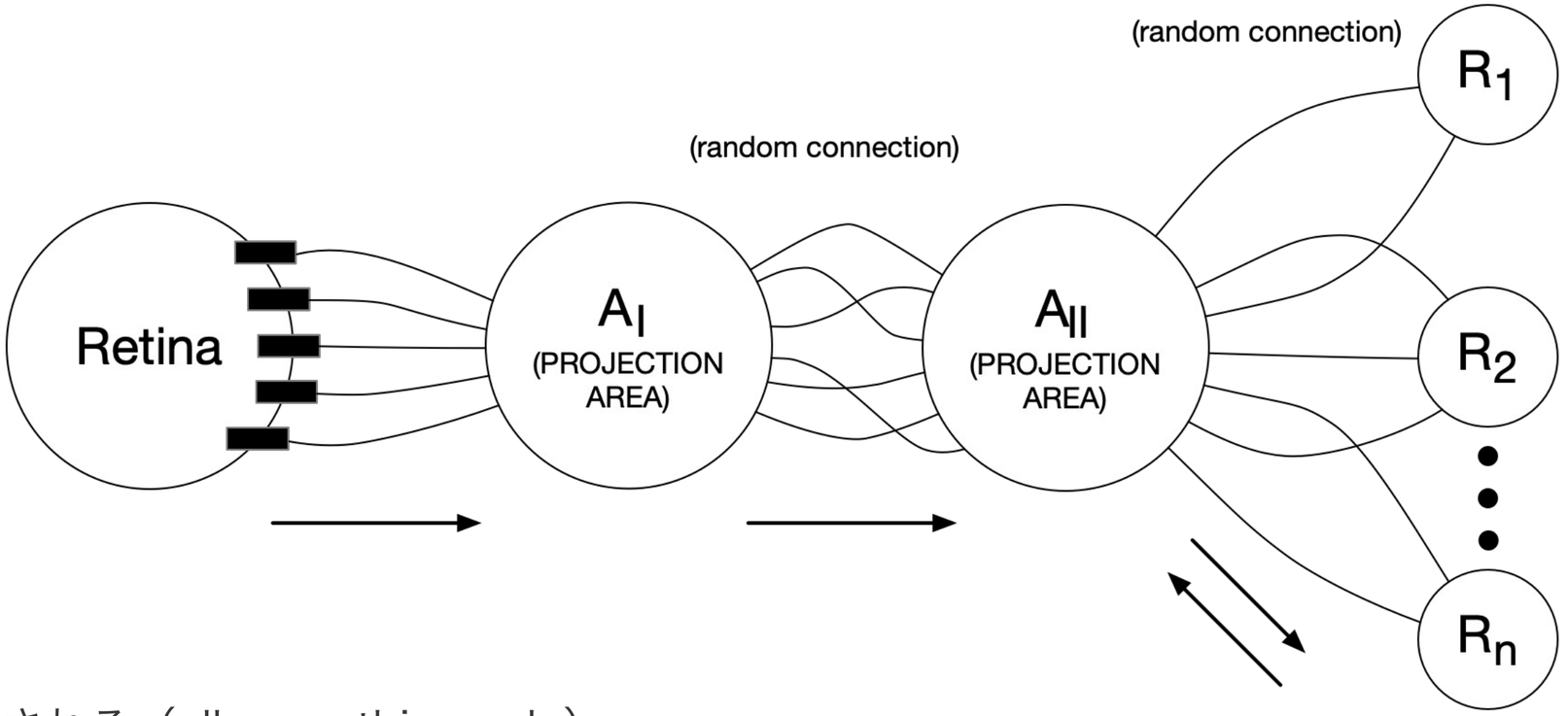
- **4層もしくは3層構造である.**
- **ランダム接続を持つ.**
- **受容野構造を持つ.**
- 層内，層間の抑制性結合を持つ.
- **フィードバック接続を持つ.**
- 脳のモデルでもあり，並列計算機でもある.

- フィードフォワード学習である（Rosenblatt, 1962）.
- 誤差のバックプロパゲーションも考えている（Rosenblatt, 1962）.

- これらを見ると，かなり先進的なモデルであったことが伺える.

パーセプトロンのネットワーク構造

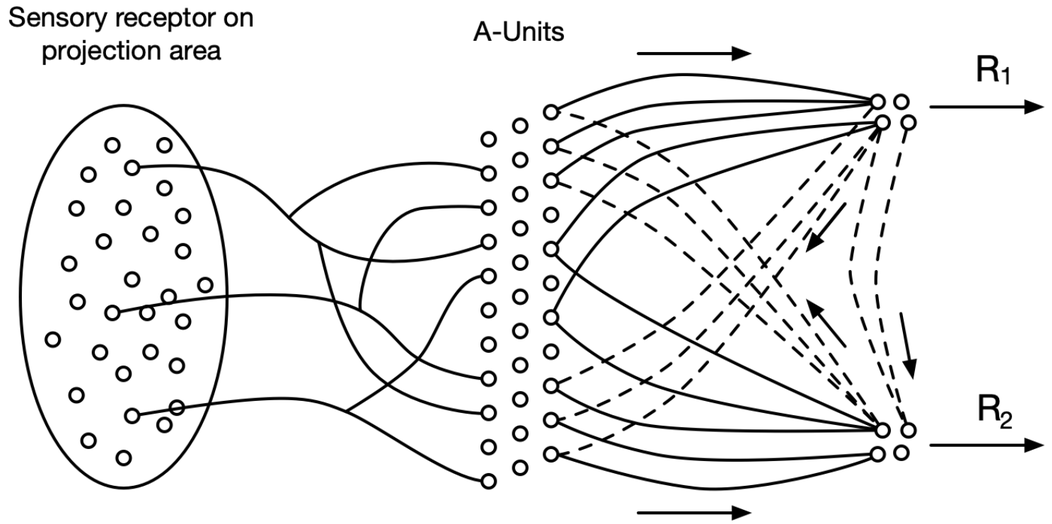
- Rossenblatt, 1958で提案されたパーセプトロンは4層で構成される.



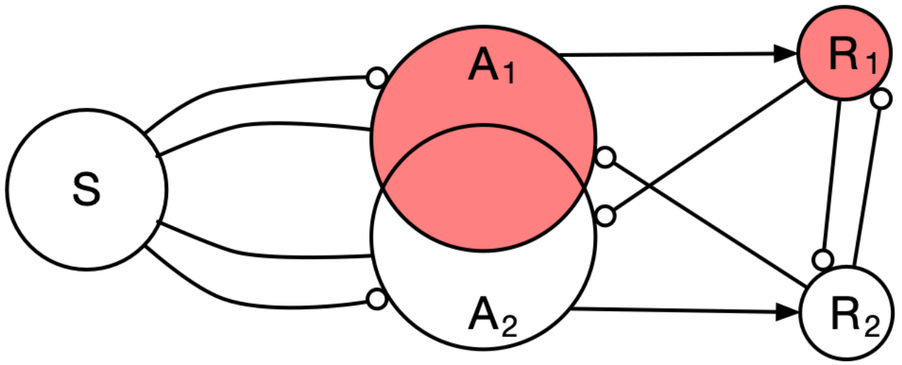
- **閾値素子**で構成される（all-or-nothing rule）.
- Retina (S-points)の出力はA_Iに送られる. **興奮性と抑制性の接続**を想定している. A_IはRetinaに**受容野を持ち**, 重みはある点を中心に指数関数的に減衰する. A_Iは省略される事がある.
- A_IとA_{II}は**ランダムに接続**している.
- A_{II}とRはランダムに接続している. 接続は**相互接続**（フィードフォワードとフィードバック）である.

Rossenblattのsimple perceptron (1958)

- Rossenblattは3層のパーセプトロンも提案している.
- RossenblattもMinskyとPapertも基本的に3層のパーセプトロンについて議論している.



モデル図. 点線は抑制性接続.

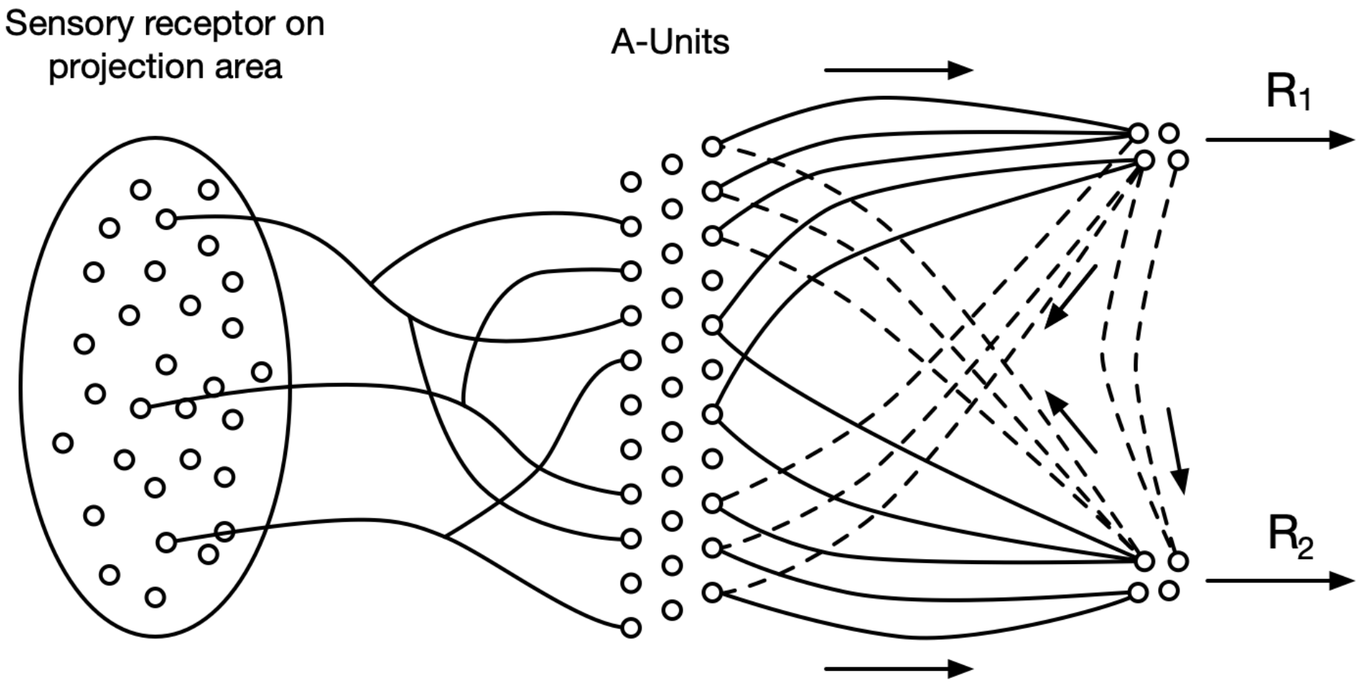


Rossenblatt曰くベン図. 白丸は抑制性接続. 色付きの領域はR1が応答したときに活性化するunitsのセット.

A-unitはそれぞれ**ランダム**にretinaに接続している.
ランダム接続は, 輪郭線というより同期領域を捉える（Rossenblattは時間変化も考慮している）.
抑制性接続により, R1が応答した場合, R2に関連するunitsのセットの応答は抑えられる.
学習するためには, A-unitか接続を変更する必要がある.

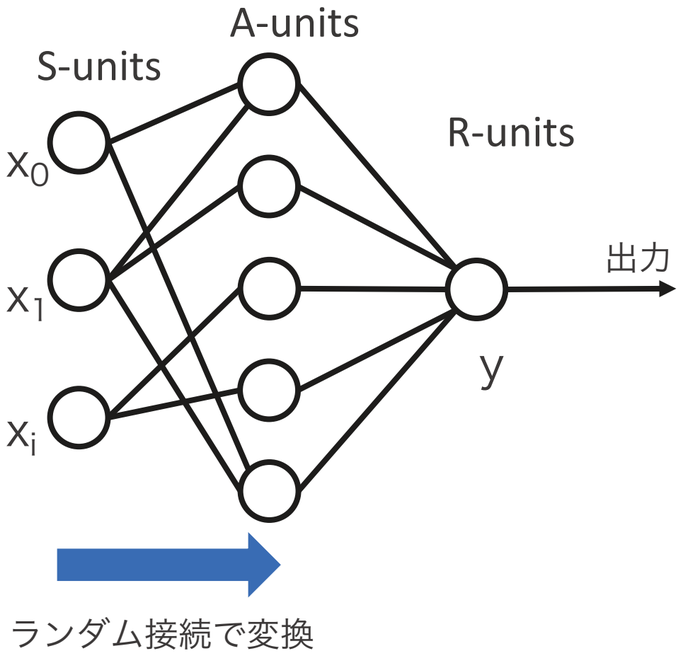
Rossenblattのパーセプトロンではランダム接続がある.

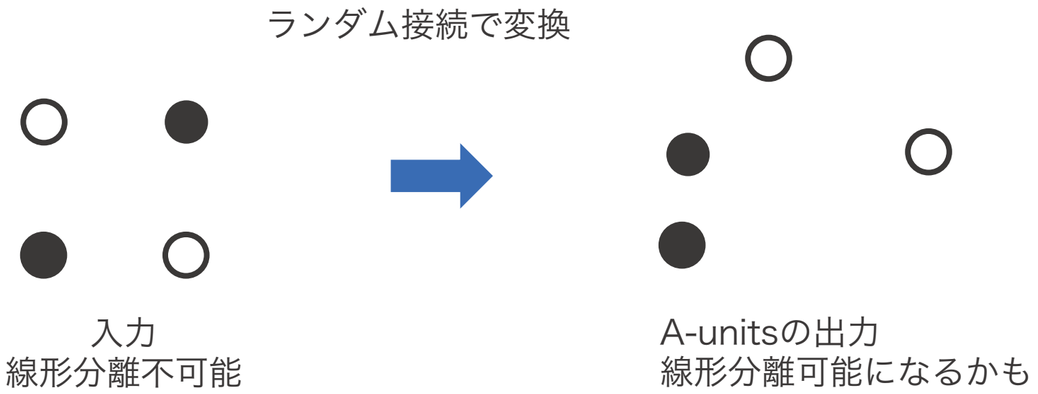
- RetinaとA-unitsはランダムに接続している.
- ランダム接続には含蓄がある.



ランダム接続の含蓄（役割）

- 2層のパーセプトロンは線形分離可能な問題しか解けない.
- 線形分離不可能な問題を解くにはどうすればよいか？
  - 入力をどうにか変形して線形分離可能にする.
  - ランダム接続で入力を変換し，偶然線形分離可能な形になることを期待する.





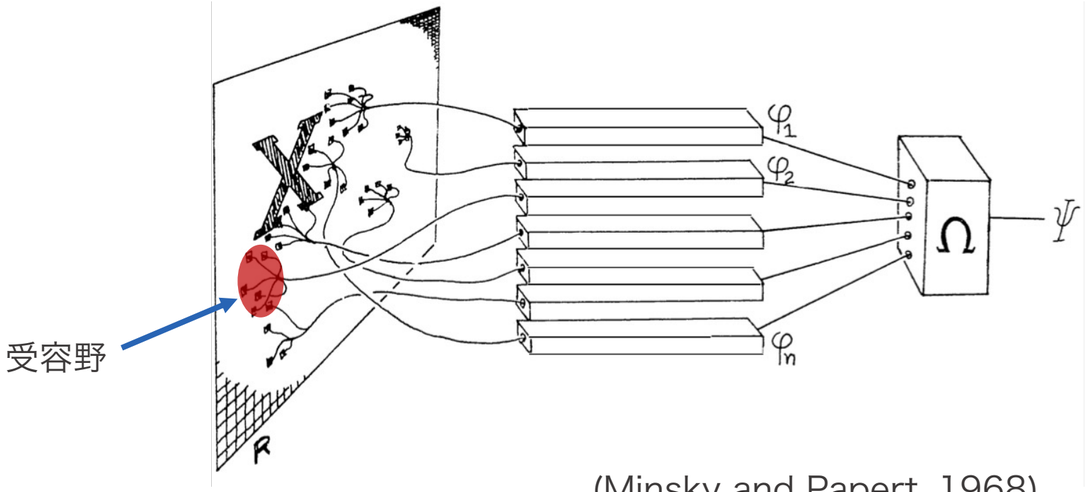
あくまでも概念図です.

ランダム接続の含蓄（受容野）

- 入力の特徴を捉える特別な接続（受容野）を考えることもできる．
- 特別な接続を作るのは難しい．一方で，ランダム接続を作るのは簡単である（Minsky and Papert, 1968）．
  - 畳み込みニューラルネットワークは，局所特徴を学習により捉える接続を作ることができる．

極端な例を考えると，S-unitsとA-unitsの接続を文字の形にしてしまえば，A-unitsは文字それぞれの文字に対応して応答する．Xの形の接続を持つA-unitはXのみに応答する．しかし，そのような接続では，文字の大きさが変わると対応できない．

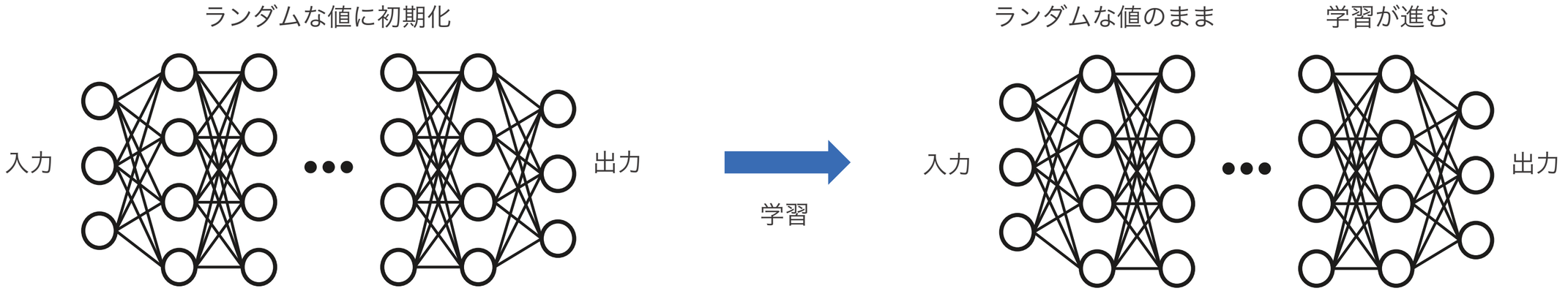
MinskyとPapertはS-unitsとA-unitsの接続について色々考察している．



(Minsky and Papert, 1968)

ランダム接続の含蓄（深い多層パーセプトロン）

- 勾配消失問題により多層パーセプトロンは層を増やしても意味がない．
  - 深い（層が多い）多層パーセプトロンは学習が出力に近い層のみで行われ，入力に近い層では行われない．そのため，層を増やしても意味がない．
  - 通常，層間の接続はランダムな値で初期化されるので，学習が進まない下の層はランダム接続であると考えられる．つまり，深い多層パーセプトロンは，Rosenblattのパーセプトロンのように，ランダムな層と全結合の層で構成されているとも考えられる．

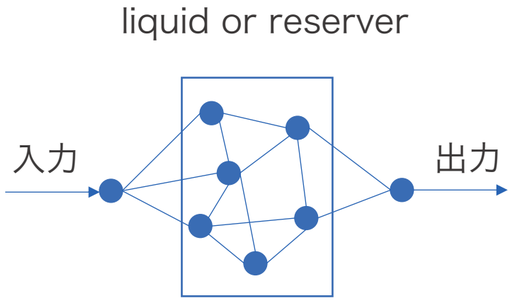


ランダム接続の含蓄（結局運次第か）

- パーセプトロンの識別能力はランダム接続で入力の特徴を捉えられるかどうかで決まる．つまり，運次第ということである．
- ニューラルネットワークの能力は運次第であるという考え方は，深層ニューラルネットワークにも当てはまるかもしれない．
  - 深層ニューラルネットワークでは，当たりのランダムな接続が存在し，当たりの接続を探しそれを学習すれば高性能になるという，宝くじ仮説というものもある (Frankle and Carbin, 2019)．（この説明は不正確なため元論文をチェックしよう）

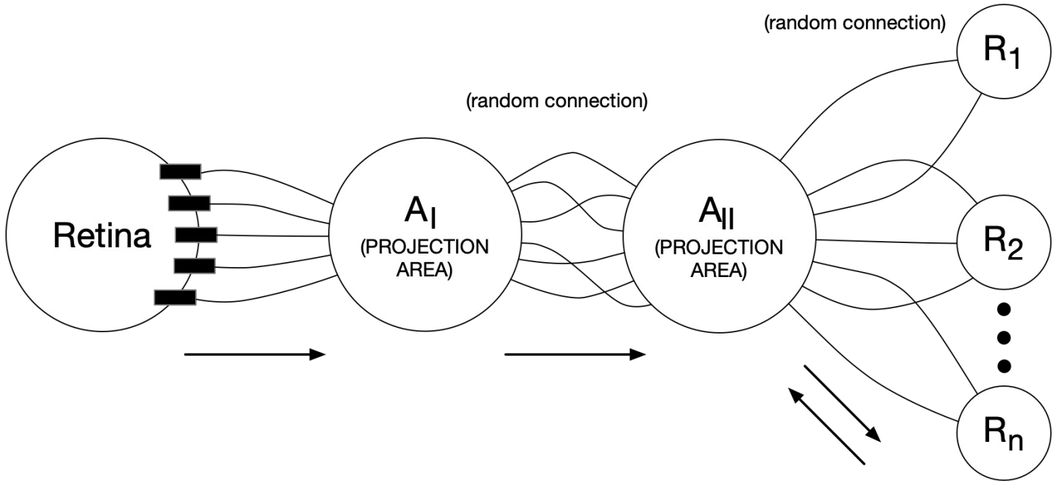
ランダム接続の含蓄（Liquid state machine）

- ランダム接続を持つニューラルネットワークは現在盛んに研究されている.
  - Rosenblattのパーセプトロンは層間をランダムに接続した.
  - 現在よく研究されているランダム接続を持つニューラルネットワークは，層内の接続がランダムであるニューラルネットワークである.
    - 例：Liquid state machine (Maass, 2002), Echo state network (Jaeger, 2001; 2002), リザーバーコンピューティング
  - Rosenblattはパーセプトロンで時系列を学習させることに限界を感じていたようだ（Rosenblatt, 1958）．しかし，Liquid state machineなどでは層内接続をランダムにし接続間の信号伝達に時間遅れを入れることで，時系列の特徴を捉えることを可能にしている.
  - Rosenblattは惜しい所まで来ていた.



フィードバック接続

- A層とR層にフィードバック接続がある.
  - これは，現在のリカレントニューラルネットワーク（RNN）に繋がる.
  - RNNは時系列を学習できるニューラルネットワークである.
  - 現在，自然言語処理などで活用されている（代表的な例が自動翻訳）.
  - Rosenblattはパーセプトロンで時系列を学習させることの限界をRosenblatt, 1958のまとめで述べている.
  - Rosenblattのパーセプトロンが連続時間であることが影響しているのだろうか（RNNは離散時間）？



パーセプトロンの学習

- RosenblattのNeurodynamicsに書かれている学習
  - Hebbian learningを採用している.
  - α -system reinforcementの例
    - ユニットiが活性化したとき，それと接続するユニットjとの間の重みは次の式で更新される.
    - $\Delta w_{ij} = \eta$

- MinskyとPepertのPerceptronsに書かれている学習の１例
  - 今Percetronの学習として伝わっている，学習手法が書かれている.

パーセプトロンは脳型ニューラルネットワークである.

- Rossenblattも脳の生理学的，解剖学的知見に基づきパーセプトロンを作成，考察している.
- 連続時間を考慮したモデル.

- そもそものニューラルネットワークは脳のニューラルネットワークのモデルであった.
  - 時代が進むにつれニューラルネットワークが人工ニューラルネットワークになった.

パーセプトロンは並列計算をするコンピュータである

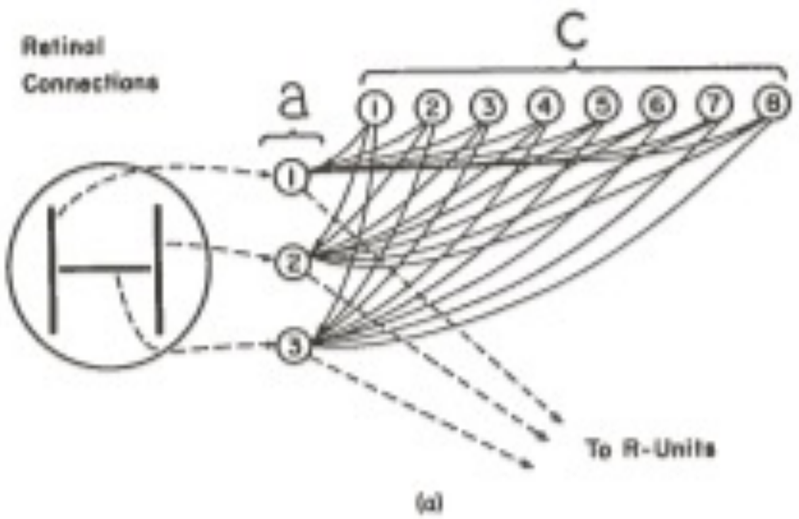
- パーセプトロンは並列計算をするコンピュータも目指している
  .

- コンピュータの歴史
  - 1946年 ENIAC
  - 1951年 UNIVAC 1 汎用コンピュータ
  - 1952年 IBM701 商用科学技術計算機
  - 1954年 IBM704 Rosenblattこれを使ってパーセプトロンを動かす.
  - 1956年 FORTRAN プログラミング言語
  - 1961年 IBM7030
  - 1964年 System/360 メインフレーム

おまけ
コンピュータの計算力が小さい時代，シミュレーションはコンピュータを使うのではなく電気回路でやっていた（数理モデルを等価回路に置き換え，実際にその電気回路を組み，その回路に電流を流すことで数理モデルを解く）.

Rosenblattのc-system

- Rosenblattの開発したc-systemは畳み込みニューラルネットワークの元祖と呼べるかもしれない.



(a)

Stimulus:	S_1	S_2	S_3	S_4
Retinal pattern:	‖	⊢	⊣	H
a_1	1	1	0	1
a_2	1	0	1	1
a_3	0	1	1	1

(b)

C-states:	C_1	C_2	C_3	C_4
c_1	1	1	1	1
c_2	1	1	0	0
c_3	1	0	1	0
c_4	1	0	0	1
c_5	0	1	0	0
c_6	0	1	1	1
c_7	0	0	0	1
c_8	0	0	1	0

(c)

(Rosenblatt, 1967)

よく言われるニューラルネットワークの冬の時代

- パーセプトロンの当時，ニューラルネットワークがブームになったと言われている．
- しかし，MinskyとPaperとのPerceptronsという書籍で，パーセプトロンの限界が示された（線形分離不可能な問題が解けない）ため，ニューラルネットワークの研究が下火になり，冬の時代が訪れたと言われている．

- 本当にそうなのだろうか？