

## 知的情報処理

### 8. ニューラルネットワーク

理工学部管理工学科  
櫻井彰人

## 何か?

- "... 人間の脳で使われていると思われる原理を用いて、精妙に作られたシステム。” — James Anderson
- "... ニューラルネットワークは、単純な処理素子を多数結合して作ったシステムである。その機能は、ネットワークの構造、結合の強さ、各素子での機能に依存して決まる。” — DARPA Neural Network Study (1988)
- “ニューラルネットワークは、単純な素子を相互結合したものである。各素子の機能は、動物の神経素子をまねたものであり、ネットワークとしての機能は素子間の結合強度に依存する。結合強度は、学習データのパターンに適応する、または、それを学習することにより、定められる。” — Kelvin Gurney

## なぜか?

- 知的な機械を作りたいから(今までもそうであったし、これからも、恐らく、ず〜っとそうであろう) — 西ではゴーレム(ドラクエにも出てくるそう。大魔神もゴーレムをもにしているとか)、江戸時代のからくり人形、鉄腕アトム、現代のロボット等々)
- しかも、コンピュータの能力も急速に発展し、脳に近づいている(かもしれない)
  - ただし、脳を計算機(の大親分)にたとえるのは間違いかもしれない。
  - その昔、時計がその時代の最も複雑な機械であったころ、脳は時計をたとえに説明された
- 本物の脳が用いるニューロンの速度は遅い。現在のCPUと比べ  $1/10^5 \sim 1/10^6$  .
  - それにも関わらず、現在のコンピュータより遥かに優秀。なぜだ?
- だったら、脳の真似をしてみよう。(人工)ニューラルネットワークと名づけて。
  - 単純素子の超並列結合で試してみよう

## 工学的だけではない

- “ニューラルネットワークは実用的な価値があると同時に、人間行動のモデルとしての価値もある。” -- Anderson

## 目次

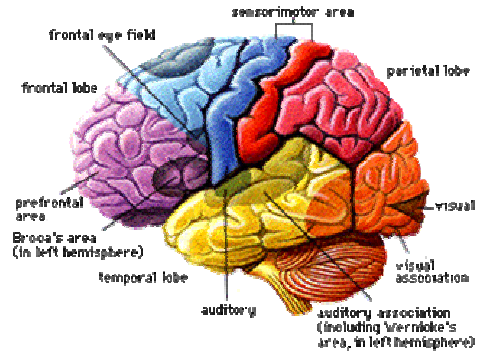
- 脳と神経回路網
- 神経素子・ネットワークのモデル
  - 多数素子の超並列実行とロバストネス
  - 学習
- 学習アルゴリズム
- 応用
  - テーマ別: 音声認識、自動運転、画像認識
  - 技術別: 関数近似、分類、連想、次元低減
- 中間層表現
- さまざまなニューラルネットワーク



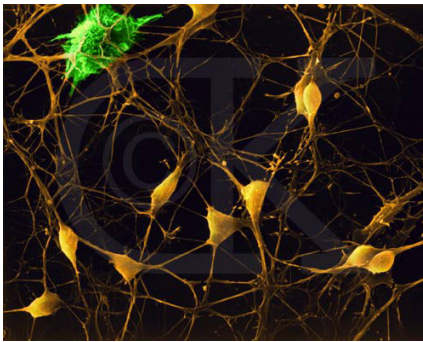
<http://www.willamette.edu/~gorr/classes/cs449/figs/oldbrain.gif>



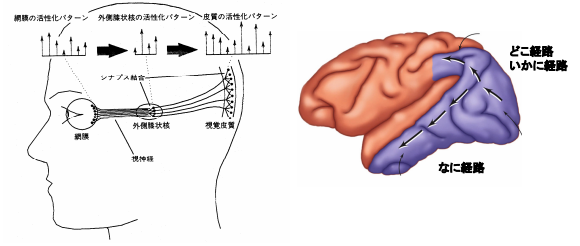
<http://www.willamette.edu/~gorr/classes/cs449/figs/brain2.jpg>



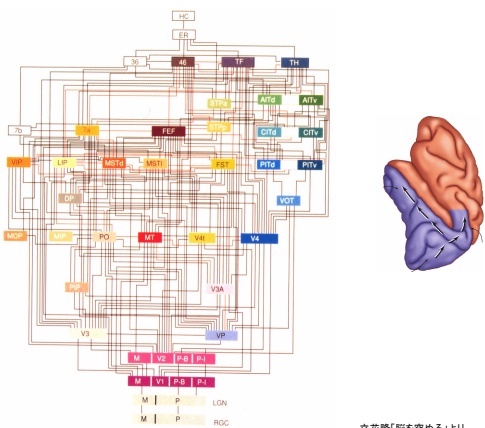
<http://www.emc.maricopa.edu/faculty/farabee/BIOBK/brain.gif>



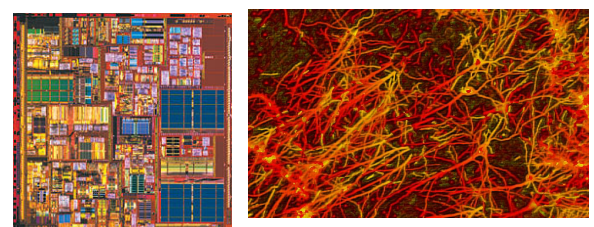
<http://www.emc.maricopa.edu/faculty/farabee/BIOBK/08705a.jpg>  
copyright: Dennis Kunkel at <http://www.DennisKunkel.com>



チャーチランド「認知哲学」から



立花隆「脳を究める」より

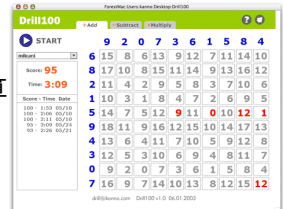


<http://www.willamette.edu/~gorr/classes/cs449/figs/neurons.jpg>  
[http://www.intel.com/pressroom/archive/photos/p4\\_photos.htm](http://www.intel.com/pressroom/archive/photos/p4_photos.htm)

	処理素子	素子サイズ	使用エネルギー	処理速度	計算のスタイル	耐故障性	学習	知能・意識
	10 <sup>14</sup> シナプス	10 <sup>-6</sup> m	30W	100Hz	並列分散	あり	あり	通常はあり
	10 <sup>9</sup> トランジスタ	10 <sup>-6</sup> m	30W	10 <sup>9</sup> Hz	逐次集中	なし	少し	なし(今のところは)

## 能力面でも違いが

- 人間なら2分～3分
  - 1秒前後/(演算し書く)
- コンピュータなら？
  - 10<sup>-9</sup>秒前後/演算
- もっと大変な数値計算なら、もっと大きな差がでる



<http://drill100.com/images/Drill100.gif>

## 能力面でも違いが

国境の長いトンネルを抜けると、雪国であった。(川端康成)

When the tunnel where the border is long is passed through, it was snow country. (Google, Babel fish)

It was a snowy country when escaped from the long tunnel of the border. (フレッシュアイ)

When the train came out of a long tunnel at the border, it was a snowy country. (エキサイト)

The train came out of the long tunnel into the snow country. (Edward G. Seidensticker)

夜の底が白くなった。信号所に汽車が止まった。  
向側の座席から娘が立って来て、島村の前のガラス窓を落した。  
雪の冷気が流れこんだ。  
娘は窓いっぱいになり出して、遠くへ叫ぶやうに、「駅長さあん、駅長さあん。」

The bottom of the night whitened. The train stopped in the signal box.  
The daughter dropped the glass window in front of Shimamura standing from the seat on the side for.  
The cold of the snow flowed.  
The daughter starts a full window, and ..distance... ..shout...  
(The daughter : as start a full window, and shouted to the distance.)  
"Length bean jam of station and length bean jam of station"  
("Stationmaster and stationmaster") (エキサイト翻訳)

The earth lay white under the night sky. The train pulled up at a signal stop.  
A girl who had been sitting on the other side of the car came over and opened the window in front of Shimamura.  
The snowy cold poured in.  
Leaning far out the window, the girl called to the station master as though he were a great distance away. (Edward G. Seidensticker)

蛇足:

英文には主語が必要だが、この文にはない。長いトンネルを抜けたのは、ネズミですか、それともゴキブリですか？長いトンネルとは、ここでは清水トンネルなので汽車を主語にしました。

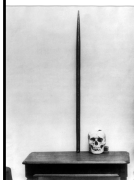
(略) 越後、信濃の国境だからと、The border of provinces of Echigo and Shinano などとしては英語のリズムが崩れます。

文学作品の翻訳にあたっては、読んで耳に心地よく響く音、euphonious であることも心がけるべきです。

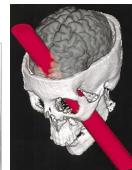
(サイデンステッカー氏講演会の質疑応答から)

[http://www.tek.co.jp/OGAWA/story\\_01.html](http://www.tek.co.jp/OGAWA/story_01.html)

## Phineas T. Gage



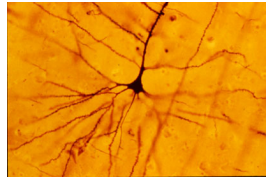
An Odd Kind of Fame: Stories of Phineas Gage



1848年9月13日、鉄道建設作業の現場監督 Phineas Gage 25歳はアメリカ合衆国 Vermont 州の小さな町 Cavendish の近くで、岩盤を爆破する仕事をしていた。この日、仕掛けたダイナマイトが爆発しないため鉄棒でつついた。その瞬間にダイナマイトは爆発し、長さ 109 cm、太さ 3 cm、重さ 6 kg の鉄棒は彼の下部から頭を貫通し敵の後方へ30メートル近く飛んだ。事故後も彼は意識があり、仲間に支えられて歩くことができた。彼は Cavendish の医師 Dr. Harlow の治療を受け約 10 週間を退院した。

New Hampshire の自宅で7ヶ月間静養した Gage は鉄道工事の職場に復帰するが、最早以前のような指導的役割を果たすことができなかった。事故以前には彼は有能で効率的な現場監督であり、精神的にもバランスがとれていた。事故後仕事に復帰した彼は、きまぐれで、非礼で、下品になり、彼の仲間に敬慕をほとんど示さなかった。また、辛抱強さを失い、頑固になり、そのくせ、移り気で、優柔不断で、将来の行動のプランもきちんと決めることができなかった。彼の友人は、彼のことを「彼は最早、Gage ではない」と評した。

<http://www.pri.kyoto-u.ac.jp/brain/brain/41/index-41.html>



ゴルジ法による染色した錐体細胞

Santiago Ramón y Cajal による  
プルキンエ細胞のスケッチ

<http://en.wikipedia.org/wiki/Dendrite>

## 能力の違い

- 単純計算は猛烈に**速い**
- 大量な計算でも**速い**
- **正確**、信頼性抜群
- 多くの情報の**統合は苦手** or できない
- 一部が壊れると全体が機能しない
- **プログラム**するしかない
- 単純な計算は**遅い**
- 大量計算なんて**できない**
- **不正確**、信頼性低い
- 多くの情報の**統合は得意**
- (壊れ方にもよるが)かなり壊れても大丈夫。
- プログラムはできないが、**自律学習**する

(外界との交流がない等の人工知能自体の課題もある)

## そこで

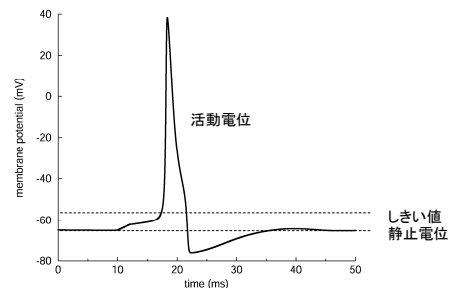
- 脳の基本原理(わかっていないのだが)を真似てコンピュータを作れば、脳のように賢くロバストかつ学習するコンピュータができるではないか？
- ポイント(と考えたこと)は
  - 単純要素をたくさんつなぎ、**超並列**動作
  - 結合の仕方、結合度合いを**適応的**に
    - 演算と同様に、結線が大切
  - データは**分散**して記憶
    - 少々壊れても推測可能

## そこでモデル化

- 細胞一個のモデル化
  - 形態の観測、
  - 動作の測定等
- ネットワークのモデル化
  - 接続形態の観測、
  - 変化(学習)の測定等

## 目次

- 脳と神経回路網
- 神経素子・ネットワークのモデル
  - 多数素子の**超並列実行**と**ロバストネス**
  - **学習**
- 学習アルゴリズム
- 応用
  - テーマ別: 音声認識、自動運転、画像認識
  - 技術別: 関数近似、分類、連想、次元低減
- 中間層表現
- さまざまなニューラルネットワーク



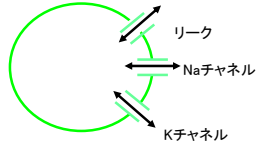
## Hodgkin-Huxley方程式

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

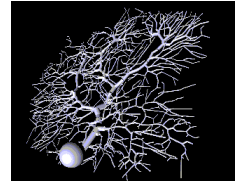
$$\frac{dm}{dt} = \alpha_m(V)(1-m) - \beta_m(V)m$$

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

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



V: 膜電位, m, h(Na), n(K): チャンネルが開く確率



GENESISによるモデル

Santiago Ramón y Cajal による  
プルキンエ細胞のスケッチ

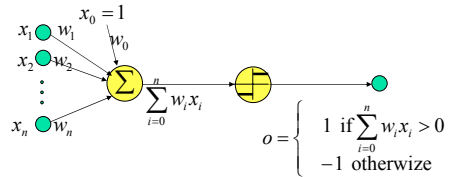
<http://www.genesis-sim.org/GENESIS/cnsls/purkcell.gif>

<http://en.wikipedia.org/wiki/Dendrite>

## 簡略化

- 思い切って簡略化
- 本質は、
  - 多くの入力
  - 閾値関数的動作
    - 「閾値」以下なら出力は何も出さない
    - 「閾値」以上になると突然動作
    - 動作したあとと暫くお休み
  - それを大量につなぐ

## McCulloch-Pitts モデル(1943)



## 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.

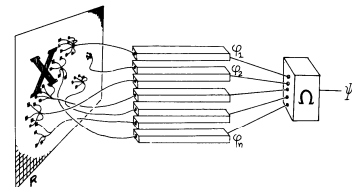
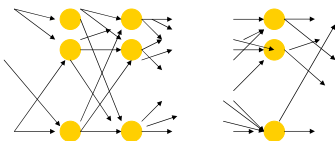
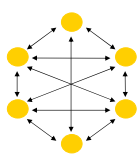


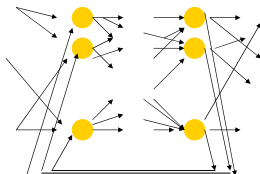
FIGURE 1. The one-layer perceptron analyzed by Minsky and Papert. (From *Perceptrons* by M. L. Minsky and S. Papert, 1969, Cambridge, MA: MIT Press. Copyright 1969 by MIT Press. Reprinted by permission.)



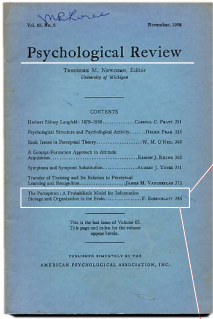
階層型(フィードフォワード)



相互結合



再帰型(リカレント)



ROSENBLATT, Frank. *The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain*. In, *Psychological Review*, Vol. 65, No. 6, pp. 386-408, November, 1958. Lancaster, PA and Washington, DC: American Psychological Association, 1958. Octavo, original blue wrappers; custom box. \$2300.

<http://www.manhattanrarebooks-science.com/rosenblatt.htm>

The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain..... F. ROSENBLATT 386

## できること

- 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
- 実は、学習ができることがポイント

## 目次

- 脳と神経回路網
- 神経素子・ネットワークのモデル
  - 多数素子の超並列実行とロバストネス
  - 学習
- 学習アルゴリズム
- 応用
  - テーマ別: 音声認識、自動運転、画像認識
  - 技術別: 関数近似、分類、連想、次元低減
- 中間層表現
- さまざまなニューラルネットワーク

## 学習とは

- システムが、試行錯誤を通してより賢くなること
- 「こういう状況で、こうしなさい」と教えられると、次の“こういう状況”では“こう”する可能性が高くなり、最終的には“こう”するようになる
- <状況を表すベクトル, 出力値> がたくさん与えられると、“状況を表すベクトル”が与えられると、対応する“出力値”が出せるようになる
- $\{ \langle x_i, o_i \rangle \}_{i=1, \dots, k}$  が与えられると、 $f(x_i) = o_i$  となる関数(ネットワーク)を作ること

## パーセプトロン Perceptron

- 下図の  $\Omega$  は、特徴抽出器の出力値の荷重付き線形和に閾値関数を施す
- 学習は、出力値が望みのものになるように、荷重を変更する

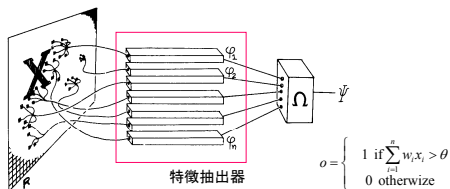


FIGURE 1. The one-layer perceptron analyzed by Minsky and Papert. (From *Perceptrons* by M. L. Minsky and S. Papert, 1969. Cambridge, MA: MIT Press. Copyright 1969 by MIT Press. Reprinted by permission.)

## 閾値は荷重で

$$\sum_{i=1}^n w_i x_i > \theta$$

$$\sum_{i=1}^n w_i x_i - \theta > 0$$

$$\sum_{i=0}^n w_i x_i > 0 \quad w_0 = -\theta \quad x_0 = 1$$

## Rosenblatt の学習アルゴリズム

初期化:  $\vec{w}$  は、任意の  $\vec{x} \in F = F^+ \cup F^-$  とする

Repeat

順番に  $\vec{x} \in F$  を選ぶ

If  $\vec{w} \cdot \vec{x} > 0$  and  $\vec{x} \in F^+$  then continue;

If  $\vec{w} \cdot \vec{x} \leq 0$  and  $\vec{x} \in F^+$  then FixPlus して continue;

If  $\vec{w} \cdot \vec{x} \leq 0$  and  $\vec{x} \in F^-$  then continue;

If  $\vec{w} \cdot \vec{x} > 0$  and  $\vec{x} \in F^-$  then FixMinus して continue;

until 間違えなくなる (FixPlusもFixMinusも呼ばなくなる)

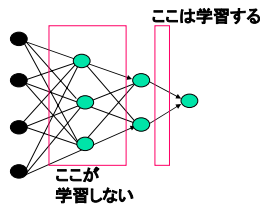
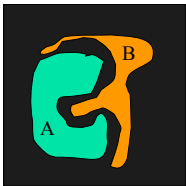
FixPlus:  $\vec{w} := \vec{w} + \vec{x}$

FixMinus:  $\vec{w} := \vec{w} - \vec{x}$

## 限界

- 何ができないか? 実は表現はできるが学習ができない
  - 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?':
  - 学習させる方法を知らない
  - 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

## 課題は



結果(出力値)が間違っているとき、結合荷重を変更すべきである。すべての荷重を変更して構わないが、問題はその変更量を決める方法が分からない

## 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 (Amari 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。しかし、この分野を再発見させたことは大きな成果である。]

## PDPの成功理由

- 素子の出力関数(閾値関数)を微分可能な関数(sigmoid関数)にかえた
- 学習問題を誤差最小化問題に変換した

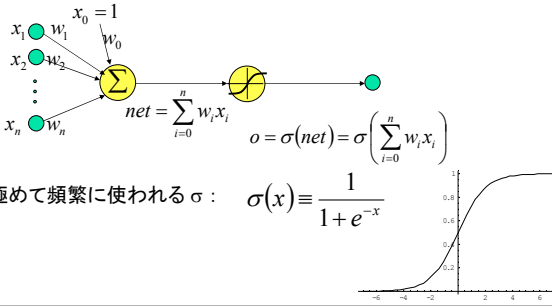
$$E(w) = \sum_{\text{すべてのデータ } x_k} (f(x_k; w) - x_k \text{の目標値})^2$$

- (非線形・多変数の)誤差最小化問題を極めて naive な方法でといた(最急降下法)

## 誤差の最小化

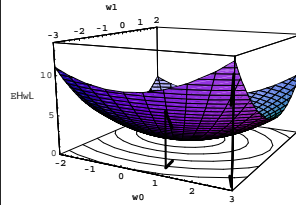
- 完全(誤差=0)を求めてはいけない
  - 我々の能力には限りがある
  - データにも誤りがあるかもしれない
  - そもそも、データの発生は確率的現象かもしれない
- (安易ではあるが)目標値と実際値の差の2乗を、全データについて足したものを誤差と考えよう
- それを最小化する荷重を決めればよい!

# シグモイド素子



# (反復)最小化法

- (最小化には様々な方法が提案されているが)
- 中でも最も単純なものが、最急降下法
  - 最大値を求めるなら、最急上昇法(あまり使わない)



最急降下方向と等高線のなす角度に注目!

# ところで、なぜ反復?

- 微分して0とおいた方程式を解けばよい! 本当か?
 
$$E(w) = \sum_{\text{すべてのデータ } x_k} (f(x_k; w) - x_k \text{の目標値})^2$$

$$\frac{\partial E}{\partial w} = 0$$
- $f$  は非線形関数ゆえ、この方程式は非線形連立方程式となる。到底、解けない
- 反復解法(少しずつ、解を改善していく方法)を考える。すなわち、 $E(w_1) > E(w_2) > E(w_3) > \dots$  となる  $w_1, w_2, w_3, \dots$  を求める方法を考える

# 誤差逆伝播法の導出



定義:

- $a_i$  = 素子  $i$  の活動度 (activation)
- $w_{ij}$  = 素子  $j$  から素子  $i$  への結合荷重
- $x_i$  = 素子  $i$  への荷重入力 (興奮度 excitation 閾値関数を適用する前の値)
- $t_i$  = 目標値
- $o_i$  = 出力素子  $i$  の活動度 ( $a_i$  と同じ)

定義により:

$$x_i = \sum_j w_{ij} a_j$$

$$a_i = 1 / (1 + e^{-x_i})$$

$$\text{出力層での誤差: } E = 1/2 \sum_i (t_i - o_i)^2$$

# 誤差逆伝播法の導出

合成関数の微分公式:  $\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_i} \frac{\partial o_i}{\partial x_i} \frac{\partial x_i}{\partial w_{ij}}$

$$\frac{\partial E}{\partial o_i} = \frac{1}{2} 2(t_i - o_i)(-1) = o_i - t_i \quad E = \frac{1}{2} \sum_i (t_i - o_i)^2$$

$$\frac{\partial o_i}{\partial x_i} = -\frac{\partial}{\partial x_i} \frac{1}{1+e^{-x_i}} = \frac{e^{-x_i}}{(1+e^{-x_i})^2}$$

$$= \frac{(1+e^{-x_i})-1}{1+e^{-x_i}} \cdot \frac{1}{1+e^{-x_i}} = \left(1 - \frac{1}{1+e^{-x_i}}\right) \frac{1}{1+e^{-x_i}} = (1-o_i)o_i$$

$$\frac{\partial x_i}{\partial w_{ij}} = a_j \quad x_i = \sum_j w_{ij} a_j$$

# 誤差逆伝播法の導出

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_i} \frac{\partial o_i}{\partial x_i} \frac{\partial x_i}{\partial w_{ij}} = (o_i - t_i) (1 - o_i) o_i a_j$$

誤差そのもの

入力活動度の影響

シグモイド関数の性質 (一般にはPfeff関数の性質)

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \quad (\text{ただし } \eta \text{ はある学習係数})$$

$$w_{ij}^{t+1} = w_{ij}^t + \eta \underbrace{e_i}_{(1-o_i)o_i} a_j$$

## 誤差逆伝播法の導出

さて、次には、隠れ層の結合荷重の変更量を求める必要がある。前と同様に、誤差関数のある特定の荷重で微分してみよう

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial a_i} \frac{\partial a_i}{\partial x_i} \frac{\partial x_i}{\partial w_{ij}}$$

(ここで、i は、今度は、隠れ層内の素子に、j は入力層、または、より早期の隠れ層の素子に対応する)

前と同様に、最後の2項は、単に:

$$\frac{\partial a_i}{\partial x_i} = (1 - a_i) a_i$$

$$\frac{\partial x_i}{\partial w_{ij}} = a_j$$

## 誤差逆伝播法の導出

しかしながら、最初の項を隠れ層に即して解釈するのはちょっと難しい。これは Minsky が *credit assignment problem* と呼んだものであり(今では普通であるが) 20 年間に渡って研究を遅滞させたものである。ポイントは、隠れ層の素子は誤差を生成するというよりは、出力素子の誤差に寄与すると考えるべきことである。したがって、総誤差を隠れ層素子の活動度で微分した値は、当該素子が全出力素子の誤差和に対する寄与の総和である:

$$\frac{\partial E}{\partial a_i} = \sum_k \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial x_k} \frac{\partial x_k}{\partial a_i} \quad (\text{ただし } k \text{ は出力素子をさす})$$

各出力素子の寄与      出力素子への入力全部(隠れ層からの入力)の寄与      隠れ層のある特定の素子の寄与

## 誤差逆伝播法の導出

以前と同様に、最初の2項は簡単に求まる:

$$\frac{\partial E}{\partial o_k} = o_k - t_k$$

$$\frac{\partial o_k}{\partial x_k} = (1 - o_k) o_k$$

第3項については:

$$x_k = \sum_j w_{kj} a_j$$

であって、 $a_i$ を含む(和の中の)項は1項だけであるから:

$$\frac{\partial x_k}{\partial a_i} = w_{ki}$$

## 誤差逆伝播法の導出

これらの項をあわせれば:

$$\frac{\partial E}{\partial a_i} = - \sum_k (t_k - o_k) (1 - o_k) o_k w_{ki}$$

隠れ層と出力層の間の結合荷重

$$\frac{\partial E}{\partial w_{ij}} = - \left( \sum_k d_k w_{ki} \right) (1 - a_i) a_i a_j$$

$$w_{ij}^{t+1} = w_{ij}^t + \eta \left( \sum_k d_k w_{ki} \right) (1 - a_i) a_i a_j$$

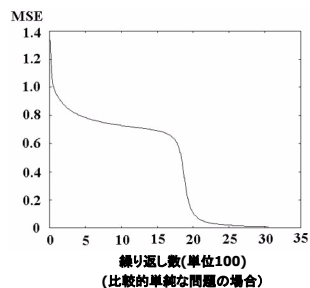
$d_i$

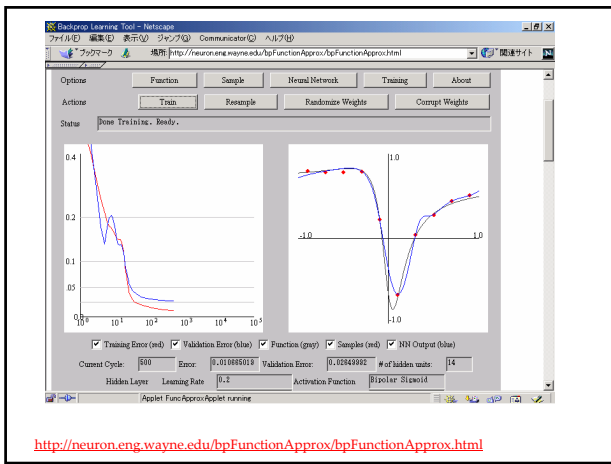
## 誤差逆伝播法の導出

隠れ層が複数になった場合には、同じパターンで計算ができる。すなわち、上の層(出力に近い層)から  $d$  と  $w$  とを求め、更新する荷重の上下から  $a$  を求める。

これが、誤差逆伝播学習アルゴリズムとなる

## 誤差の減少の様子





## というわけで Error 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...

## 目次

- 脳と神経回路網
- 神経素子・ネットワークのモデル
  - 多数素子の超並列実行とロバストネス
  - 学習
- 学習アルゴリズム
- 応用
  - テーマ別: 音声認識、自動運転、画像認識
  - 技術別: 関数近似、分類、連想、次元低減
- 中間層表現
- さまざまなニューラルネットワーク

## 何ができるか?

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

## 音声認識

<http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-3/www/ml.html>

## ALVINN: 自動運転

- 高速道路で米国横断 (Dean Pomerleau 1995)

<http://web.mit.edu/6.034/www/bob1.9/node10.html>

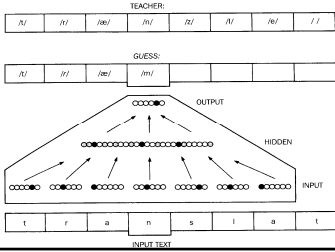
# NETTalk

## NETTalk の興味深い課題:

- 多くの利用可能な規則
- 例外が多い

Table 1 Symbols for phonemes used in NETTalk.

Symbol	Phoneme	Symbol	Phoneme
/a/	father	/C/	chin
/b/	bet	/D/	ditik
/c/	bought	/E/	bet
/d/	bake	/G/	ging
/e/	bed	/I/	bit
/f/	fin	/K/	kit
/g/	guess	/L/	batle
/h/	head	/M/	abow
/i/	Pete	/N/	button
/j/	Ken	/O/	boy
/k/	let	/Q/	quest
/m/	met	/R/	ford
/n/	net	/S/	shin
/o/	boat	/T/	thin
/p/	pet	/U/	book
/r/	red	/V/	boot
/s/	sit	/X/	excess
/t/	test	/Y/	carte
/u/	lure	/Z/	leisure
/v/	test	/@/	bat
/w/	wet	/_/	Nazi
/x/	about	/^/	examine
/y/	yxt	/`/	one
/z/	zoo	/~/	logic
/_/	bate	/~/	bat



## 学習できるだけではなく

- 面白いことがわかった。それは、
- 中間層には、プログラマが意図しなかった内部表現が発生することがある
- よくよく見ると「意味深い」表現であったりする
- 実は、「情報の圧縮」すなわち、「意味抽出」が行われうることを示せる

## 目次

- 脳と神経回路網
- 神経素子・ネットワークのモデル
  - 多数素子の超並列実行とロバストネス
  - 学習
- 学習アルゴリズム
- 応用
  - テーマ別: 音声認識、自動運転、画像認識
  - 技術別: 関数近似、分類、連想、次元低減
- 中間層表現
- さまざまなニューラルネットワーク

## NETTalk

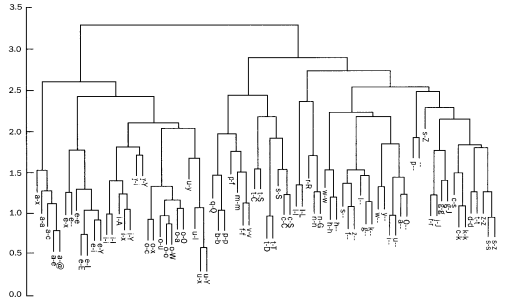
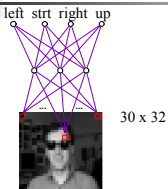


Figure 7.6 Hierarchical clustering of hidden units for letter-to-sound correspondences. The vectors of average hidden unit activity for each correspondence ("l-p" for letter l and phoneme p) were successively merging from right to left in the binary tree. The scale at the top indicates the Euclidean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

## 顔画像の認識(向きの判定)

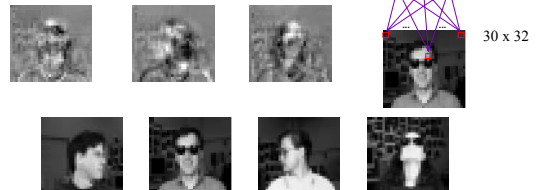


### ■ 顔画像の例



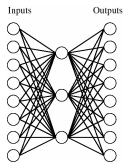
## 顔画像学習の結果

### ■ 学習後の中間層への荷重



## 中間層表現 – 簡単な実験

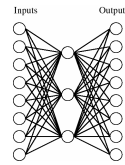
- これは学習できるか？



Input	Output
10000000	→ 10000000
01000000	→ 01000000
00100000	→ 00100000
00010000	→ 00010000
00001000	→ 00001000
00000100	→ 00000100
00000010	→ 00000010
00000001	→ 00000001

## 中間層表現(2)

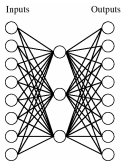
- 学習結果



Input	Output
10000000	→ .89 .04 .08 → 10000000
01000000	→ .01 .11 .88 → 01000000
00100000	→ .01 .97 .27 → 00100000
00010000	→ .99 .97 .71 → 00010000
00001000	→ .03 .05 .02 → 00001000
00000100	→ .22 .99 .99 → 00000100
00000010	→ .80 .01 .98 → 00000010
00000001	→ .60 .94 .01 → 00000001

## 中間層表現(3)

- 学習結果



Input	Output
10000000	→ .89 .04 .08 → 10000000
01000000	→ .01 .11 .88 → 01000000
00100000	→ .01 .97 .27 → 00100000
00010000	→ .99 .97 .71 → 00010000
00001000	→ .03 .05 .02 → 00001000
00000100	→ .22 .99 .99 → 00000100
00000010	→ .80 .01 .98 → 00000010
00000001	→ .60 .94 .01 → 00000001

## これで分かったこと

- 分散表現が実現できた。けれども
  - ロバストか？
  - 少々の破壊(荷重の変更)には yes
  - 大きな破壊には再学習が必要
- 学習ができた。
- (最新ハードウェアで実装すれば)速い

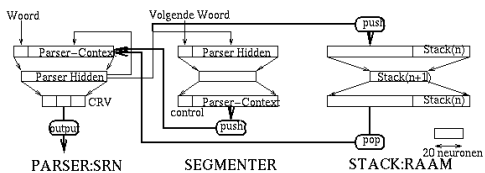
## 問題は残る

- 学習が遅い
  - 特にスケールラブルでない
  - つまり問題が大きくなるとかなり遅くなる
- 従って大きな問題に適用できない
  - 最初の夢(脳の実現!)は夢のまま
  - ニューラルネットワークだけで大きなシステムを作ることができない
- しかし、工夫がある
  - 例えば、高速化方法
    - テクニックではなく、本質的な方法
  - 例えば、組み合わせ
    - 人間の脳だって、機能モジュールの組み合わせ！
  - 例えば、support vector machine
- 問題があるにせよ、(機械学習の手法としては)素性の分からない問題をまず解いてみるには、よい方法である

## 目次

- 脳と神経回路網
- 神経素子・ネットワークのモデル
  - 多数素子の超並列実行とロバストネス
  - 学習
- 学習アルゴリズム
- 応用
  - テーマ別: 音声認識、自動運転、画像認識
  - 技術別: 関数近似、分類、連想、次元低減
- 中間層表現
- **さまざまなニューラルネットワーク**

## Subsymbolic processing



## Support Vector Machine

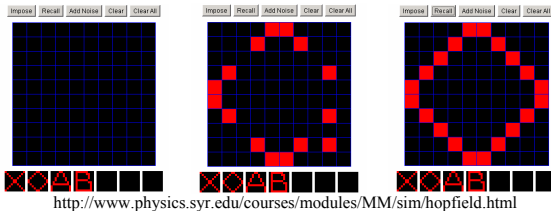
- (直感的に説明すると) SVMは
- パーセプトロンみたいなもの。ただし、
- 入力ベクトルを高次元空間に写像する
  - 策もなくやると後が大変だし、大嘘になるので、kernel 関数なるものを用いる(誤差関数も重要)。
- 線形分離可能に(近く)する
- (パーセプトロンのアルゴリズムを適用したいところだが) 2次最適化法を用いる
  - 誤差関数が、実は、秀逸
- 自然言語処理にはなくてはならない道具
  - 入力次元が非常に高い。これに対処できるのは SVM だけか？

## その他のニューラルネットワーク

- 何種類もある
  - 次に2種類を紹介
- リカレントネットワーク
  - 時系列予測
  - 文法の学習
- パルス・ニューラルネットワーク
  - 神経らしく、パルスで動く
  - 10年前はコンピュータが遅くて研究できなかった
- カオスネットワーク
  - 実は、本物のニューロンは、これかも
- Liquid state machine
  - カオス系の予測精度が高い。

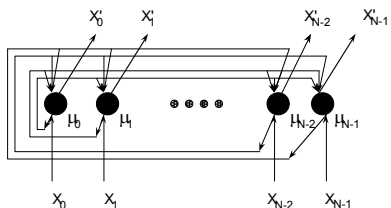
## 連想

- (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。



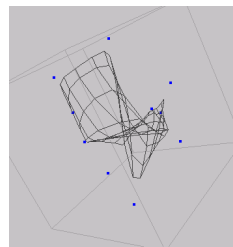
## Hopfieldネットワーク

- 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。

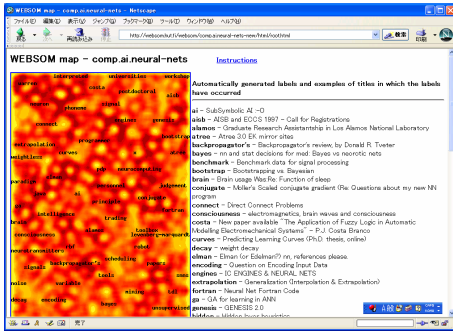


## Kohonenマップ

- 自己組織化マップ self organizing map SOM
- 高次元(少なくとも3次元)内に散らばる点を、できるだけ滑らかに2次元曲面でつなぐ
  - (2 or 高次元内の)点を曲線で滑らかにつなぐ場合も
  - 多数の点が対象の場合には、頻度情報が反映される。少数の点の場合には、点間距離が反映される
- クラスタリングに使える
- 文書分類にも



# Websom



<http://websom.hut.fi/websom/milliondemo/html/root.html>

## まとめ

- 脳と神経回路網
- 神経素子・ネットワークのモデル
  - 多数素子の超並列実行とロバストネス
  - 学習
- 学習アルゴリズム
- 応用
  - テーマ別: 音声認識、自動運転、画像認識
  - 技術別: 関数近似、分類、連想、次元低減
- 中間層表現
- さまざまなニューラルネットワーク