

# 情報意味論 コネクショニズム connectionism

櫻井彰人  
慶應義塾大学理工学部

## まずは

- ニューラルネットとは ( or コネクショニズム or PDP Parallel Distributed Processing)?
- ニューロンと関係あるのか?
- 何ができるのか?
- どうやってできるのか?
- 人間の認知とどう関係するのか?

## ニューロンとは?

- “There is no such thing as a typical neuron”, Arbib, 1997

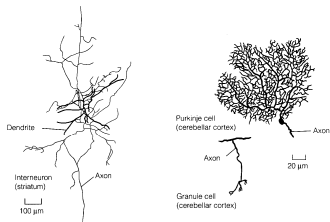


Fig. 1. The morphologies of three common types of neuron. The full length of the axons is not shown. The bifurcating axon of the granule cell extends for several millimeters in each direction. Note how the axon of the interneuron branches extensively.

## 多分典型的なニューロン

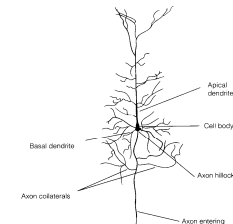
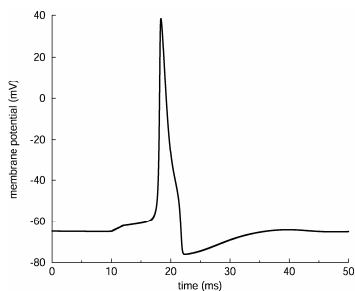


Fig. 1. Key features of a neuron. A drawing of a pyramidal cell showing the distribution of neurites (dendrites and axon).

## ニューロンの活動 その1



## ニューロンの活動 その2

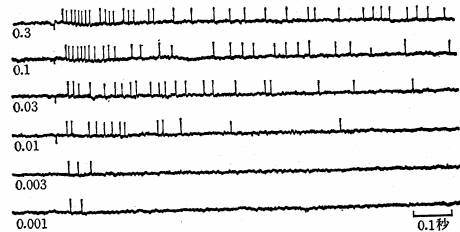


図 27 ネズミの味覚の神経線維に現われるインパルス. 数字は食塩のモル濃度 (C. Pfaffmann)

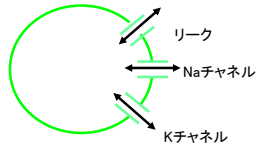
## Hodgkin-Huxley方程式

$$C_m \frac{dV}{dt} = -g_L(V - E_L) - g_{Na}m^3h(V - E_{Na}) - g_Kn^4(V - E_K) + I$$

$$\frac{dm}{dt} = \alpha_m(V)(1-m) - \beta_m(V)m$$

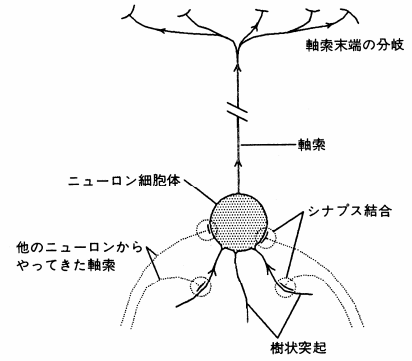
$$\frac{dh}{dt} = \alpha_h(V)(1-h) - \beta_h(V)h$$

$$\frac{dn}{dt} = \alpha_n(V)(1-n) - \beta_n(V)n$$



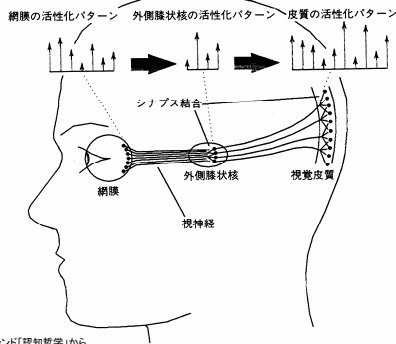
V: 膜電位、m, h(Na), n(K): チャンネルが開く確率

## ニューロンの結合



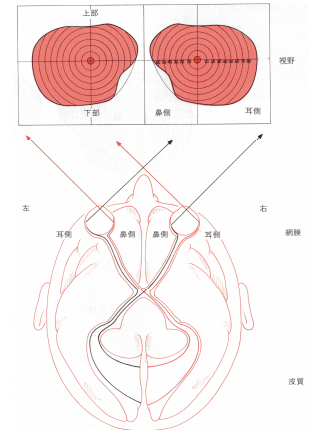
チャーチランド「認知哲学」から

## 視覚神経の結合



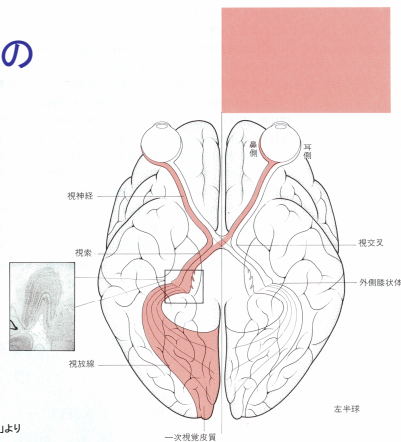
チャーチランド「認知哲学」から

## 網膜との結合



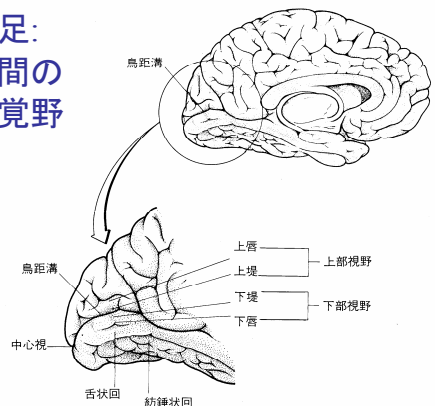
ゼキ「脳のビジョン」より

## 網膜との結合 (続)



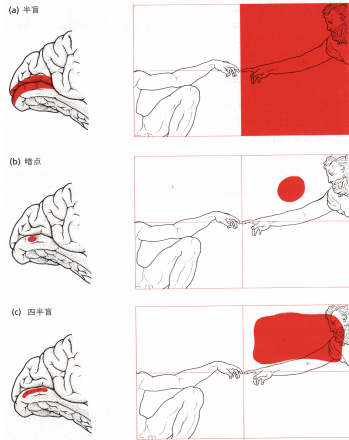
ゼキ「脳のビジョン」より

## 補足: 人間の視覚野



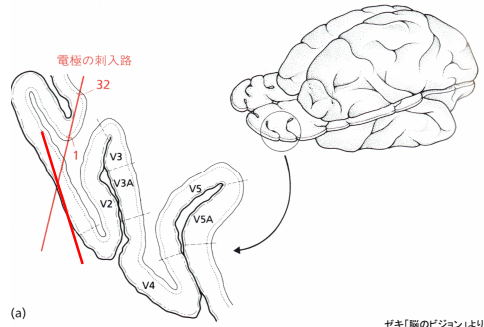
ゼキ「脳のビジョン」より

# 大脳性盲



ゼキ「脳のビジョン」より

# 電極の挿入



(a)

ゼキ「脳のビジョン」より

# V1野コラム構造の発見

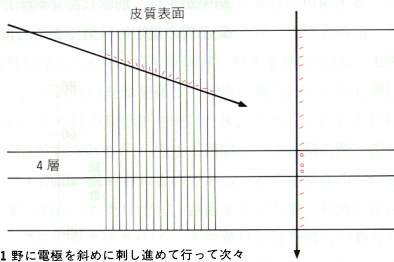


図 22-2 V1野に電極を斜めに刺し進めて行って次々に捉えられる細胞の方位選択性の変化 (Hubel, D.H. & Wiesel T.N. (1977). *Proc. R. Soc. Lond. B* 198, 1-59. より転載). (発見自体は、1959年。一次体性感覚野については、Mountcastle が1957年)

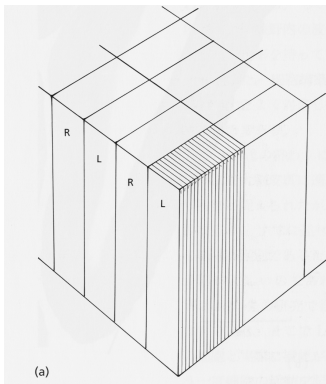
ゼキ「脳のビジョン」より

# 解剖学的手法 (眼優位)



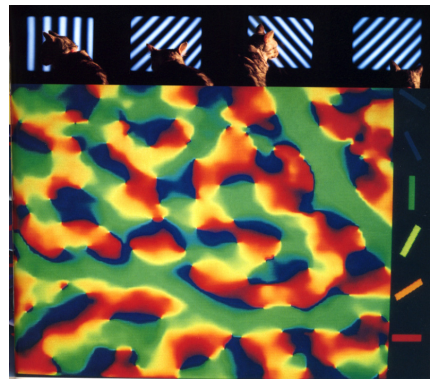
ゼキ「脳のビジョン」より

# 眼優位コラムと方位優位コラム



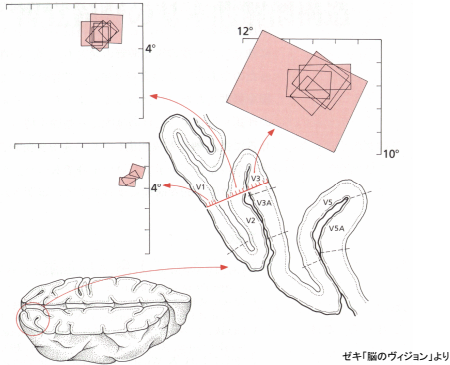
ゼキ「脳のビジョン」より

# 光活性物質 (方位選択性)



「原典の小平宙・人体 II」より

## V1~V3野の受容野の大きさ



## 階層構造概念の発展

- Hubel & Wiesel 以前:
  - (一次) 視覚野で中継し、連合野で高次処理
- 高次視覚野におけるコラムと受容野の変化
  - V1→V2→V3→... といった階層構造
  - V1: 特徴抽出 (特に傾き)
  - V2: 対象受容野の拡大
  - V3: 形の特徴
  - V4: 色
  - V5: 運動

## ネットワークがモデル化できるか？

- できない。ニューロンモデルが複雑すぎる。
- 単純化する以外にない
  - 40年前の話ですから。
  - といっても、今でも事情はそうは変わらない

## McCulloch and Pitts

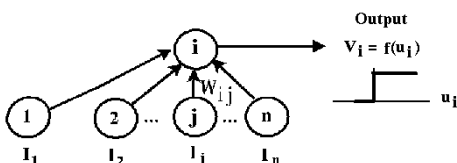
- Warren S. McCulloch and Walter Pitts (1943) "A logical calculus of the ideas immanent in nervous activity", *Bulletin of Mathematical Biophysics*, 5: 115-133.



- ニューロンの非常に単純化した(しかし数学的な)モデル
- もしニューロンがこのようなであれば、任意の関数が計算できることを示した
- 

## ニューロンのモデル

- 個々のニューロンは、入力値をある関数に従って処理する
- 初期のモデルは階段関数を用いた。現在では、シグモイド関数を用いることが多い



## 視覚野のモデル(単純型細胞)

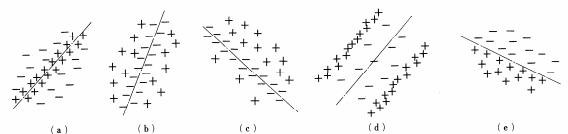


図 4.4 種々の単純型細胞の受容野 (Hubel & Wiesel<sup>1)</sup> を変更)

# 視覚野のモデル(複雑細胞)

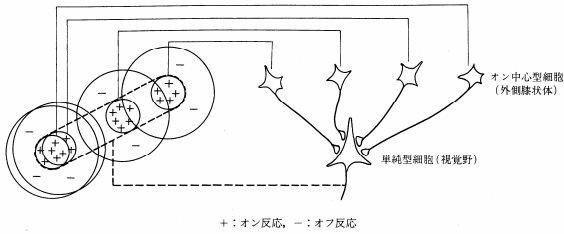


図 4.5 単純型細胞のシナプス結合モデル (Hubel & Wiesel<sup>19)</sup> を変更)

福島「神経回路と情報処理」より

# 視覚野のモデル(複雑型細胞)

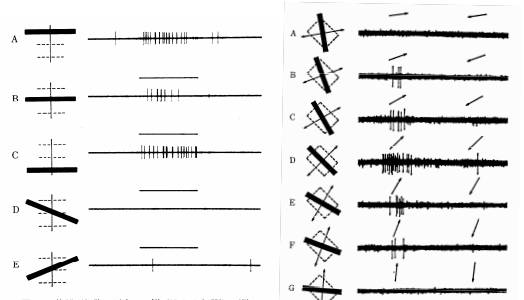


図 4.6 複雑型細胞の反応の一例 (Hubel & Wiesel<sup>19)</sup> 刺激パターンの点滅に対する反応を示す。

図 4.7 運動刺激に対する複雑型細胞の反応 (図 4.6 とは別の細胞) (Hubel & Wiesel<sup>19)</sup>)

福島「神経回路と情報処理」より

# 視覚野のモデル(超複雑型細胞)

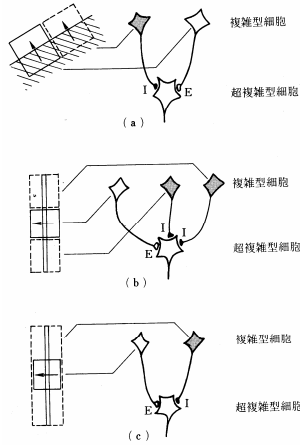


図 4.10 超複雑型細胞のシナプス結合モデル (Hubel & Wiesel<sup>19)</sup> を変更) E: 興奮性シナプス, I: 抑制性シナプス。

福島「神経回路と情報処理」より

# 視覚野のモデル(超複雑型細胞)

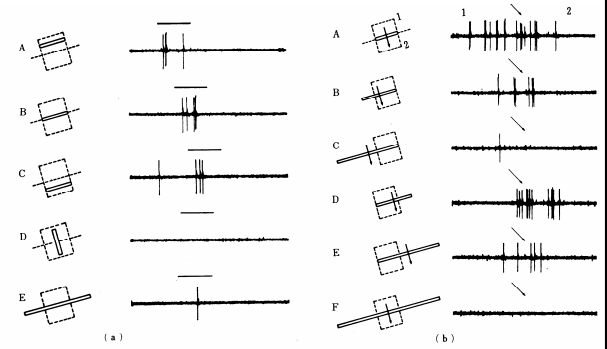


図 4.9 超複雑型細胞の反応 (Hubel & Wiesel<sup>19)</sup>)

福島「神経回路と情報処理」より

# コグニトロン

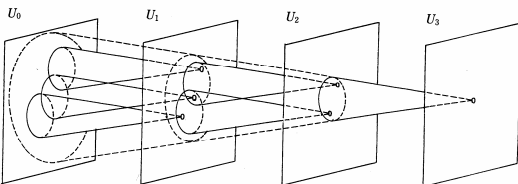


図 5.4 コグニトロンにおける細胞層間の結合の広がり (福島<sup>20)</sup>) 個々の細胞の結合領域は小さいが、各細胞はそれぞれ一つの層の多数の細胞の出力情報を統合して行くので、層を重ねるごとに受容野は大きくなる。

福島「神経回路と情報処理」より

# ネオコグニトロン

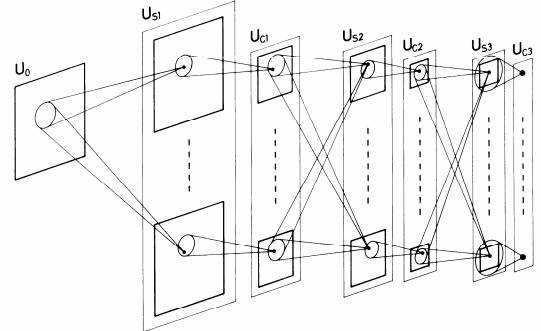


図 6.1 ネオコグニトロンの各層の細胞間の結合状態を示す概念図 (福島<sup>12,27)</sup> を変更)

福島「神経回路と情報処理」より



## 脳のモデル化ができるか？

- できない。
- と思う。
- 参考: コンピュータが(当分の間)脳にならない理由
  - active 素子の個数の違い
  - 結線数の違い
  - ソフトウェアの違い。
    - 真似するにせよしないにせよ、全くわかっていない。測定もできない
    - 大雑把には、測定できるようになってきている
  - cf. 「ポスト・ヒューマン誕生—コンピュータが人類の知性を超えるとき」(原題 "The Singularity is Near") (レイ・カーツワイル, NHK出版, 2007年)

## ニューロンにヒントを得た情報処理

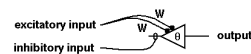
- ニューラルネットは、ニューロンにヒントを得た情報処理モデルというべき
- 脳について知りえたことより、大幅に簡化されているのが、普通—とはいえ、イノベーションはしばしば脳研究から、例:
  - Spiking neural nets

## ニューロンにヒントを得た情報処理

- ニューラルネットモデルは、超並列
  - 単純な処理装置の大量接続
- 記号処理とは異なった情報処理が行える
  - 異なる原始機能 (実行が容易) が使える
- 学習ができる

## 再掲: McCulloch and Pitts

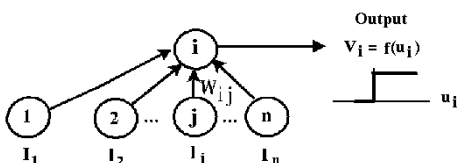
- Warren S. McCulloch and Walter Pitts (1943) "A logical calculus of the ideas immanent in nervous activity", *Bulletin of Mathematical Biophysics*, 5: 115-133.



- ニューロンの非常に単純化した(しかし数学的な)モデル
- もしニューロンがこのようなであれば、任意の関数が計算できることを示した
- けれども、学習は...? あとで。

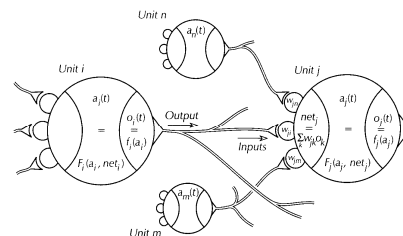
## 再掲: 処理系としてのニューロン

- 個々のニューロンは、入力値をある関数に従って処理する
- 初期のモデルは階段関数を用いた。現在では、シグモイド関数を用いることが多い



## モデルニューロンのネットワーク

- ニューロンの単純化した数学的モデルが考えられた...



## Donald Hebb

- Donald O. Hebb (1949) "The Organization of Behavior", New York: Wiley
- "What fires together, wires together"
- 生物学的には首肯できる
- 今でも用いられることがある (普通は使わない), 「ニューロン間の結合荷重は、その活動の相関に依存して変化する」という一般的な概念は、広く用いられている。

## パーセプトロン Perceptron

- Rosenblatt, F. (1957). "The perceptron: A perceiving and recognizing automaton (project PARA)", Technical Report 85-460-1, Cornell Aeronautical Laboratory.
- Rosenblatt, F. (1962). "Principles of Neurodynamics.", Spartan Books, New York.

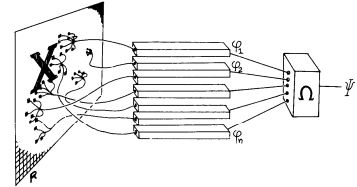


FIGURE 1. The one-layer perceptron analyzed by Minsky and Papert. (From *Perceptrons* by M. L. Minsky and S. Papert, 1969, Cambridge, MA: MIT Press. Copyright 1969 by MIT Press. Reprinted by permission.)

## パーセプトロン

- 何ができるか?
  - 文字(アルファベット)認識
  - いくつかのパターン認識課題 (形の認識等.)
  - しかも、パーセプトロン学習規則は、それが解くことができる全ての課題について、解を発見することができる、と証明できる

## The Perceptron

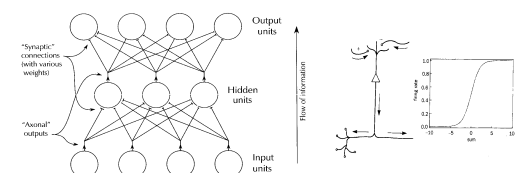
- 何ができないか?
  - パリティ
  - 結合性
  - XOR 問題
  - 線形分離可能でない問題
- Marvin L. Minsky and Seymour Papert (1969), "Perceptrons", Cambridge, MA: MIT Press
- McCulloch & Pitts ニューロンのネットワークは Turing 機械と等価; でも 'それで?':
  - 学習させる方法知らない
  - 予想: 任意のネットワークを学習させるアルゴリズムは、単に、存在しない

## PDP

- "Perceptrons" のせいで、この分野の研究が20年遅滞したという...
- 転機: D.E. Rumelhart, J.L. McClelland, eds., "Parallel Distributed Processing: Explorations in the Microstructure of Cognition", MIT Press, 1986.
  - 論文の集成、数学的なものから哲学的なものまで
  - うまくいった実験結果をたくさん示している一方:
  - 誤差逆伝播学習アルゴリズム back propagation learning algorithm: 結局のところ多くのニューラルネットワークの学習を可能とした。
  - [実は、類似の技法は、この間、発見されていた (Amar 1967; Werbos, 1974, "dynamic feedback"; Parker, 1982, "learning logic") ので、再発見という言葉が適しているよう。しかし、この分野を再出発させたことは大きな成果である。]

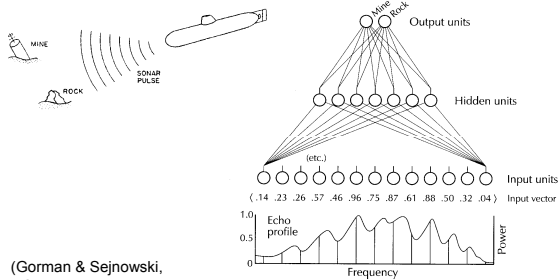
## Back Propagation

- 適用範囲: multi-layer 'feed-forward' (only) networks:



- 結合荷重は、誤差を逆伝播したものに比例して修正される...

# 何ができるか?



(Gorman & Sejnowski, NC 1(1), 75-89 (1988))

# どうやって?

- 誤差関数の最急降下 (濃い霧の中の山くんだり...)
- 多層神経回路網のとき、この最急降下法がきれいに書けたことが良かった

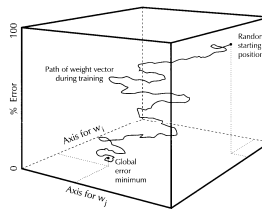


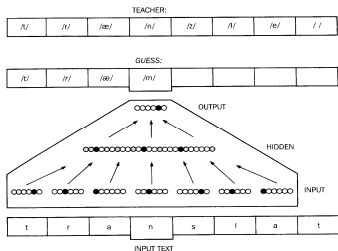
Figure 10.6. Learning by gradient descent in weight/error space. (Axes are shown for only two of the 105 synaptic weights.)

# NETTalk

- NETTalk: 興味深い課題:
  - 多くの利用可能な規則
  - 例外もおおい

Table 1 Symbols for phonemes used in NETTalk.

Symbol	Phoneme	Symbol	Phoneme
/ə/	garbe	/C/	chin
/b/	ket	/D/	dlis
/d/	bought	/E/	bet
/dʒ/	abbe	/F/	fang
/e/	hake	/I/	bit
/f/	fin	/K/	sexual
/g/	guess	/L/	burll
/h/	head	/M/	abow
/i/	Pete	/N/	button
/k/	Acen	/O/	top
/l/	let	/Q/	quest
/m/	met	/R/	bird
/n/	net	/S/	slin
/o/	boat	/T/	thin
/p/	pet	/U/	book
/r/	red	/V/	boak
/s/	st	/X/	excess
/t/	test	/Y/	cate
/tʃ/	late	/Z/	leisure
/v/	vest	/θ/	bat
/w/	wet	/ʃ/	Nasi
/x/	about	/ʒ/	examine
/y/	yet	/ʒ/	one
/z/	soo	/ʒ/	logic
/ʒ/	bite	/ʒ/	bat



# NETTalk

- NETTalk の学習過程で興味深い現象が観測された:
  - まず、子供のように、舌足らずでしゃべる
  - 広い規則を学習する、一般化しすぎだが
  - 最後に、例外も学習するようになる
- 訓練データに対しては、98% の正解率
- 新奇データに対しても、86% の正解率 (cf. DECTalk なら 95% の正解率; 10 年 vs. 一夏だが!)

# NETTalk

- NETTalk が神経生理学的に妥当であるとは、誰も言わない。けれども、もし脳に少しでも似ているなら、どうやって学習したかを少しでも知りたいのが人情であろう
- 実際、様々な統計的手法が開発され、ニューラルネットの荷重と活動度により形成される「表現」が調べられた
- NETTalk については、その一つがクラスタ分析である...

# NETTalk

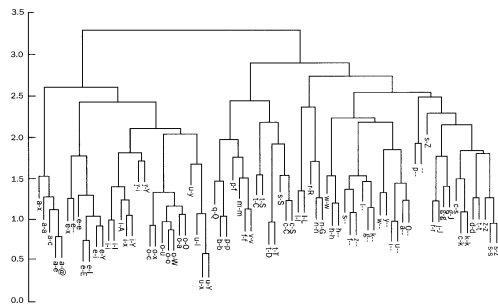


Figure 7.6 Hierarchical clustering of hidden units for letter-to-sound correspondences. The vectors of average hidden unit activity for each correspondence ("l-p" for letter l and phoneme p) were successively merging from right to left in the binary tree. The scale at the top indicates the Euclidean distance between the clusters. (From Sejnowski and Rosenberg 1987.)

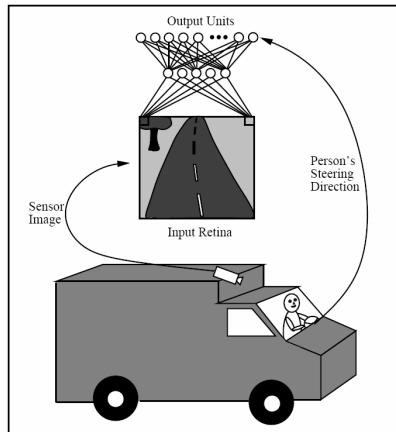
## NETTalk

- NETTalk は、このクラスタ構造を直接教わることはなかった、データから自分で学習したのである
- 学習タスクを再実行するたびに（開始時の初期荷重はランダムに定める）まったく異なる荷重と活動度ベクトルが得られる、しかしクラスタ構造はほぼ同一になる
- 注：ニューラルネットワークが学習するときには、データは規則なり事実なりの形で記憶されることはない。分散した、記号化まえの表現 *sub-symbolic representations* を用いている

## 再掲: 自動運転

### Autonomous Learning Vehicle In a Neural Net (ALVINN): Pomerleau *et al*

Navlab-5 に到り終了 (1995). 高速道路を 70mph で. "No Hands Across America"  
[http://www-2.cs.cmu.edu/afs/cs/user/jochem/www/nhaa/nhaa\\_home\\_page.html](http://www-2.cs.cmu.edu/afs/cs/user/jochem/www/nhaa/nhaa_home_page.html)



## Navlab

### NO HANDS ACROSS AMERICA NAVLAB USA TOUR '95



## ちょっと寄り道

### ■ その後の自動走行研究

2004年 NHK「疾走ロボットカーレース」



2007年 Urban Challenge



[http://en.wikipedia.org/wiki/DARPA\\_Urban\\_Challenge](http://en.wikipedia.org/wiki/DARPA_Urban_Challenge)

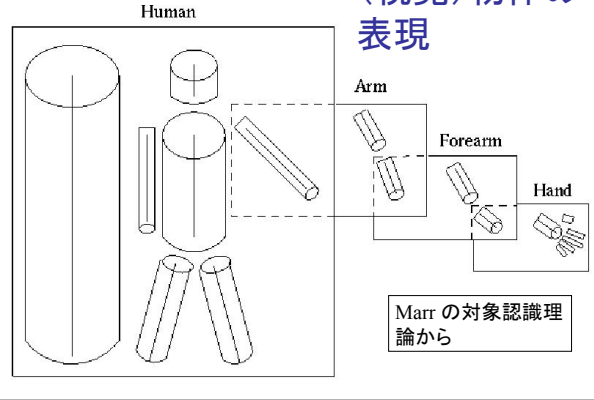
## 心理学と関係あるのか?

- Broadbent (J. Exp. Psych. 114, 189-192, 1985) : 記憶や言語タスクに関する心理学的実験の結果の記述レベルは、ニューラルネットが情報を記憶する方法に関する事実と、根本的に異なる
- 主張:
  - 心理学的研究が発見する事実は *computational* レベル (どんなタスクが行われているか)
  - ニューラルネットが扱っているのは、単に実装レベル *implementational level* であり、心理学については何も言っていない

## Marr の3レベル

- David Marr, Vision, 1982
- 3レベル:
  - 計算論的
  - アルゴリズム的
  - 実装的
- 非常に影響力のある本 (古き良きAI 的ではあるが):
  - 計算論的: どんなタスクをなすべきか?
  - アルゴリズム的: 当該タスクを達成するための、効率的なルールベースの方法は何か?
  - 実装的: どんなハードウェア上で実行すればよいか? (古き良きAIでは、この最後の点はあまり重要ではない, Turing 機械と等価なアーキテクチャならどんなアルゴリズムも実行できる.)

## (視覚) 物体の表現



## 全体・部分階層



S	S
S	S
S	S
SSSSSSSS	S
S	S
S	S
S	S

## 実装は問題になるか?

- Feldman (BBS, 8(2), 265-313, 1985): *The 100-step program constraint* (別名 '100-step rule')
- ニューロンは遅い、一つのニューロンの動作が何であっても、人間の日々のタスクを完遂するには、100個以上のその動作を(直列には)行うだけの時間はない
- ニューロンは超並列を用いて、この問題を解決しているように思える(原理的には可能、例えば、 $\sim 10^{10}$  個のニューロンがあるし、それぞれが  $\sim 10^3$  個程度の(他の素子への)結合を持っている)

## 心理学の対象はどのレベル?

- Rumelhart & McClelland は、心理学的データ(記憶、言語等)が関係するのは:
  - 性能、ノイズや他の悪条件のもとでの性能低下、ある特定の問題は難しいかやさしいか、どの問題が早く解けてどの問題は時間がかかるか、情報はどう表現されているか、等々
- しかし、両者は、ニューラルネットワーク研究も全く同じ問題を対象としていると論じている。少なくとも、ニューラルネットワークと心理学とは同じアルゴリズムレベルの議論をしていると言う事はできる。すなわち、我々が何をやっているかではなく、どうやっているか、という問題である。

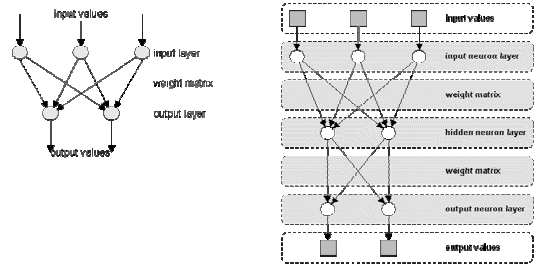
## レベルはいくつあるか?

- Marr の3レベルは単純化しすぎであろう、Rumelhart and McClelland も Churchland and Sejnowski も、複数レベルを考慮する必要があるとしている:
  - 生物化学的
  - 細胞膜
  - 単細胞
  - 神経回路
  - 脳のサブシステム
  - 脳のシステム
  - 脳地図
  - 中枢神経システム全体

## ネットワークの形

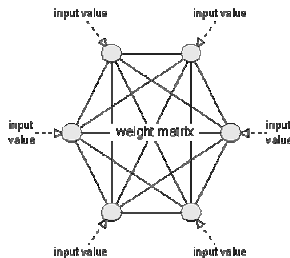
- 階層型(フィードフォワード型)
  - 多層ネットワーク
- 相互結合
  - フィードバック型(ディレイ付き、クロック付き)
  - 純相互結合
- モジュール&その結合

## 階層型(多層パーセプトロン)



## 相互結合 (Hopfield)

- 対称,自己結合なし
- 活動度非同期更新
- 重みの学習なし
- 平衡状態を求める
  - 各素子の活性値(0,1)を全エネルギー減少方向に変化させる
  - あるエネルギー関数の最小値に対応

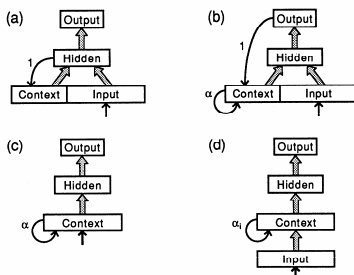


## 相互結合 (Boltzmann machine)

- (基本構造は) Hopfieldと同じ
- ただし、活性値(0,1)の更新は確率的
  - 全エネルギー減少方向へ。増加する確率 $>0$
  - このままでは、永遠に振動し続けるので、
    - 温度パラメータ $T$ を持たせ、
    - $T$ を徐々に変化させることにより、大域的最小値へ収束させる(simulated annealing by Kirkpatrick)
- 学習も考えることができる

## 相互結合 (recurrent)

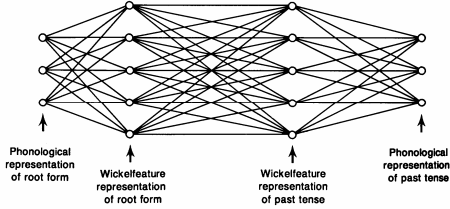
- 階層型+フィードバック(離散時間遅れ)



## 階層型(フィードフォワード型)

- 基本的な機能: 関数近似
  - 入力-出力関係を近似して表現する
  - どんなに変な関数でも(中間素子数を増やせば)必要な精度で近似する方法(結合荷重の組合せ)がある
  - その組合せを見つけることを「学習」という
    - 入力値-出力値の例をたくさん見せて、それらの裏にある(一般的な)入出力関係を「学習」する

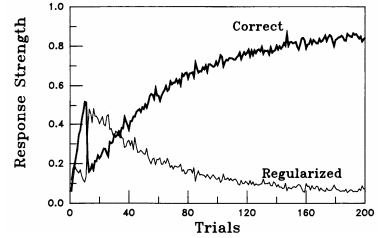
## 英語動詞の過去形の学習



- 英語の規則動詞、不規則動詞の学習を行わせた
  - 入力: 動詞原形、出力: 動詞の過去形
- 音韻的知見に基づく
- 未学習動詞に対しても正しい出力を出す。

Rumelhart et al 『Parallel Distributed Processing』

## 学習中の正解率の変化



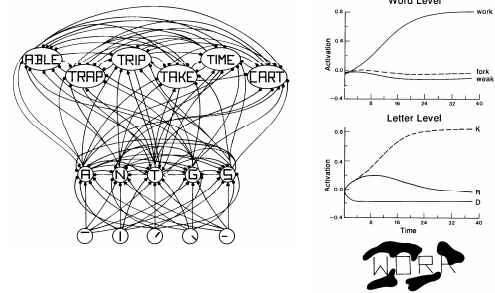
(注)「人間の過一般化とその解消の過程に類似している」との主張は、Chomsky派から極めて強い反論を受けた

Rumelhart et al 『Parallel Distributed Processing』

## 相互結合型 (Hopfield/Boltzmann)

- 基本的な機能: 安定状態の探索
  - どんな初期状態からでも、結合荷重等によって定まるある状態 (安定状態) に移行する
  - 欲しい安定状態を実現する結合荷重が存在する
  - その結合荷重が求まりそうな学習方法がある

## 文脈のモデル



Rumelhart et al 『Parallel Distributed Processing』

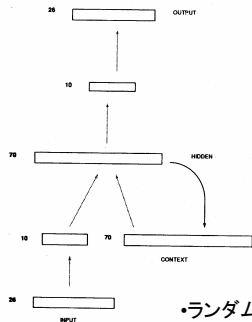
## 組み合わせ最適化に関する注意

- Hopfield ネットワーク等でとける
- しかし、専門のアルゴリズムより遅い
- 本来は、ニューラルネットワークの苦手な計算である。
  - 問題固有の「テクニク」を計算に組み込みたい
- 但し、小さい問題なら「とにかく解ける」という特質はある

## 相互結合型 (recurrent)

- 基本的な機能: 状態機械 (オートマトン)
  - 1 time unit 毎、入力値 (feedback 値も含め) に基づき、次の活性化値を決める
  - 時系列の分類、予測、変換等ができる
    - 不定長入力 (例: 文字列) 上の関数 (分類等) (feedforward では不定長入力は扱えない)
  - 有限状態とは限らない

# 文法の学習 (Elman)



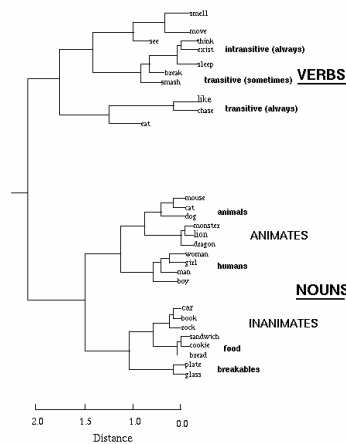
$S \rightarrow NP VP \dots$   
 $NP \rightarrow PropN \mid N \mid N RC$   
 $VP \rightarrow V (NP)$   
 $RC \rightarrow who NP VP \mid who VP (NP)$   
 $N \rightarrow boy \mid girl \mid cat \mid dog \mid boys \mid girls \mid cats \mid dogs$   
 $PropN \rightarrow John \mid Mary$   
 $V \rightarrow chase \mid feed \mid see \mid hear \mid walk \mid live \mid chases \mid$   
 $feeds \mid sees \mid hears \mid walks \mid lives$

- Additional restrictions:**
- number agreement between N & V within clause, and (where appropriate) between head N & subordinate V
  - verb arguments:  
 $chase, feed \rightarrow$  require a direct object  
 $see, hear \rightarrow$  optionally allow a direct object  
 $walk, live \rightarrow$  preclude a direct object

ランダムに10,000文生成し学習させる

J.L. Elman 「Distributed Representations, Simple Recurrent Networks, and Grammatical Structure」

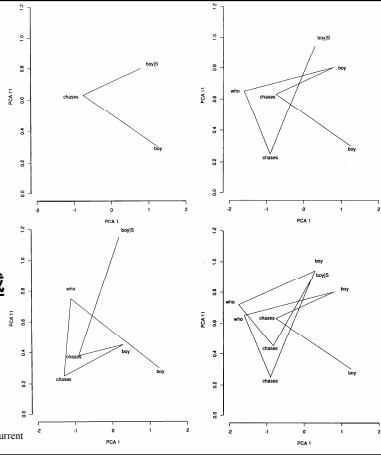
# 単語間の近接性



J.L. Elman 「Language as a dynamical system」

# 埋め込み文表現

- “who chases boy” の表現がどのような文脈でも似ている
- 埋め込み文を越えて agreement がとれている

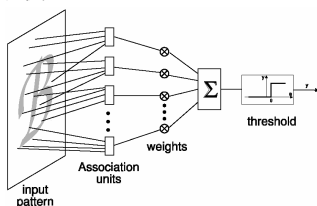


J.L. Elman 「Distributed Representations, Simple Recurrent Networks, and Grammatical Structure」

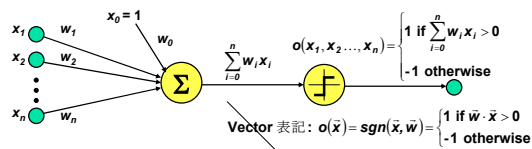
# さて、実装の話に入ろう

# 多義語: パーセプトロン

- パーセプトロン: 同じ言葉で別のものを指している
  - 線型閾値素子: 次のスライド
  - 元祖パーセプトロン: 下記、これが本当!
  - シグモイド素子: 次回
  - シグモイド素子のネットワーク: 多層パーセプトロンと呼ばれる。次回
  - 線型閾値素子のネットワーク: 多層パーセプトロン。稀
- 本講義では、習慣に従い「間違った」用法に従う
- 元祖パーセプトロン
  - Rosenblatt 1962
  - Minsky and Papert 1969

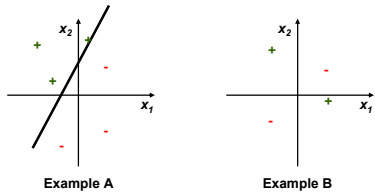


# パーセプトロン Perceptron



- パーセプトロン Perceptron: 単一ニューロンのモデル
  - 別名 線型閾値素子 Linear Threshold Unit (LTU) or Linear Threshold Gate (LTG)
  - 素子への純入力 net input: 線型関数  $net = \sum_{i=0}^n w_i x_i$
  - 素子の出力: 純入力に閾値関数 threshold function を施したもの (閾値 threshold  $\theta = w_0$ )
    - 純入力に施して出力を得る関数を 活性化関数 activation function と呼ぶ
- パーセプトロンネットワーク Perceptron Networks
  - パーセプトロン同士が 荷重つき結合 weights  $w_i$  によって繋がっている
  - Multi-Layer Perceptron (MLP): 下の方

## パーセプトロンの決定境界



- パーセプトロン: 重要な関数がいくつも簡単に表現できる
  - 論理関数 (McCulloch and Pitts, 1943)
  - e.g., 簡単な荷重で  $AND(x_1, x_2)$ ,  $OR(x_1, x_2)$ ,  $NOT(x)$
- 表現できない関数もある
  - e.g., 線型分離可能でないもの
  - 解: パーセプトロンのネットワーク

## パーセプトロン学習アルゴリズム

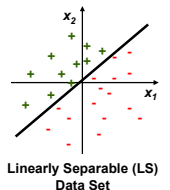
- 学習規則 = 訓練規則 training Rule
  - 教師付き学習に特有の話ではない
  - いつ、どこ: モデルの更新
- Hebbの学習則 Hebbian Learning Rule (Hebb, 1949)
  - アイデア: もし2個の素子が両方とも active ("firing") であれば, 結合荷重は増加する
  - $w_{ij} = w_{ij} + r \cdot o_i \cdot o_j$ , 但し  $r$  は学習係数 learning rate で、定数である
  - 神経生理学的に、ほぼ、支持されている
- パーセプトロン学習アルゴリズム Perceptron Learning Rule (Rosenblatt, 1959)
  - アイデア: 各入力ベクトルに対して出力値が与えられているなら, 荷重を漸進的に更新することにより, 当該出力値が出力できるようにする
  - 2値出力 (Boolean, Boolean-valued) を仮定; 単一パーセプトロン素子
  - $w_i \leftarrow w_i + \Delta w_i$   
 $\Delta w_i = r(t - o_i)x_i$
  - 但し  $t = c(x)$  は目標出力値,  $o$  はパーセプトロンの現在の出力値,  $r$  は学習係数, 正定数であれば何でも良い. 1でよいので、実は、パーセプトロン学習アルゴリズムでは、 $r$  は不要
  - $D$  が線型分離可能 linearly separable であれば, 収束する.  $r$  が十分小さいことを条件とする説明もあるがそれは誤り

## パーセプトロン学習アルゴリズム

- 単純な勾配降下 Gradient Descent アルゴリズムである
  - このアイデアは、適当な表現を用いれば、概念学習にも記号学習にも適用可能
- アルゴリズム Train-Perceptron ( $D \equiv \{<x, t(x) \equiv c(x)>\}$ )
  - 荷重  $w_i$  をランダム値に初期化する // パーセプトロン時は0に初期化してもよい
  - WHILE 正しい出力をしない事例がある DO
  - FOR それぞれの事例  $x \in D$
  - 現在の出力  $o(x)$  を計算
  - FOR  $i = 1$  to  $n$
  - $w_i \leftarrow w_i + r(t - o_i)x_i$  // perceptron learning rule.  $r$  is any positive #
- パーセプトロン学習可能性
  - 復習:  $h \in H$  のときのみ学習可能 - i.e., 線型分離可能 linearly separable (LS) functions
  - Minsky and Papert (1969) Perceptrons: 元祖パーセプトロンの表現・学習の限界を示した
    - ・ 注: 素子一個では parity ( $n$ -変数 XOR:  $x_1 \oplus x_2 \oplus \dots \oplus x_n$ ) 関数が表現できない、というのは既知
    - ・ e.g., 画像の symmetry, connectedness は (元祖パーセプトロンで) 表現できない
    - ・ "Perceptrons" のせいで ANN 研究が10年近く遅れたといわれるも、どこまで真実か。

## 線型分離

- 定義
  - $f(x) = 1$  if  $w_1x_1 + w_2x_2 + \dots + w_nx_n \geq \theta$ , 0 otherwise
  - $\theta$ : 閾値
- 線型分離可能か?
  - 注:  $D$  が線型分離可能だからといって、真の概念  $c(x)$  が線型分離可能とは限らない
  - 選言 disjunction:  $c(x) = x_1' \vee x_2' \vee \dots \vee x_m'$
  - $m$  of  $n$ :  $c(x) =$  at least 3 of  $\{x_1', x_2', \dots, x_m'\}$
  - 排他的 exclusive OR (XOR):  $c(x) = x_1 \oplus x_2$
  - 一般の DNF:  $c(x) = T_1 \vee T_2 \vee \dots \vee T_m$ ;  $T_i = I_i \wedge L_i \wedge \dots \wedge L_k$
- 表現の変換
  - 線型分離可能でない問題を線型分離可能な問題に変換できるか?
  - それは意味のあることなのか? 現実的なのか?
  - 現実問題の重要な部分を占めるのか?



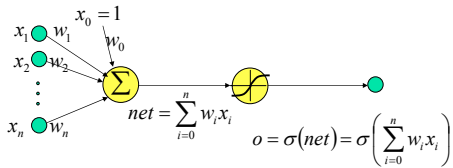
## パーセプトロン学習の収束

- パーセプトロン学習の収束定理
  - 主張: もし訓練データと consistent な荷重集合があれば (i.e., データが線型分離可能なら), パーセプトロン学習アルゴリズムは収束する
  - 証明: 探索空間が境界のある順序をなしている ("模の幅" が厳密に減少していく) - 参照 Minsky and Papert, 11.2-11.3
  - 注意 1: 収束までの平均時間は?
  - 注意 2: もし線型分離可能でなければどうなるのか?
- パーセプトロン循環定理
  - 主張: 訓練データが線型分離可能でなければ パーセプトロン学習アルゴリズムにより得られる荷重ベクトルは、ある有界集合内に留まる. 荷重が整数ベクトルなら、有限集合内に留まる.
  - 証明: もし十分に絶対値が大きい荷重ベクトルから始めると、絶対値は殆ど大きくなれないことが示せる; 訓練事例の次元  $n$  の数学的帰納法による - Minsky and Papert, 11.10
- よりロバストに、またより表現力を上げるには?
  - 目的 1: もっとも良い近似を発見するアルゴリズムの開発
  - 目的 2: 表現の制約を超える新しいアーキテクチャの開発

## 他のモデル

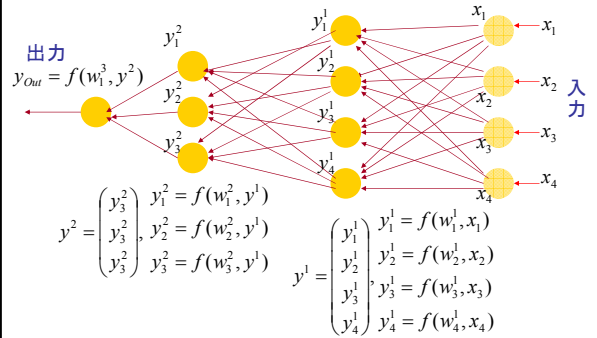
- perceptron も非線形モデルである.
  - 線形関数と階段関数
- 階段関数は微分不能故、扱いにくい
  - 微分可能にできないか?
- フィードフォワードは単純すぎ
  - フィードバックつき
  - 相互結合
- 教師なし学習であるべき
  - 自己組織化型

# シグモイド素子



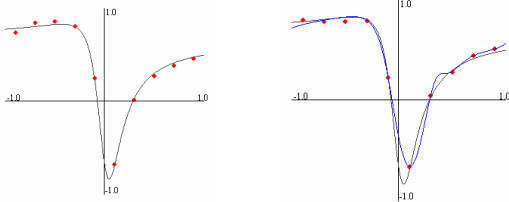
極めて頻繁に使われる  $\sigma$  : 
$$\sigma(x) \equiv \frac{1}{1 + e^{-x}}$$

# 基本的な計算



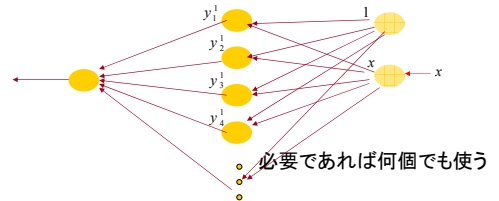
# ニューラルネットの原理

- 一側面: 関数近似・回帰である
  - 例: 1入力・1出力とする



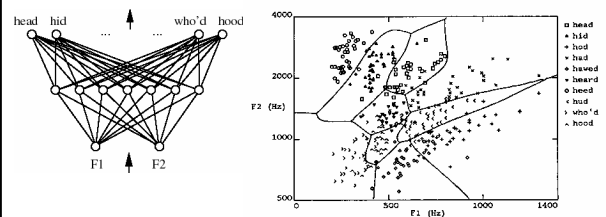
# 近似定理

- ニューラルネットワークの中間素子数を必要なだけ用意できるなら、任意の滑らかな関数を任意の精度で近似することができる


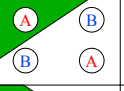



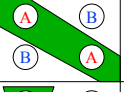



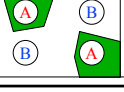




<http://neuron.eng.wayne.edu/bpFunctionApprox/bpFunctionApprox.html>

# 音声認識

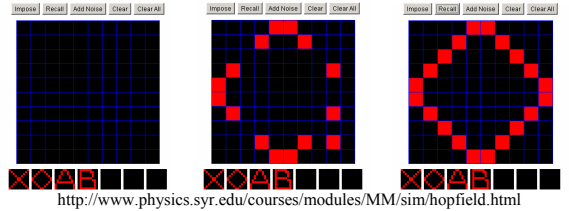


## 領域と境界面

構造	境界面の形	例1 対XOR問題	例2	例3
中間層なし 	超平面			
中間層2素子 	2超平面、それらを滑らかにしたもの			
中間層多素子 	任意(但し、素子数に依存)			

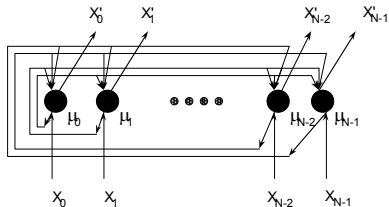
## 連想

- (A, a), (B, b), (C, c), ... というデータを記憶し、(A, ?) と聞かれたら ?=a と答える
- 変形した・ノイズの乗った図形から元の図を復元する。

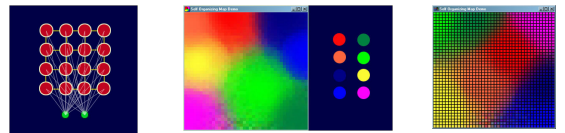


## Hopfieldネットワーク

- 相互結合型。通常、時を刻みながら、過去の自分達の値を入力として、次の出力(これが次の入力となる)を決める。



## SOM Self-Organizing Maps

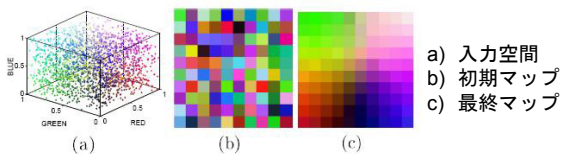


- どの素子(左図の赤)も、入力素子(左図の緑)と同様に接続、相互接続はない。
- RGB値を入力する(だから左図は不適當。入力は3次元)。頻度は中央右側の色のみ。SOMは同じ色を集めようとする。しかし、3次元空間を2次元に持つてくるから少し無理はある。
- 右図は別の実験。各桁は一個の素子に対応する。

<http://www.ai-junkie.com/ann/som/som1.html>

## 別名 Kohonen マップ

- (同様に)2次元のマップ。
- 入力は3次元座標。RGBに対応させて表示。
- 初期値(3次元→2次元の対応)はランダム。



## Kohonenマップ

- ワイヤフレームによるデモ

<http://rfrhs8012.fh-regensburg.de/~saj39122/jfroehl/diplom/e-sample.html>