

入門神経回路網モデル

浅川伸一 <asakawa@twcu.ac.jp>

目次

1	神経回路網モデルとは何か?	1
2	神経回路網モデルの分類	2
2.1	学習方式による分類	2
2.2	結合方式による分類	2
2.3	ANN と BNN	3
3	ニューロンの生物学的基礎	3
3.1	神経細胞 (neuron)	3
3.2	神経細胞の結合様式	4
4	単一ニューロンの動作方程式	5
4.1	マッカロック・ピッツの形式ニューロン	6
4.2	マッカロック・ピッツのモデルに不応期の情報を組み込む	7
4.3	出力が連続関数の場合	7
4.4	連続時間モデル	8
5	結合係数の更新式 —ヘップ則とデルタ則—	9
5.1	ヘップ則	9
5.2	デルタ則	10
6	階層型のネットワーク	10
6.1	階層型ネットワークの行列表現	11
6.2	パーセプトロン	13
6.2.1	パーセプトロンの学習	14
6.3	バックプロパゲーション (誤差逆伝播法)	18
6.3.1	XOR 問題、線形分離不可能な問題	18
6.3.2	誤差逆伝播法 (一般化デルタルール)	19
6.4	汎化能力と過学習	21
6.5	中間層ユニット数の決定問題	23
6.6	中間層のユニット数の増減法	23

6.7	リカレントネット — 文脈情報、時系列情報の取り込み — . . .	24
6.8	動径基底関数ネット RBF	27
7	相互結合型のネットワーク	27
7.1	同期更新と非同期更新	29
7.1.1	簡単な例	29
7.1.2	同期的更新	30
7.1.3	非同期更新	31
7.2	アソシアトロン	32
7.2.1	連想記憶	32
7.2.2	アソシアトロン	33
7.3	ホップフィールドモデル	33
7.3.1	エネルギー関数	34
7.3.2	連想記憶	35
8	自己組織化	36
8.1	リンスカーのシミュレーション	38
8.2	主成分分析	39
8.3	パーセプトロンモデル	39
8.4	コホネンの SOM	41
9	おわりに	42
9.1	本章で紹介した以外の話題	42
9.2	神経回路網モデルのすすめ	43

人間の感情と、他の生物のそれと、近代的な型の自動機械の反応との間に鋭い乗り越えられない区画線を引く心理学者は、私が私自身の主張に慎重でなければならないのと同様に、私の説を否定するのに慎重でなければならない

— N. Wiener, The Human Use of Human Beings(人間機械論, みすず書房, p.73) —

この章は 1997 年および 1999 年に千葉大学文学部で行なわれた認知情報科学特論「神経回路網モデル」の講義ノートを元としている。冒頭の引用はこの講義の初回に受講者¹に提示して、人間の行動と計算機シミュレーションとの関係を考えてもらうためのものである。まず最初に、神経回路網モデルの簡単な定義(1節)を与え、神経回路モデルのおおまかな分類(2節)を示している。次に、神経回路網モデル構築の準備として、生理学的な事実の概説(3節)と、モデル構築する際に必要な神経細胞の動作方程式の記述(4節)を試みてから、シナプス結合の変化を記述する古典的な学習則(5節)を紹介している。実際のモデルの説明としては、認識モデルとしての階層型のネットワーク(6節)、連想記憶や最適化問題に適用可能な相互結合型のネットワーク(7節)、および、パターン分類モデルあるいはトポグラフィックマップ形成モデルとしての自己組織化(8節)の3つの話題を解説した。紙面の都合上、この章に記したのは全講義内容のおよそ半分程である。ここに記した以外の話題については最後(9節)に代表的な文献を挙げておいた。

1 神経回路網モデルとは何か?

神経回路網モデルとは、脳の振る舞いを模倣するための表現のことである。必ずしもプログラムによって表現されている必要は無いが、数式を使って表現される場合が多い。ニューラルネットワークモデル(neural network)、PDP(parallel distributed processing)モデル、あるいはコネクショニストモデル(connectionist)と呼ばれることもある。論文などのタイトルに上記のいずれかの言葉を入れるとき、心理学者は PDP モデルやコネクショニストモデルを使う傾向がある。ここでは、神経回路網モデルを、生体の中枢系で行なわれている情報処理の機能、性能、および特性を記述するための抽象化された表現、と定義しておく。

¹受講対象者は全員、若干の数学的知識(初等の線形数学)とプログラミングの知識(C, Java, Perl 等)の科目を履修済であった。履修済といっても全員エキスパートであることを意味しない。履修者のなかには全くプログラミングが不得手な学生も含まれていた。

2 神経回路網モデルの分類

だいぶ以前になるが大手家電メーカー各社から一斉にニューロと名付けられた洗濯機やその他の家庭電化製品が発売されたことがある。ニューロ家電製品と我々が考えているニューラルネットワークとはどこが異なるのであろうか。結論を述べると原理的には同じものであるが、目標が異なるということになる。

大まかに分けるとニューラルネットワークモデルには以下のような分類が存在する。

2.1 学習方式による分類

学習方式による分類には次の2つがある。

1. 教師あり、パーセプトロンなど
2. 教師なし、自己組織化、特徴マップ

外部からネットワークに対して望ましい出力(教師信号)を与えて、ネットワークに同じ出力を返すように学習させることを教師あり学習という。一方、明示的な教師信号を用いない学習を教師なし学習と呼ぶ。この場合ネットワークは入力信号の統計的性質を学習することになる。これら2つの大分類にしたがって、6節と7節で階層型のネットワークを、8節で自己組織化の概要を紹介する。

2.2 結合方式による分類

神経素子間の結合方式による分類には次の2つがある。

1. 階層型
2. 相互結合型

与えられた入力信号が特定の方向にしか伝播しないような回路を階層型(layered)の回路という。一方、信号が回路内を循環したり逆方向に伝搬したりする方の回路を相互結合型という。順方向への信号の伝播をフィードフォワード、逆方向へのそれをフィードバック(または帰還)と呼んで区別するとがある。さらに、連想記憶と呼ばれる回路については、相互想起型、自己想起型の区別などがある。

上記の分類にしたがって6節(階層型ネットワーク)と7節(相互結合型ネットワーク)で概要を紹介する。

2.3 ANN と BNN

ここで ANN と BNN という言葉を紹介しておこう。ANN とは Artificial Neural Network のことで脳の情報処理様式にヒントをえた工学的な応用に主眼がおかれている。上述の洗濯機などは ANN の一つの方向である商業的な実現の一例である。一方 BNN は Biological Neural Network の略で、我々人間を含む生物が如何にして情報を処理しているのかを研究するモデルに対して付与される言葉である。

多くの心理学者は BNN に興味あるはずである。最近では生理学的、解剖学的知見に加えて、機能的脳画像研究と結びつけて研究が行われる場合も多い。とりわけ時間解像度、空間解像度とも精度が飛躍的に向上している機能的脳画像研究と結び付いた研究は、従来の心理学的研究手法を根本から変革する力すら持っているように思われる。

ここでは、まず BNN の背景となるニューロンの生物学的基礎についてあらかじめ概観し、さらに、そうしたニューロンを人工的に実現する方法や、基本的な学習則についても概観しておきたい。

3 ニューロンの生物学的基礎

3.1 神経細胞 (neuron)

脳は莫大な数 (10^{10} 個以上ともいわれる) の神経単位 (ニューロン neuron) から成り立っている。このニューロンが脳の情報処理における基本単位である。複数のニューロンが結合してニューラルネットワークが形成されている。

個々のニューロンは、単純な処理しか行なわないが、脳はこのニューロンが相互に結合された並列処理システムであると捕えることができる。

ニューロンは、

1. 細胞体
2. 樹状突起
3. 軸索

とよばれる部分からなる。樹状突起はアンテナ (入力)、軸索は送電線 (出力) と考えれば分かりやすい。

ニューロンの内部と外部とでは Na^+ , K^+ イオンなどの働きにより電位差がある。通常、内部電位は外部よりも低い。外部を 0 としたときの内部の電位を膜電位という。入力信号が無いときの膜電位を静止膜電位という。

情報は樹状突起から電気信号の形でニューロンに伝達され、すべての樹状突起からの電気信号が加え合わされる。樹状突起からやってくる外部電気信号の影響で膜電位が一定の値 (しきい値 約 $-55mV$) を越えると約 1 msec の

間膜電位が急激に高くなる。このことをニューロンが興奮した (あるいは発火した) という。ニューロンの興奮は、軸索をとおって別のニューロンに伝達される。一旦興奮したニューロンはしばらくは興奮することができない。これを不応期という。読者は、電位変化に現われる興奮 (パルス)、不応期などの概念を図 1 によって確認されたい。

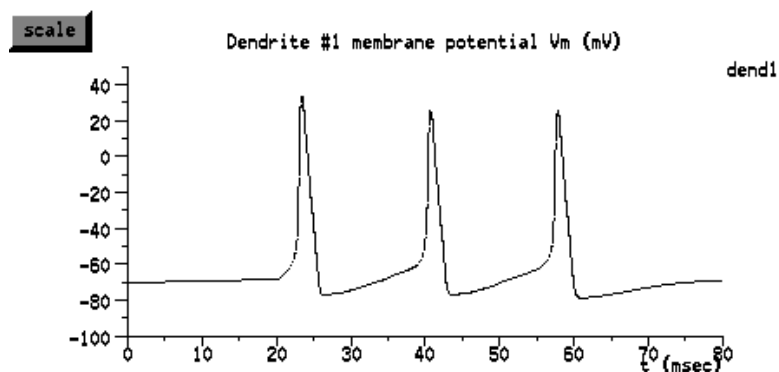


図 1: 膜電位の変化, GENESIS という有名なシミュレータの出力結果である。GENESIS 中の Neuron というスクリプトによって出力される結果を示した。ニューロンの動作はホジキン・ハックスレー方程式 (Hodgkin & Huxley(1952), しばしば H-H 方程式などと略記される) に従うものとしてシミュレーションが行われている。H-H 方程式は Na^+ , K^+ , Ca^+ などの濃度勾配に基づいた膜電位の変化を記述するコンダクタンスベースのニューロンの動作方程式の代表例である。README によると “Neuron” は複数の大学で教材として使われている。興味のある読者は <http://www.bbb.caltech.edu/GENESIS> を参照されたい

ニューロンの興奮 (1 msec だけなのでパルス pulse と呼ぶことがある) は軸索をとおって他のニューロンに伝達される。軸索を通る興奮の伝達速度は 100 m/s くらいである。たとえば、文字を見て音声を発声するまでの応答時間は、たかだか、1 秒程度で、ニューロンの応答時間を 10 数ミリ秒とすると多めに見積もっても 100 程度のニューロンしか通過していないことになる。このことは「100 step のプログラムの制約」と呼ばれる。

3.2 神経細胞の結合様式

ニューロンからニューロンへ情報が伝達される部分をシナプス (synapse) と呼ぶ。シナプスに興奮が到達するたびに送り手側 (シナプス前ニューロン) のニューロンからある種の化学物質が放出される。この化学物質は受け手側 (シナプス後ニューロン) の膜電位をわずかに変化させる。

化学物質の種類によって、膜電位を高めるように作用する場合（興奮性のシナプス結合）と逆に低めるように作用する場合（抑制性のシナプス結合）とがある。この他のシナプス結合としては、別の興奮性のシナプス結合の伝達効率を抑制するように働くシナプス結合（シナプス前抑制）が存在することが知られている。多くの研究者の努力にもかかわらず、今のところシナプス結合の種類はこの3種類しか発見されていない。我々は上記3種類のシナプスだけを考えればよさそうである。送り手のシナプスの興奮が興奮的に働くか抑制的に働くかは、送り手の側の細胞の種類によって異なることが知られている（Daleの法則）。

一つのニューロンには多いもので数万個のシナプス結合が存在する。多数の軸索にシナプス結合を通して興奮（あるいは抑制）が伝えられると細胞体を伝わる途中で重なり合う。すべての膜電位の変化の総和によってニューロンの膜電位の変化が決定される。すべてのシナプス結合の和のことを空間加算という。あるシナプスによって膜電位が変化し、その変化が減衰する前に次の興奮が伝達されれば、まだ残っている直前の電位変化に加え合わされて膜電位の変化が起きる。このことを時間加算という。

樹状突起を介したニューロン間の結合の強さは、しばしば変化することが知られている。これを学習という。

心理学との関連で言えば、

1. シナプス結合の強度変化 = 長期記憶
2. ニューロンの活動が保持されている状態 = 短期記憶

と考えてよいだろう。

4 単一ニューロンの動作方程式

本節では、前節で述べたニューロンの振舞いを数学的に記述することを考える。個々のモデルやネットワークの動作に入る前に、単一ニューロンの記述に注目するのは、原理的に本節のニューロンモデルを複数組み合わせればどんなモデルも記述可能であると考えられるからである。

本節では、ニューロンが振動数符号化法のみを利用している — すなわちニューラルネットワークにおけるすべての情報はニューロンの発火頻度によって伝達される — ことを仮定して話を進める²。このような記述の仕方は integration-and-fire モデルと呼ばれ、図1のホジキン・ハックスレー方程式に代表されるモデルをコンダクタンスモデルと呼んで区別することができる。H-H 方程式およびコンダクタンスモデルについては割愛した。

²たとえば アエベルス (Abeles, 1982) はニューロンがいろいろなシナプスから 同期的に到達するパルスを受けるとき、個々のパルスに対する反応の和より遥かに強い反応を示すことがある (synfire chains) ことを報告している。

一つのニューロンの振る舞いは、 n 個のニューロンから入力を受け取って出力を計算する多入力、1 出力の情報処理素子である (図 2)。 n 個の入力信号を x_1, x_2, \dots, x_n とし、 i 番目の軸索に信号が与えられたとき、この信号 1 単位によって変化する膜電位の量をシナプス荷重 (または結合荷重, 結合強度とも呼ばれる) といい w_i と表記する。抑制性のシナプス結合については $w_i < 0$, 興奮性の結合については $w_i > 0$ である。このニューロンのしきい値を h , 膜電位の変化を u , 出力信号を z とする。出力信号 z は

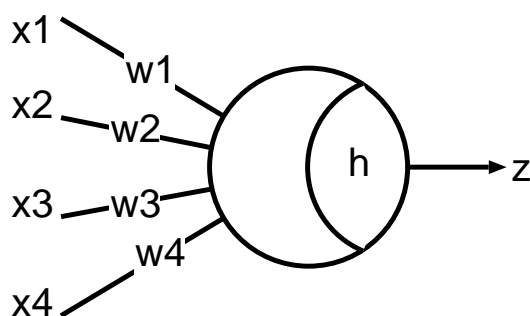


図 2: ニューロンの模式図

$$z = f(\mu) = f\left(\sum_{i=1}^n w_i x_i - h\right) \quad (1)$$

と表される。 $f(\mu)$ は出力関数であり、 $0 \leq f(\mu) \leq 1$ の連続量を許す場合や、 0 または 1 の値しかとらない場合などがある。連想記憶などを扱うときなどは $-1 \leq f(\mu) \leq 1$ とする場合もある。

4.1 マッカロック・ピッツの形式ニューロン

信号入力の荷重和

$$\mu = \sum_{i=1}^n w_i x_i \quad (2)$$

に対して、出力 z は u がしきい値 h を越えたときに 1 , そうでなければ 0 を出力するモデルのことをマッカロックとピッツ (McCulloch & Pitts, 1943) の形式ニューロン (formal neuron) と呼ぶ。マッカロック・ピッツの形式ニューロンは神経細胞の振る舞いを記述するもっとも古く (1943)、単純な神経細胞のモデルであるが、現在でも用いられることがある。

$$z = \begin{cases} 1, & u > 0 \text{ のとき} \\ 0, & \text{それ以外} \end{cases} \quad (3)$$

とすればマッカロック・ピッツのモデルは

$$z = 1 \left(\sum_{i=1}^n w_i x_i - h \right) \quad (4)$$

と表すことがある。(4) 式中の 1 は数字ではなく (3) 式で表される関数の意味である。このモデルは、単一ニューロンのモデルとしてではなく、ひとまとまりのニューロン群の動作を示すモデルとしても用いることがある。

4.2 マッカロック・ピッツのモデルに不応期の情報を組み込む

不応期が時間 r だけ存在するとすれば、

$$z = \begin{cases} 1, & u > 0 \text{ かつ過去 } r \text{ 時間内に出力が } 1 \text{ でなかったとき} \\ 0, & \text{それ以外} \end{cases} \quad (5)$$

とすれば不応期を表現することができる。

さらに、ニューロンが興奮すると時間 s だけしきい値が b_s だけ上がって興奮しにくくなると仮定すると

$$z(t) = 1 \left[\sum w_i x_i(t) - h - \sum_{s=1}^T b_s(t-s) \right] \quad (6)$$

と書くことができる。ここで $x_i(t)$ は時間 t での入力、 $z(t)$ は時間 t での出力である。(4) 式と比べて追加された項 $\sum_{s=1}^T b_s(t-s)$ が不応期を表現しており、この値が大きいと [] 内が負になるので発火できなくなる。

時系列データのように、過去に起こった出来事が現在の活動に影響する及ぼすモデルでは、過去の出来事を現在時刻との差 $(t-s)$ の関数と考えて、全時間について足し合わせることが多い。このことを「過去の履歴」と表現することもある。連続時間を仮定するモデルでは時間についての総和 \sum 部分が積分 \int に置き換わる。

より一般的には相互結合回路 (後述する 6.7 節、単純回帰型ネットワークなど) においては、長期的なニューロンの活動を表現する必要があり、ニューロンの活動履歴を表現するには不応期の説明で用いた式 (の変形) が必要になってくる。単純回帰型ネットワークが、なぜ時系列データを学習したり、言語知識を獲得できるのかは、現在のニューロンの状態が過去の活動履歴を保持する能力があるか否かに依存している。

4.3 出力が連続関数の場合

時間 t における入力信号は $x_i(t)$ は i 番目のシナプスの興奮伝達の時間 t 付近での平均ととらえることができる。すると、最高頻度の出力を 1, 最低

(興奮無し) を 0 と規格化できると考えて $0 \leq f(\mu) \leq 1$ とする。入出力関係は f を用いて

$$z = f(\mu) = f\left(\sum w_i x_i(t) - h\right) \quad (7)$$

のように表現される。このモデルは、ニューロン集団の平均活動率ととらえることもできる。

良く用いられる出力関数 f の形としては、 $\mu = \sum w_i x_i - h$ として、

$$f(\mu) = \frac{1}{1 + e^{-\mu}}, \quad (8)$$

や

$$f(\mu) = \frac{1}{2} \tanh \mu - 1, \quad (9)$$

などが使われることが多い。どちらを用いるかはほとんど使う人の趣味の問題ともいえる。どちらの関数も微分が極端に簡単になるという理由が大きい。後述するように出力 z が結合荷重や入力信号の微小変化に対してどのように変化するかを調べる必要があるからである。(8) 式および (9) 式は、入力信号の重みつき荷重和 μ としてニューロンの活動が定まることを示している。後述するバックプロパゲーション則 (6.3.2 節) で必要となるので、 μ の微小な変化がニューロンの活動どのような影響を与えるか調べるために (8) および (9) を μ で微分することを考える。

$$f(x) = \frac{1}{1 + e^{-x}}, \quad (10)$$

を x について微分すると

$$\frac{df}{dx} = f(x)(1 - f(x)) \quad (11)$$

$$f(x) = \tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (12)$$

を x について微分すると

$$\frac{df}{dx} = 1 - \tanh^2 x = 1 - \{f(x)\}^2 \quad (13)$$

\tanh は双曲線関数である。

4.4 連続時間モデル

時間 t を連続だとみなし、 $\mu(t)$ を静止膜電位 のとき 0 をとるものとする。電位 $\mu(t)$ は時定数 τ で減衰する項と外部入力に応じて増減する項との和で表せるとするモデルは、以下のような微分方程式になる。

$$\tau \frac{d\mu(t)}{dt} = -\mu(t) + \sum_{i=1}^n w_i x_i - h \quad (14)$$

x_i は一定 (外部入力は一一定) で $\mu(0) = 0$ の初期条件のもとで解くと、

$$\mu(t) = -\mu_0 (1 - e^{-t/\tau}) \quad (15)$$

となる。ただし $\mu_0 = \sum w_i x_i - h$ 。

5 結合係数の更新式 —ヘップ則とデルタ則—

神経回路網モデルでは、シナプスの結合荷重を変更することを神経回路網の学習と呼ぶことが多い。学習時における結合荷重の変化を記述したルールを学習則 (learning rule) と呼ぶ。ここでは 2 つの神経回路網の学習規則を紹介する。これら 2 つ学習規則はいずれも歴史的な価値があり、かつ、現在でもその応用が研究されている基本的な学習則である。

5.1 ヘップ則

「同時に発火したニューロン間のシナプス結合は強められる」ことを主張しているのがヘップ則 (Hebbian rule) である (Hebb, 1949)。ヘップの学習則と表記されることも多い。ヘップ則は以下のように定式化できる。ニューロンの発火状態を 1、休止状態を 0 と表現することにし、ニューロン y_i からニューロン x_j へのシナプス結合強度を w_{ij} とする。このときヘップの学習則は、シナプス結合強度の変化 Δw_{ij} として表現され、

$$\Delta w_{ij} = \lambda x_j y_i, \quad (16)$$

と書くことができる。ここで $\lambda (\geq 0)$ は学習定数という。 x_j と y_i は 1 と 0 の 2 とおりの状態にしかならないため、可能な組み合わせは 4 通りである。このうち $\Delta w_{ij} \neq 0$ となる組み合わせは、 $(x_j = 1)$ かつ $(y_i = 1)$ の場合だけ

		x_j	
		1	0
y_i	1	λ	0
	0	0	0

Table 1: ヘップの学習則

である。

y が 1 または 0 の値しか取らないことに注意すれば、(16) は

$$\Delta \mathbf{w} = \begin{cases} \lambda \mathbf{x} & y \text{ が発火したとき,} \\ 0 & \text{それ以外するとき,} \end{cases} \quad (17)$$

と書くことができる。すわなち結合荷重係数は、入力ニューロンと出力ニューロンの同時発火頻度に比例して、入力ニューロン x の λ 倍だけ増大することを意味する。

ヘップ則の変形はさまざまに考えられていて、例えば同時に発火しなかった時にシナプス結合を弱めるアンチヘップ則、減衰項を加える方法、入力と出力の差 (の 2 乗) を利用する方法、などがある。ヘップ則の変形 (の一部) は、8 節でも簡単に触れることにする。

5.2 デルタ則

デルタ則はデルタルール、LMS 則、あるいは Widrow-Hoff の学習則などと呼ばれることもある。

デルタ則の説明のために、 n 個の入力層素子と 1 個の出力素子 y からなる単純な 2 層のネットワークを考えよう。出力素子 y の活動は、入力層からの信号 $x_i (i = 1 \dots n)$ の重みつき荷重和 $\sum w_i x_i$ で定まるとする。望ましい出力 (教師信号) を y^* とすれば、デルタ則は、望ましい出力 (教師信号) y^* と実際の出力 y の差 (デルタ) に入力信号 x を掛けた形

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \eta \delta_t \mathbf{x}_t = \mathbf{w}_t + \eta (y_t^* - y_t) \mathbf{x}_t, \quad (18)$$

で表現される。ここで η は学習係数と呼ばれる定数である。

デルタ則とは δ^2 を最小にする規準を導入した学習則である。すなわち、入力信号と望ましい出力とが与えられたとき

$$f(\mathbf{w}) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^{\infty} \delta^2 = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^{\infty} (y_i^* - y_i)^2 \quad (19)$$

という関数の極限を考え、この関数 $f(\mathbf{w})$ を最小化することを考えるのである。 $f(\mathbf{w})$ は結合係数 w で決まるので、 w を微小に変化させたとき (微分係数) の $f(\mathbf{w})$ 変化量の逆方向 (f を w で微分しマイナスをかける) に逐次 w を変化させていくことで極小解に達する (図 3) ことが予想できる。これは、最急降下法と呼ばれる最適化問題の解法の一つである。

デルタ則の欠点は 2 層の結合係数しか扱えないことである。デルタ則を多層の回路での学習に適用できるようにしたものが 6.3 節で紹介する一般化デルタルール、あるいは、バックプロパゲーション法とよばれる学習則である。

6 階層型のネットワーク

パーセプトロンに代表されるフィードフォワード型の結合を持つ階層型のネットワークは、パターン認識 (pattern recognition)、情報圧縮 (data compression)、運動制御 (motion control)、雑音除去 (noise reduction)、および

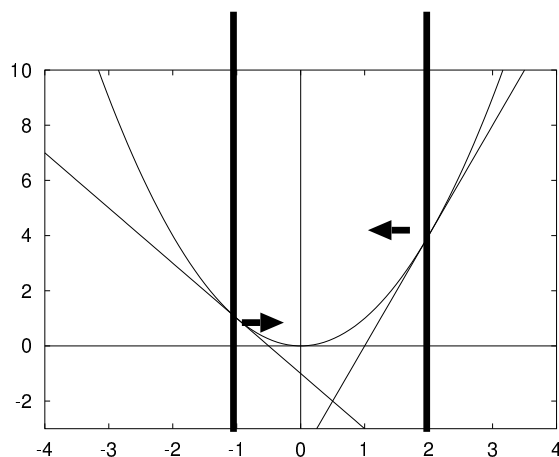


図 3: 最急降下法 (例えば伊理 (1981)) は接線の傾きと反対方向に向かって進めば極小点に達するというアイデアに基づいている。図にあるとおり接線の傾きが正であれば負の方向に、反対に接線の傾きが負であれば正の方向に少しだけ進めば極小点に近付くことができる。この操作を繰り返せばやがて極小点に達することができる。移動量は学習係数に比例する。学習係数が小さいと極小点に達するまでに時間がかかるが、学習係数が大きいと極小点を飛び越してしまい収束しない。

時系列予測 (time series prediction) などへの理論的、または応用的研究が試みられている。ここでは、パーセプトロンの学習について、幾何学的表現を用いてやや詳しく解説し、線形分離可能な問題について紹介した。続いてバックプロパゲーション法を導入し、階層型のネットワークについての幾つかの話題を解説している。

6.1 階層型ネットワークの行列表現

以降では表記を簡単にするために線形数学の表記、すなわちベクトルと行列による表記方法を導入する。 n 個の入力信号の組 (x_1, x_2, \dots, x_n) をまとめて x のようにボールド体で表す。本章では一貫してベクトル表記には小文字のボールド体を、行列には大文字のボールド体を用いることにする。例えば n 個の入力信号の組 $(x_1, x_2, \dots, x_n) = x$ に対して、同数の結合荷重 $(w_1, w_2, \dots, w_n) = w$ が存在するので、加算記号 \sum を使って表現していた任意のニューロンへの全入力 $\mu = \sum w_i x_i$ はベクトルの内積を用いて $\mu = (w \cdot x)$ と表現される。なお、横一行のベクトルを行ベクトル、縦一列のベクトルを列ベクトルと呼ぶことがある。本章では行ベクトルと混乱しないように、必要に応じて列ベクトルを表現する際には $(x_1, x_2, \dots, x_n)^T = x$ とベクトル

の肩に Γ を使って表現することもある。

図 4 のような単純な 2 層の回路を例に説明する。図 4 には、3 つの入力素

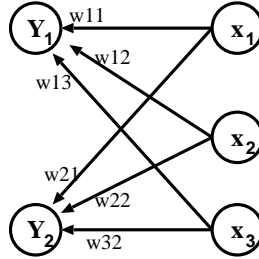


図 4: ネットワークの行列表現

子³と 2 つの出力素子の活性値 (ニューロンの膜電位に相当する) x_1, x_2, x_3 と y_1, y_2 、および入力素子と出力素子の結合強度を表す $w_{11}, w_{12}, \dots, w_{32}$ が示されている。これらの記号をベクトル x, y と行列 w を使って表すと $y = Wx$ となる。図 4 の場合、ベクトルと行列の各要素を書き下せば、

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (20)$$

のようになる。

行列の積は、左側の行列の i 行目の各要素と右側の行列 (ベクトルは 1 列の行列でもある) の i 列目の各要素とを掛け合わせて合計することなので、以下のような、加算記号を用いた表記と同じである。

$$\begin{aligned} y_1 &= w_{11}x_1 + w_{12}x_2 + w_{13}x_3 = \sum_i w_{1i}x_i \\ y_2 &= w_{21}x_1 + w_{22}x_2 + w_{23}x_3 = \sum_i w_{2i}x_i \end{aligned} \quad (21)$$

これを、 m 個の入力ユニットと n 個の出力ユニットの場合に一般化すれば、

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} \quad (22)$$

$$\mathbf{y} = \mathbf{W}\mathbf{x} \quad (23)$$

と表現できる。しきい値の扱いについては、常に 1 を出力する仮想的なニューロン $x_0 = 1$ を考えて W に組み込むことも可能である。

実際の出力は y の各要素に対して

$$f(y) = \frac{1}{1 + e^{-y}} \quad (24)$$

³ユニットと呼ばれることもある

のような非線型変換を施すことがある。

階層型のネットワークにとっては、(24) の非線型変換が本質的な役割を果たす。なぜならば、こうした非線型変換がなされない場合には、ネットワークの構造が何層になったとしても、この単純なシナプス結合強度を表す行列を W_i ($i = 1, \dots, p$) としたとき、 $W = \prod_{i=1}^p W_i$ と置くことによって本質的には 1 層のネットワークと等価になるからである。

$$\mathbf{y} = W_p W_{p-1} \cdots W_1 \mathbf{y} = \left(\prod_{i=1}^p W_i \right) \mathbf{y} \quad (25)$$

6.2 パーセプトロン

パーセプトロンとはローゼンブラット (Rosenblatt, 1958) によって提案された図 5 のような 3 層の階層型ネットワークモデルである。パーセプトロンはマッカロック・ピッツの形式ニューロンを用いて学習則にヘップ則 (5) 式を使ったモデルで、単純な認識能力を獲得することができる。最下層の入力

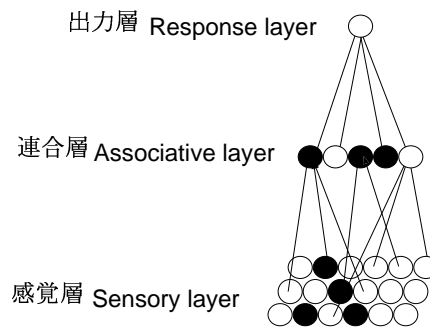


図 5: パーセプトロンの模式図

層は外界からの刺激入力表現している。中間層では入力情報の変換が行われ、最上位層である出力層で認識に到る。

簡単のため、出力ユニットが 1 つしかない場合を考えれば、パーセプトロンは入力層に現われるパターンを 1 と 0 とに 2 分する機械であるということができる。出力層の i 番目のユニットへの入力 (膜電位の変化) u_i は

$$u_i = \sum_j w_{ij} x_j - \theta_i = \mathbf{w}_i \cdot \mathbf{x}_i - h_i, \quad (26)$$

と表現される。ここで中間層の j 番目のユニットの出力 y_j とこのユニットとの結合係数を w_{ij} 、しきい値を h_i とした。このユニットの出力 y_i (活動電

位、スパイク)は、

$$y_i = [u_i] \quad \begin{cases} 1 & \text{if } u_i \geq 0, \\ 0 & \text{otherwise} \end{cases} \quad (27)$$

と表される。すなわち、活性値 u_i は、式 (27) のような関数によって、1 または 0 の出力に変換されることになる。活性値が 0 より大きければ 1 を出力し、それ以外の場合は 0 となる。

6.2.1 パーセプトロンの学習

パーセプトロンの学習は連合層から反応層への結合強度の変化として表現される。最初の入力がなされる感覚層から連合層への結合強度は一定で変化しない。すなわちパーセプトロンは 3 層のネットワークではあるが、学習を考える際には、入力層と出力層の間だけ考えれば良い。

パーセプトロンが特定のパターンに対してだけ出力するようになるためには、学習によってネットワーク内部の結合強度を変化させる必要がある。その際、パーセプトロンには種々の入力パターンが与えられ、それぞれが検出すべきパターンであるか否かが「教師信号」として与えられる。具体的には、入力信号を 2 分する問題を学習する場合、検出すべきパターンの時に教師信号として 1 が、それ以外のパターンの場合には、教師信号として 0 が与えられる。ここでいう教師信号とは、のデルタ則の説明での望ましい出力に相当する。

パーセプトロンの学習は中間層から出力層への結合荷重の変化として表現される。中間層から入力層への結合荷重については考慮されないことに注意が必要である。パターン c に対する教師信号 (望ましい出力) を t_c と書くことにするとパーセプトロンの学習、すなわち結合強度の更新式は

$$\Delta w_{ji} = \eta \delta x_{c,j} = \eta (t_{c,j} - y_{c,j}) x_{c,j}. \quad (28)$$

と表すことができる。 η は学習係数と呼ばれる定数である。パーセプトロンの出力と教師信号と差分 $\delta = t_{c,j} - y_{c,j}$ のことを誤差信号と呼ぶ。上式は w_{ji} の更新式として

$$w_{ji}(n+1) = w_{ji}(n) + \eta \Delta w_{ji}(n) = w_{ji}(n) + \eta (t_{c,j} - y_{c,j}) x_{c,j}. \quad (29)$$

のような漸化式として表現されることがある。ベクトル表現すれば

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta (\mathbf{t}_c - \mathbf{y}_c) \mathbf{x}_c. \quad (30)$$

ここで、 $\mathbf{w}(n)$ は n 回目の学習終了時点での結合強度を入力層のユニットだけ並べて (w_1, w_2, \dots, w_N) としたものである。 \mathbf{t}_c はパターン c に対応する教師信号を表し、 \mathbf{y}_c はパターン c に対するパーセプトロンの出力を表す。 \mathbf{x}_c は入力信号である。

入力ユニット数 2、出力ユニット数 1 の 2 層で構成される単純なネットワークを考えれば、このネットワークへの入力全体は 2 次元平面で表現できる。この平面のことを入力空間と呼ぶことがある。この場合は入力層のユニッ

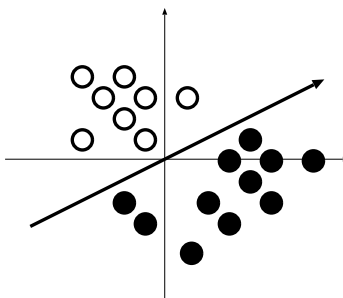


図 6: パーセプトロンの入力の幾何学的表現と判別直線

ト数が 2 だったから平面であった。パーセプトロンにおける学習とはこの空間を 2 分割するような領域に分割する判別直線を見つけることであるとい

ことができる。
いま、しきい値が 0、すなわち入力データを 2 群に分ける直線が原点を通る場合を考えることにする。すなわち、図 6 で判別直線より上の白丸には 1 を、判別直線より下の領域にある黒丸には 0 を出力するような学習を考える。 $w_1x_1 + w_2x_2 > 0$ であれば 1 を出力し $w_1x_1 + w_2x_2 \leq 0$ ならば 0 を出力するとは、ベクトル (x_1, x_2) とベクトル (w_1, w_2) との内積の正負の判断をしているのと同義である。なぜならベクトルの内積とは 2 通りの表現があっ

$$\mathbf{w} \cdot \mathbf{x} = w_1x_1 + w_2x_2 = |\mathbf{w}| |\mathbf{x}| \cos \theta \quad (31)$$

だからである。上式の最右辺の正負は 2 つのベクトルのなす角 $\cos \theta$ のみによって決まる。すなわち 図 7 のように斜線をつけた領域にある全てのベク

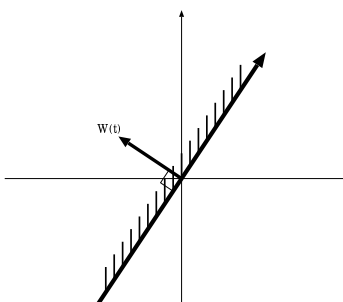


図 7: 2 群を判別するとき斜線の領域で正になるとは、法線ベクトルとの内積が正であることを意味する

トルは、2 群を判別する直線の法線ベクトル (2 つの結合強度を要素とするベクトル) との内積が正である。入力信号と結合強度ベクトルのなす角 θ が $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$ の範囲のとき $\cos \theta > 0$ となり、そうでなければ負になるからである。

このパーセプトロンに学習すべきデータが一つ入って来たと仮定しよう。図 8 は 1 と出力すべき ($y = 1$ すなわち法線ベクトルとの内積が正であるべ

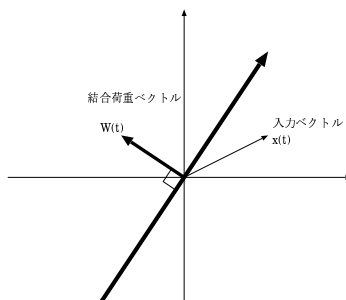


図 8: 学習すべきデータ

き) データを、誤って 0 と出力してしまったという事態である。このとき、式 (30) 内の $(t - y)$ は $1 - 0 = 1$ になるので結合荷重の更新式、すなわち法線ベクトルの更新式は

$$\mathbf{w}(n) + \eta(t_c - y_c) \mathbf{x}_c = \mathbf{w}(n) + \eta \mathbf{x}_c. \quad (32)$$

となってベクトルの足し算になる (図 9)。これによって判別直線が回転し (図

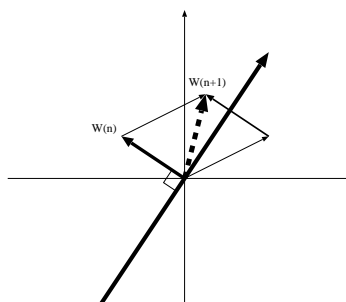


図 9: パーセプトロンの学習則による結合荷重ベクトル w の更新

10)、今度は法線ベクトル w とデータ x とのなす角が 90 度以内になる。0 と出力すべきデータを、誤って 1 と出力してしまった場合はベクトルの引き算になる。

最後に判別直線が原点を通らない場合、すなわち閾値が 0 ではない場合を見ておこう (図 11)。この場合は、原点を通る直線を並行移動したことになる

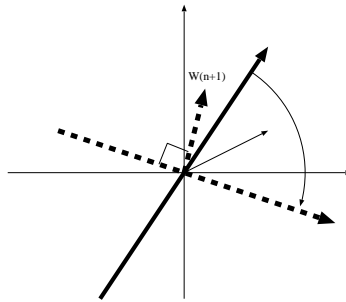


図 10: 判別直線の回転

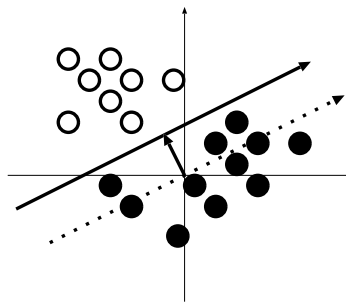


図 11: しきい値の意味

ので、原点を通る判別直線では 1 と答えるべき領域 (図 11 中の白マル) に入ってしまった点を底アゲして 0 と答えるべき領域 (図 11 中の黒マル) の領域にするために行われると考えることができる。

パーセプトロンによるパターン分類とは、出力細胞が 1 の場合、入力パターン空間 (n 個の入力があれば n 次元空間) を $wy = \theta$ を満たす境界 ($n-1$ 次の超平面) によって 2 つの部分空間に分割することであると言える。逆に言えば、入力パターン空間が一本の判別直線によって分割できないようなパターンをパーセプトロンは学習することができない。このことを線形分離可能性 (linear separability) という。

以上をまとめると、出力層と入力層との間の結合強度を入力データ空間におけるベクトルと考えれば、パーセプトロンの出力は結合強度ベクトルと入力データとの内積が 0 より大きければ 1 を、小さければ 0 を出力する機械であるみなすことができる。また、パーセプトロンの学習は、内積の大きさによって結合強度ベクトルを回転させることだと言うことができる。このようにして訓練されたパーセプトロンでは、入力データ空間上で線形分離可能な問題ならば、必ず学習が可能であることに注意されたい。逆に言えば、線形分離不可能な問題でも、適切に次元を設定することで線形分離可能な空間に写像するような中間層表現を得ることができればパーセプトロンの持つ限界を越

えることができるというアイデアに結び付く。

ミンスキーとパパート (Minsky & Papert, 1988) はパーセプトロンのベクトル表示について悲観的な考え方を持っているようであるが、ここでは理解のしやすさを優先してベクトル表示による説明を用いた。

パーセプトロンのような階層型のネットワークは、中間層の素子を基底関数とみなすことで、関数近似法の一つと考えることができる。このことから、他の数学的な関数近似手法との関連⁴が指摘できる。一般にこれらの数学的手法は入力変数の次元数 n が増えれば、一定の近似誤差内で関数を近似するためには、必要なパラメータ数が n 乗のオーダーで増加することが知られており、このことは次元の呪い (curse of dimensionality) と呼ばれている。多層パーセプトロンによる関数近似では、この次元の呪いを避けることができることが指摘されている。

6.3 バックプロパゲーション (誤差逆伝播法)

6.3.1 XOR 問題、線形分離不可能な問題

パーセプトロンでは絶対に解けない問題に排他的論理和 (XOR) 問題がある。排他的論理和とは、2 つの入力のうちいずれか一方のみが 1 のとき、1 を出力する問題である。図 12 左を見るとわかるとおり、XOR 問題は一本の判別直線で白マルと黒マルを分離できない、すなわち線形分離不可能な問題である。

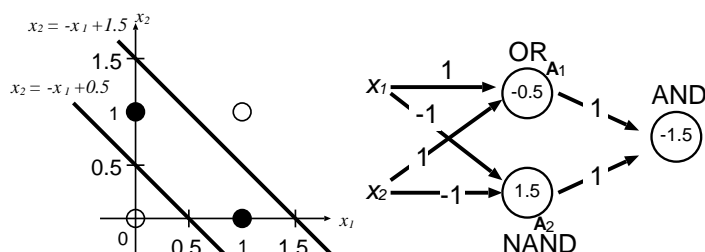


図 12: XOR 問題の幾何学的表現と XOR 問題を解くためのネットワーク

図 12 右は図 12 左の幾何学表現を対応するネットワークとして表現したものである。一番左が入力層、一番右が出力層、中間の 2 つが中間層である。ユニットの中に書かれた数値は各ユニットのしきい値を示している。中間層の 2 つのユニットのうち上の OR と書かれたユニットは、 $x_1 + x_2 - 0.5 > 0$ のとき発火する。この式を書き換えると、 $x_2 > -x_1 + 0.5$ となるので図 12 左の下斜線より上の領域に対応する。一方、中間層の下 NAND(not and)

⁴たとえばテーラー展開、フーリエ級数展開、スプライン関数など。

と書かれたユニットは、 $-x_1 - x_2 + 1.5 > 0$ のとき発火するから、移項して $x_2 < -x_1 - 1.5$ とすれば、図 12 左の上斜線より下の領域に対応していることが分かる。さらに、AND と書かれた出力ユニットは、2つの中間層ユニットからの情報の論理積 (AND) を計算するユニットになっている。そこで、2つの中間層ユニットの両方が発火する場合だけ、出力ユニットも発火し、1 を出力する。これは、図 12 左では、「下の斜線より上の領域」でかつ「上の斜線より下の領域」に対応する。すなわち、図中の黒丸の領域だけが分離されることになる。このような2本の直線は図中にいくらかでも引けることから、XOR 問題の解も無数に存在することが分かる。

入力層		中間層		出力
x_1	x_2	a_1	a_2	r
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

Table 2: 図 12 に対応する XOR 問題の真偽表

図 12 左にあるとおり、中間層の素子 1 個は 1 つの線形判別関数に相当すると考えてよい。中間層から出力層への結合では各出力の論理積 AND を計算していることに相当する。 n 個の中間層を用意すれば原理的には $\frac{n^2+n+2}{2}$ 個のカテゴリ分類が可能である。パーセプトロンが XOR 問題を解くことができない理由は入力層から中間層にいたる結合強度を変更する手段がないことである。

6.3.2 誤差逆伝播法 (一般化デルタルール)

XOR 問題でも見たように、パーセプトロンの問題点は学習が出力層と中間層の間だけで行われ、下の層に伝播しないことである。この点を改良したのがバックプロパゲーション (一般化デルタルール) と呼ばれる学習則である。

m 層のネットワークを考え、 k 層の i 番目のユニットへの総入力を x_i^k 、このユニットの出力を y_i^k 、 $k-1$ 層の i 番目のユニットから k 層の j 番目のユニットへの結合係数を $w_{ij}^{k-1,k}$ と表記する。各ユニットの出力は

$$y_i^k = f(x_i^k) = \frac{1}{1 + e^{-x_i^k}} \quad (33)$$

$$x_j^k = \sum_i w_{ij}^{k-1,k} y_i^{k-1}, \quad (34)$$

で定義されているものとする。あるデータ x と教師信号 t が与えられたとき

教師信号と出力との 2 乗誤差を

$$E = \frac{1}{2} \sum_j (y_j^m - t_j)^2 = \frac{1}{2} \sum_j \delta^2 \quad (35)$$

と表記する。総和記号 \sum の前の $1/2$ は微分したときに式を簡単にする程度の意味しかないので本質的ではない。この誤差関数 E は、教師信号と出力との差の 2 乗に比例して大きくなる。そこで、 E が減少する方向に w の値を逐次更新することがバックプロパゲーション法の基本的な発想である。(35) 式の誤差 E は各 y_j^m の 2 次関数とみなすことができるので $E \geq 0$ であり、 $E = 0$ となるのは、すべての y_j^m に対して $y_j^m - t_j = 0$ のとき、すなわち完全に学習が成立したときだけである。

$$\Delta w_{ij}^{k-1,k} = -\epsilon \frac{\partial E}{\partial w_{ij}^{k-1,k}} \quad (36)$$

(36) 式で E は y_j^m の関数であるが、さらに y_j^m は x_j^m の関数であり、さらにさらに x_j^m は $w_{ij}^{m-1,m}$ の関数であるから合成関数の微分公式により

$$\Delta w_{ij}^{m-1,m} = -\epsilon \frac{\partial E}{\partial w_{ij}^{m-1,m}} \quad (37)$$

$$= -\epsilon \frac{\partial E}{\partial y_j^m} \frac{\partial y_j^m}{\partial x_j^m} \frac{\partial x_j^m}{\partial w_{ij}^{m-1,m}} \quad (38)$$

$$= -\epsilon (y_j^m - t_j) y_j^m (1 - y_j^m) y_i^{m-1} \quad (39)$$

$$= -\epsilon \delta_j^m y_i^{m-1}. \quad (40)$$

となる (ただし $\delta_j^m = (y_j^m - t_j) y_j^m (1 - y_j^m)$)。もし仮に (33) 式で与えられている出力関数が線形関数 $y(x) = x$ であれば、(40) 式は $\Delta w_{ij} = (t_j - y_j) y_i^{m-1}$ となってパーセプトロンの学習式と一致する。

次に中間層以下第 n 層 ($n \neq m$) のユニット y_j^n の結合強度の更新には、

$$\frac{\partial E}{\partial y_j^n} = \sum_k \frac{\partial E}{\partial y_k^{n+1}} \frac{\partial y_k^{n+1}}{\partial x_k^{n+1}} \frac{\partial x_k^{n+1}}{\partial y_j^n} \quad (41)$$

$$= \sum_k \frac{\partial E}{\partial y_k^{n+1}} \frac{\partial y_k^{n+1}}{\partial x_k^{n+1}} \frac{\partial}{\partial y_j^n} \sum_i w_{ik}^{n,n+1} y_i^n \quad (42)$$

$$= \sum_k \delta_k^{n+1} w_{kj}^{n,n+1}, \quad (43)$$

を誤差信号 δ として再帰的に計算する。以上をまとめると、結合強度の修正量 $w_{ij}^{k-1,k}$ は

$$\Delta w_{ij}^{k-1,k} = -\epsilon \delta_j^k y_i^{k-1}. \quad (44)$$

$$\text{where } \delta_j^k = \begin{cases} (y_j^k - t_j) y_j^k (1 - y_j^k), & k = m \text{ 出力層のとき} \\ \left(\sum_l \delta_l^{k+1} w_{jl}^{k,k+1} \right) y_j^k (1 - y_j^k), & \text{otherwise} \end{cases} \quad (45)$$

となる。式 (45) を見ると誤差の計算がデータ処理とちょうど逆の流れで入力層まで伝播するようになっている。これが誤差逆伝播法と呼ばれる所以である。

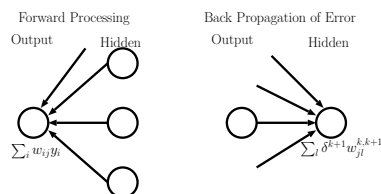


図 13: 誤差逆伝播の模式図

6.4 汎化能力と過学習

学習すべきデータ集合の中からいくつかのサンプルを選び訓練課題セットとして回路網に学習させる。このとき、学習させた学習課題セット以外のデータをテスト課題として選び、このテスト課題の成績を調べることで学習したルールの一般化能力を測定することができる。

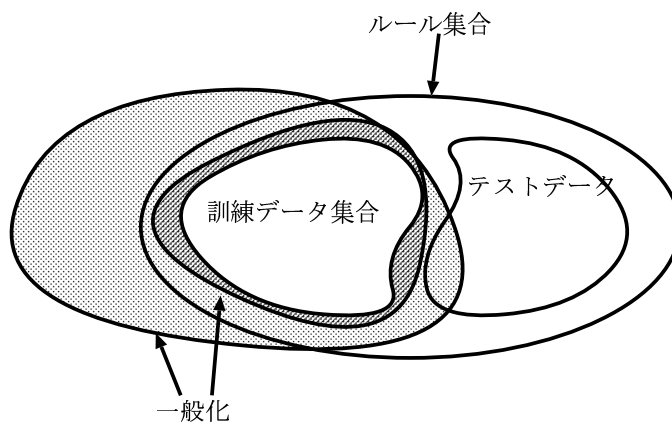


図 14: 汎化能力と過学習

学習した内容がルールの適用範囲と完全に重なることが理想であるが、図 14 で示したように一般化にはさまざまな可能性が考えられる。例えば入出力とも 0,1 のデータで、入力層が N 個、出力層が 1 個である場合を考えよう。入力パターンの総数は N 個の入力層の可能な組み合わせ、すなわち 2^N 個存在する。これらの入力集合を 0 か 1 かに分類する課題では、全部で 2^{2^N} とおりの分類が可能である。この中から M 個の入力を選んで訓練した場合に

は、残りの分類パターンはすべて一般化になるので $2^{2^N} - M$ とおりの一般化が可能になる。このことは入力層の素子数 N が大きくなると、実質的に無限大の一般化が考えられることを意味する。

一方、訓練課題セットでは正解を得ることができるが、テスト課題では正解できないことがある。学習が進行しすぎるとしばしば観察される現象で、過剰な学習がなされたことを意味する(過学習 overfitting)。図 14 は、誤った方向に一般化がなされた場合と、過学習によって訓練課題にだけ正解するようになった場合とを表したものである。

図 15 では正弦曲線 $y = 0.5 + 0.4 \sin(2\pi x)$ を 3 層のネットワークに学習させた例が示されている。この例では 0 から 1 までの 0.05 刻みの各 x 座標を入力信号として与え、対応する y の値を学習させた。実際の教師信号に若干のノイズを加えてある。一般に教師信号に少量のノイズを加えたデータを学習させることで、ネットワーク一般化能力が向上するとされている。ただし、データセット数に対して中間層の数が多いときに繰り返し学習を進行させると過学習が生じる。図 15 中の 2 本の点線のうち、ほぼノイズを付加し

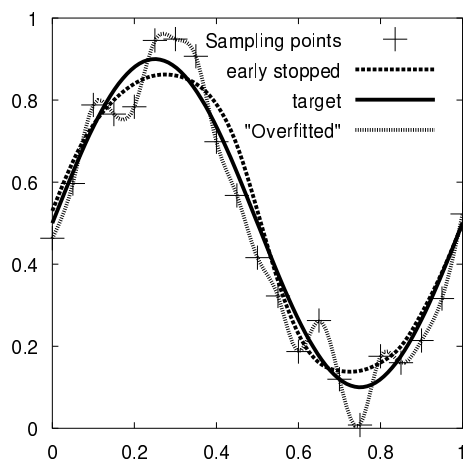


図 15: 正弦曲線を 3 層バックプロパゲーション法によって学習させた例。実線が $0.5 + 0.4 \sin(2\pi x)$ の曲線であり、プラス (+) の記号で実際に用いられた教師信号 (ノイズ付加) が示されている。2 本の点線によって、中間層を少なくして学習終了基準を甘く設定した結果と、中間層の数を増やして意図的に過学習を起こさせた結果とを示した。

た教師信号完全に学習している点線では、過学習によって真の曲線を学習するのではなく真の関数とノイズとの合成積 convolution を学習している。他方の点線では、ほぼ望み通りの結果を得ているが⁵、最大値 0.9 付近、最小値

⁵実際に用いた中間のユニット数は 6

0.1 付近での真の関数とのずれがやや大きくなっていることが読み取れる。

6.5 中間層ユニット数の決定問題

入力情報が N ビットの情報を持っているとき、全入力情報を損失無く表現するためには 2^N 個の中間層ユニットを用意すれば十分であることはすぐに分かる。ところが、これは中間層のユニット数決定のための必要条件ではない。最適な中間層のユニット数を決定するためには、6.4 節の一般化の問題を踏まえて議論する必要がある (甘利、村田、Müller(1997), Elman ら (1996))。

村田ら (1994) は、神経回路網を確率機械と捉えて、情報量規準を用いて中間層を定める手法を提案した。彼らは赤池の情報量規準 AIC (Akaike's Information Criterion, 坂本、石黒 & 北川 (1983)) を拡張した NIC (Network Information Criterion) を提案している。データが与えられたときのモデルの対数尤度に自由パラメータ数の 2 倍を加えたものが赤池の情報量規準 AIC と呼ばれている。(データの当てはまりを表す量である) 対数尤度が同じならば自由パラメータ数の少ないモデルを選択すべきであることを AIC は主張している。複数のモデル間で AIC を計算、比較して最適モデルを選択しようとするのが AIC の基本となるアイデアである。村田らの提案した NIC は、中間層のユニットをパラメータと考え、訓練データ上で、任意の入力信号に対するモデルからの出力と教師信号 (正解) との「ずれ」に、パラメータ数を加えたものとして定義されている。もし、モデルが真の入出力関係を実現可能であり、かつ、上記の「ずれ」が対数のマイナスで定義されているならば、NIC と AIC とは係数を除いて一致する。

6.6 中間層のユニット数の増減法

中間層のユニット数をネットワークに学習させるという手法も開発されている。最初は十分な数のユニットを用意して、全結合させ、学習中に不要な結合を刈り込む手法を枝刈り法 (pruning あるいは weight elimination) という。結合強度 w_{ij} が徐々に 0 になるような傾向を持たせて、学習によって結合強度が補強されない限り不要な結合が除去されるように方法を総称している。具体的には w_{ij} の更新後に

$$w_{ij}^{\text{new}} = (1 - \epsilon)w_{ij}^{\text{old}} \quad (46)$$

このことはバックプロパゲーションで使われる誤差の 2 乗和 E_0 に w_{ij} の 2 乗和を加えた

$$E = E_0 + \gamma \frac{1}{2} \sum w_{ij}^2 \quad (47)$$

を新たに誤差関数として誤差逆伝播アルゴリズムによる結合強度の更新式

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \quad (48)$$

を適用するのと定数項を除いて等価である。

枝刈り法は最適なネットワーク構造を探索するための手段として用いられることがある。枝刈り法のように、最初は全結合を作っておいて、あとで不要なものを刈り取ることは網膜から外側膝状体への投射、外側膝状体から第一次視覚野の間でも観測されている事実である。脳内でどのニューロンがどの役割を果たすかが出生直後から決まっているわけではない。出生後の環境によって柔軟に対応できるようにするためには、このほうが有利なのだとの解釈も成り立つ。

一方、小数の中間層ユニットから出発して必要に応じてユニットを追加していく方法も開発されている。この方法は、構成法 (constuructive method) あるいは動的ノード生成法 (dynamic node creation) という。誤差の減少幅が小さくなって収束しないときに新しい中間層を加えることによって実現される (Reed & Marks II, 1999)。

6.7 リカレントネット — 文脈情報、時系列情報の取り込み —

一般に時系列情報や文脈依存の情報を処理させるために考えられたネットワークとして単純回帰型ネットワーク (simple recurrent network, SRN) あるいはリカレントネットと呼ばれる回路がある⁶。これらの例は、提案者の名

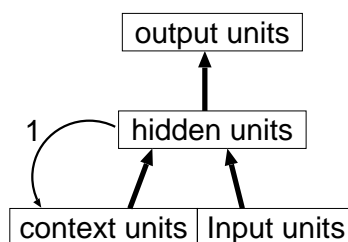


図 16: エルマンネット

前で エルマンネット (Elman net, 図 16), ジョーダンネット (Jordan net, 図 17) と呼ばれる。図では、ある層の一部が帰還信号 (recurrent feedback) を受け取るようになっている。このような帰還信号を受け取る素子を文脈層あるいは関連層と呼ぶ。

エルマンネットでは、入力層は入力信号を処理する入力ユニットと、直前の中間層の状態を入力とする文脈ユニットとで構成される。文脈ユニットは以前の中間層をコピーするためだけ (すなわち中間層から文脈ユニットへの結合強度は 1.0) である。結合強度の学習は順方向の結合についてだけ行われるので、通常の誤差逆伝播法がそのまま適用できる。

⁶時間的な順序関係を表現しようとしたものに 時間遅れネット (Time Delay Network) とよばれるものがある。

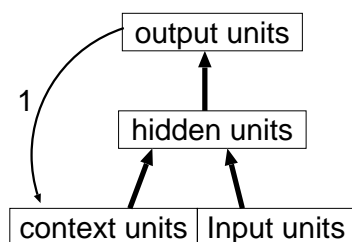


図 17: ジョーダンネット

ある時刻 t で処理される内容は、その時点での入力信号と、それ以前の時刻 $t-1$ までで処理された回路の状態を表す信号とを同時に処理する。すなわち、文脈層は過去の状態を記憶していることを意味する。この結果、ある時刻 t でのネットワークの状態は現在の入力と過去の入力履歴の集合によって決まることになる。例えば、図 18 において過去の影響を考えれば時刻 t における中間層の状態 $h(t)$ は

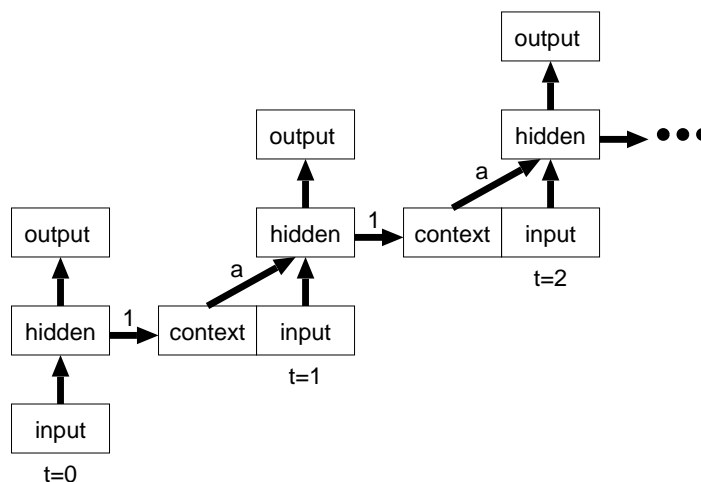


図 18: エルマンネットの時間発展

$$\begin{aligned}
 h(t) &= I(t) + ah(t-1) \\
 &= I(t) + a(I(t-1) + h(t-2)) \\
 &= I(t) + aI(t-1) + a^2I(t-2) + a^2h(t-3) \quad (49) \\
 &= \sum_{\tau=0}^T a^\tau I(t-\tau)
 \end{aligned}$$

と表すことができる。ここで $I(t)$ は時刻 t における入力を表している。文脈層からの影響 a が 1 より小さければ過去の入力からの影響が指数関数に従っ

て小さくなることを表している (1 より小さければ過去の影響は急速に小さくなる)。このことは、エルマンネットが一つ前の状態を保存しておくという単純な構造にもかかわらず、過去履歴に依存した出力を生じることを示している。

このような単純回帰型ネットワークの強力な性質を利用して、エルマンネットでは言語情報処理などへの応用が試みられている。一方、ジョーダンネットでは運動制御への応用が試みられている。エルマンネットとジョーダンネットの違いは、後の処理で利用する形式が出力層で表現される形がよいのか、中間層の形式の方がよいのかという違いである。例えば、入力信号を視覚情報、ジョーダンネットの出力を現在の手や腕の位置だとすると、現在の手や腕の位置と視覚情報とから、手や腕を操作して目標物を掴むための運動を制御する問題を解くことができるだろう。

さらにアトラクタネット (attractor net) と呼ばれるネットワークも提案されている。以下に英単語の意味を学習するためにヒントンとシャリス (Hinton & Shallice, 1991) によって提案されたモデルを示す。

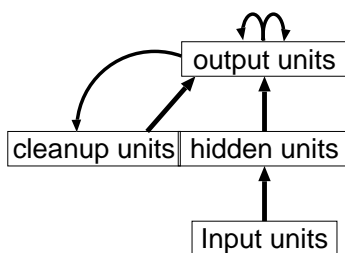


図 19: ヒントンとシャリスのアトラクタネット

一般にリカレントネットを数学的に解析するためには、出力ベクトル y に対して、文脈層からの結合係数行列を W とし時刻 t における入力を $I(t)$ と表すことにすれば

$$y(t) = Wy(t-1) + I(t), \quad (50)$$

と表すことができる。各素子への入力は互いに独立であると考えれば、 W の対角成分に $I(t)$ の対応する値を加えることで W と $I(t)$ をまとめて $W(t)$ と表すことができる。これにより、(50) 式は

$$y(t) = W(t)y(t-1), \quad (51)$$

と表現できる。(51) 式からリカレントネットの挙動は行列 $W(t)$ の固有値によって定まると言える。例えば、固有値が 1 より大きければ、対応する固有ベクトルの方向へ大きくなる。

6.8 動径基底関数ネット RBF

階層型ネットワークの最後に心理学モデルとして応用が考えられる RBF について触れる。

動径基底関数 (RBF: radial basis function) ネットとは、中間層ユニットの出力が基底関数 ϕ として定義されたネットワークである。

$$f(\mathbf{x}) = \sum_{i=1}^n w_i \phi(|\mathbf{x} - \mathbf{t}_i|). \quad (52)$$

基底関数としてガウシアン関数 $\phi(r) = \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ がよく用いられる。ガウシアン関数は釣鐘状の曲線であり、入力信号とガウシアン関数の中心からの距離 ($|\mathbf{x} - \mathbf{t}_i|$) に応じて出力が小さくなる。釣鐘の裾野の広さは σ^2 によって決まる。

2次元の RBF としてのガウシアン関数は、視覚情報処理 (Marr (1982)) においてしばしば登場するように心理学者にとって馴染み深いものであるが、高次認知過程のモデルとしても応用が考えられている。たとえば、カテゴリー判断では、刺激間の距離として心理量が考えられるような場合に、複数の刺激属性をネットワークへの入力とし、どのカテゴリーに属するかを RBF ネットワークに学習させれば、そのまま心理学的なモデルとなる。概念学習や表情認知研究などは、しばしばカテゴリー効果が現れる。カテゴリー効果を説明するために、任意のカテゴリー (あるいは顔や表情) の集合を一つの RBF として表現することになる。この場合ガウシアン関数のピークを与える刺激がプロトタイプとなり、もっとも典型性評定実験では高い値を得ることになる。

RBF の学習は、結合係数について微分するのではなく、上記のガウシアン関数を記述するパラメータそれぞれについて微分し、バックプロパゲーション法などを使って学習させればよい。

7 相互結合型のネットワーク

相互結合型のネットワークとはユニットが互いに結合しているネットワークである。各ユニットの活性値は時間とともに変化し、さまざまな状態を遷移することになる。このようなネットワークの振舞いは連想記憶や巡回セールスマン問題のような最適化問題などへの応用が試みられている。ここではアソシアトロンとホップフィールドモデルを取り上げ簡単に紹介する。ボルツマンマシン (Ackley, Hinton & Sejnowski, 1985; Hinton & Sejnowski, 1986a) については取り上げられなかったが、興味のある読者は参考文献をあたっていただきたい。

相互結合型のネットワークの単純な例として、図 20 ではルーメルハートら (1986) が例示したネッカーキューブ (Necker cube) の例を示した。それぞれ

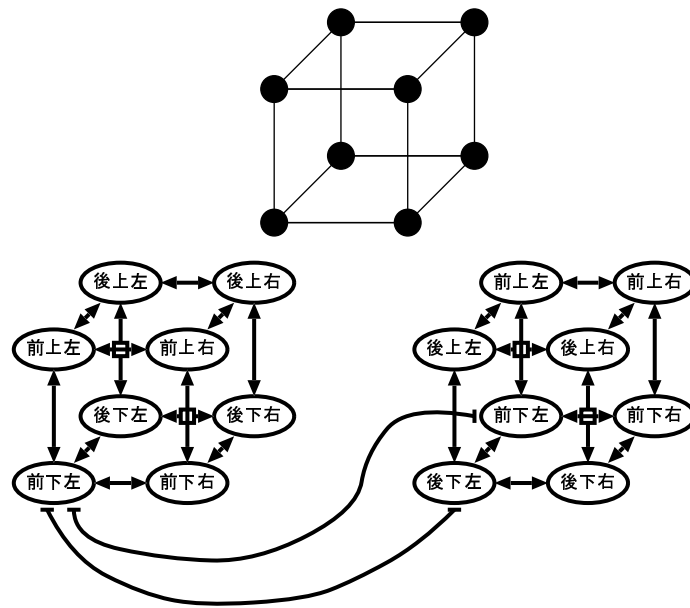


図 20: ネッカーキューブのニューラルネットワークモデル。ルーメルハートら (1986) 第 14 章より。各ユニットが頂点を意味する 16 個のユニットで構成されている。左右の見えの違いを頂点のラベルで表現されており、右の 8 個のユニットと左の 8 個のユニットがそれぞれ異なる見えを表している。正の結合を矢印 (\leftrightarrow) で、負の結合を (T) で表されている。一致した見えの間では興奮性の、矛盾する見えの間では抑制性の結合でニューロンが相互に結合されている。対応するラベル間と対応する位置間では抑制性の結合が組み込まれている。図ではすべての結合が描かれているわけではない。

のユニットの動作方程式に 4 節で示したような不応期の考え方などを導入すれば、左右の見えが突然反転するような動作をさせることも可能である。

相互結合型のネットワークを記憶装置と見た場合、自己想起型、連想記憶型の 2 種類に分けることがある。自己想起型のネットワークとは、予めネットワークに記憶させたパターンの一部を入力し、記憶させたパターンを復元することで実現される。心理学的には、曖昧な情報から記憶を検索するという意味での記憶検索のモデルとして有効であろう。一方、連想記憶型とは手がかり刺激を入力し、手がかり刺激と対になって記憶されている目標刺激を取り出すようなニューラルネットワークである。連想記憶型のニューラルネットワークは対連合学習のような記憶課題のモデルとなり得る。ただし自己想起型、連想記憶型いずれの場合でも、ニューラルネットワークモデルでは同じ構造が用いられる。例えば 4 ユニットからなるネットワークに $(1, 1, 1, 1)$ のようなパターンが記憶されている場合、 $(1, 1, 0, 0)$ を与えて $(1, 1, 1, 1)$ を想

起させると考えれば自己想起型であるが、4つのユニットを前後半2つに分けて考え、前半の(1,1)を手がかり刺激(キー刺激、あるいはキーベクトル)、後半部分を想起すべき内容(目標刺激、データベクトル)と考えれば対連合学習のモデルとなる。

7.1 同期更新と非同期更新

相互結合型のネットワークの更新方法として、同期的更新と非同期的更新の2種類が存在する。後述するホップフィールドのモデルは非同期更新であり、アソシアトロンは同期更新である。離散的時間変化($t = 0, 1, 2, \dots$)を考えて話を進めると、同期的更新とは時刻 t から時刻 $t + 1$ へ移るとき全ユニットの状態が同時に更新されることを意味し、非同期的更新とは任意の時刻では一つのユニットの状態が変化することをいう。

7.1.1 簡単な例

簡単のため図 21 のような 3 つのユニット x_1, x_2, x_3 が相互に結合している場合を考える。 x_1 と x_2 との結合が互いに抑制性(-1)であり、他は結合は興奮性(+1)である。任意の2つのユニット A, B を考えて、 $A \rightarrow B$ の結合係

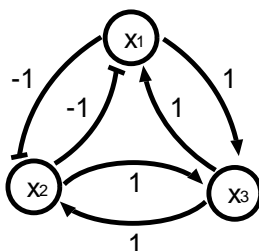


図 21: 簡単な相互結合のネットワーク

数の大きさが $B \rightarrow A$ と同じであるとき、対象結合という。対象結合でないユニットの組が一つでもあれば非対称結合のネットワークと言う。図 21 の結合係数は

$$W = \begin{pmatrix} 0 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}, \quad (53)$$

のように行列表現が可能である。 W の i 行 j 列目の要素 w_{ij} は j 番目のユニットから i 番目のユニットへの結合強度である。時刻 t における3つのユニットの状態を $(x_1(t), x_2(t), x_3(t))^T = \mathbf{x}(t)$ とすれば時刻 $t + 1$ における各ユニットの状態は $\mathbf{x}(t + 1) = W\mathbf{x}(t)$ と表現できる。対象結合であれば

$w_{ij} = w_{ji}$ が成り立つ。 W の対角要素がゼロ $w_{ii} = 0$ であることは自己結合が無いことを表している。

ユニットは 0 か 1 の 2 状態を取る 4.1 節のマッカロック・ピッツの形式ニューロンとする。このとき、ネットワークの状態は表 3 のとおり全部で 8 とおり存在する。

Table 3: 図 21 の全状態
状態

	状態							
x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1

各ユニットの状態が 0, 1 のような離散量で、かつ離散時間で表現された相互結合型のネットワークは、一般的な計算機モデルであるセルオートマトン (cellular automata) と類似の構造を持っていることが指摘できる。

7.1.2 同期的更新

図 21 で、同期的更新の場合には、(53) 式に図 3 の行列を右から掛ければ得られる。仮にしきい値が 0.1 だとすれば結果は以下ようになる。ユニッ

Table 4: 図 21 のネットワークの同期的更新。(しきい値を 0.1 にした場合)

時刻	素子	状態							
t	x_1	0	0	0	0	1	1	1	1
	x_2	0	0	1	1	0	0	1	1
	x_3	0	1	0	1	0	1	0	1
t+1	x_1	0	1	0	0	0	1	0	1
	x_2	0	1	0	1	0	0	0	1
	x_3	0	0	1	1	1	1	1	1

トの状態を (x_1, x_2, x_3) と表記し、すべての状態変化を矢印で表した状態遷移図を図 22 に示す。図 22 中の (000), (011), (101) はこの状態から動かないので不動点、または固定点 (fixed point) ということもある。状態 (001) と状態 (110) との間では循環することを意味し、このような状態をリミットサイクル (limit cycle) という。これらの状態は、一旦この状態になると他の状態を取ることができなくなるので、安定であるという。また、(010), (100), (111) は状態 (001) へ引き込まれるといいこれらの状態はリミットサイクルへの引き込み領域または流域 (basin) であるという。また、引き込まれる点 (または状

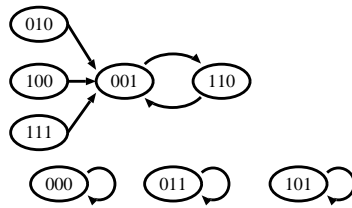


図 22: 図 21 で同期的更新の場合の状態遷移図

態) をアトラクタ (attractor) という。アトラクタには固定点、リミットサイクルの他に、複雑な挙動を示すカオスアトラクタと呼ばれるものも存在する。

7.1.3 非同期的更新

図 21 でしきい値を -0.1 に設定した場合の非同期的更新の状態遷移図を図 23 に示す。非同期更新では 1 度に 1 つの素子しか変化しないため、矢印で繋がっている状態間では 0 個または 1 個の素子だけが変化していることが分かる。図 22 と比較して大きな特徴は、全ての状態が矢印で結ばれていることと、矢印が右から左へと繋がっていることである。このことから、どのよう

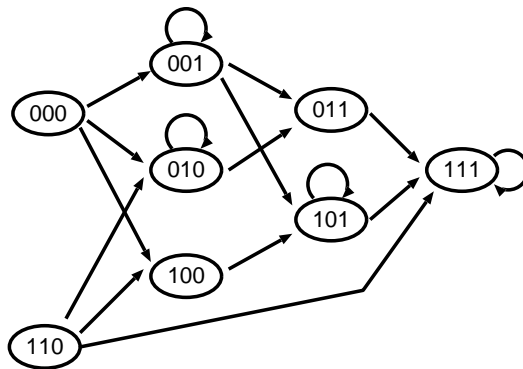


図 23: 図 21 で非同期的更新の場合の状態遷移図

な状態から出発しても、元の状態に留まるか、もしくは右側の状態へと遷移することがわかる。一旦より右側の状態へと遷移すると左側へと戻ることはない。あたかも地形図のように左側の状態は高く、右側は最も低いかのごとくであり、すべての状態は、最も低い状態 (111) へ向かって転がり落ちていくかのようなアナロジーを用いることができる。ホップフィールドモデルでエネルギー関数を導入する際に、この類推が正しいことを示すことにする。

(000), (110) のような状態は、初期値が (000) または (110) でない限りこれら状態へ遷移することがないという意味で“エデンの園”と呼ばれる。

7.2 アソシアトロン

7.2.1 連想記憶

私たちは「梅干し」から「すっぱい」を連想する。こうした連想をネットワークに記銘させる場合を考える。ベクトル $x = (1, 1, 1, 0)^T$ で「梅干し」を表し、 $y = (1, 0, 0)^T$ で「すっぱい」を表すとする。ここで x^T はベクトルの転置である。このとき $y = Wx$ をみたす行列 W が見つければ連想記憶と呼ぶことができる。ニューラルネットワークでは図 24 のように表現される。4次元ベクトルによって表現された入力パターン x から想起すべき3次元ベクトル y を取り出す過程と捉えることができ、記銘すべきパターンは入力層から出力層への結合強度として表現されていると見なすことができる。いま記名

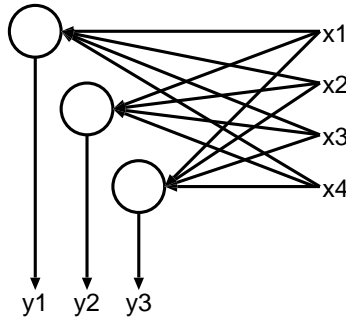


図 24: 連想記憶のニューラルネットワーク表現

すべきパターンが m 個あるとして連想パターン対を $(x^{(i)}, y^{(i)})$, $i = 1 \dots m$ と表現すれば、 j 番目の入力から k 番目の出力素子への結合強度は

$$w_{jk} = \sum_{i=1}^m x_j^{(i)} y_k^{(i)}, \quad (54)$$

と表現できる。実際に

$$\begin{pmatrix} (1, 0, 0, 0) \\ (0, 1, 0, 0) \\ (0, 0, 1, 0) \\ (0, 0, 0, 1) \end{pmatrix} \rightarrow \begin{pmatrix} (0, 0, 1) \\ (0, 1, 0) \\ (0, 1, 1) \\ (1, 0, 0) \end{pmatrix} \quad (55)$$

のような刺激-反応対を記憶する結合強度行列（あるいは結合係数行列という）は

$$W = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \quad (56)$$

となる。

自己想起の場合には、結合係数行列 m 行 m 列の正方行列になり、この行列のことを自己相関行列と呼ぶ。先の例では刺激ベクトルが直交していたために正しく想起できたが、互いに似通った記銘パターンが複数存在する場合には、似通ったパターンどうして誤想起が生じる。誤想起のことを雑音、またはクロストークと呼ぶ。連想記憶の数学的な解析については甘利 (1978) に詳しい。

想起を工夫して「りんご」→「まるい」→「すいか」→「あまい」→「チョコレート」のように想起を連続的に行なうことも可能である。このことを連鎖的想起あるいは動的想起と呼ぶ。連鎖的想起を応用すれば自由連想法などの心理モデルとなり得るだろう。

7.2.2 アソシアトロン

中野 (1979) のアソシアトロンは結合強度 (結合係数行列) が自己相関行列であり、自己想起、連想記憶、のどちらにも用いることができる記憶装置である。比較的単純な数学的構造をもっているため実現が簡単で実用性が高い。記憶のモデル、概念学習のモデルとしての応用が研究されている。

記銘すべき項目を n 次元列ベクトル $s = (s_1, s_2, \dots, s_n)$ で表現する。 s の各要素は $-1, 0, +1$ の 3 値をとり、 $+1$ と -1 とが意味を持ち、 0 は中立あるいは無意味なパターンとして扱われる。記名されるべき項目を表すニューロン間の結合係数は、 m 個のパターンを埋め込むとして前節の (54) 式で表される。この式と s の各要素は $-1, 0, +1$ しか取らないことから w の各要素も ± 1 または 0 となる。想起されるパターン \hat{s} は相関行列 w を用いて

$$\hat{s} = \phi(\phi(w)s), \quad (57)$$

と表される。関数 ϕ は以下のようなしきい値関数である。

$$\phi(x) = \begin{cases} +1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}, \quad (58)$$

自己想起の場合には、記銘されているパターンと類似した入力に対して記銘パターンが想起されることに対応し、一方連想想起では、入力パターンを手がかり (キーワード) と想起すべき内容 (ボディー) に分けてキーワードとボディー部分をすべてゼロにした入力ベクトルをアソシアトロンに入力し、ボディー部分に現われる情報を想起内容として取り出せば良い。

7.3 ホップフィールドモデル

ホップフィールド (Hopfield & Tank, 1985)⁷ モデルの特徴は、

⁷<http://dope.caltech.edu/>

1. ユニット間の結合強度 (結合係数行列) が対称である。 $w_{ij} = w_{ji}$ ただし $w_{ij} = 0$ すなわち自己結合係数は存在しない。
2. ユニットの状態変化は非同期的更新を用いる。すなわち一回に任意の一つの素子の状態しか変化しない。

である。すなわち 7.1.1 節で紹介した図 21 の非同期更新の例はホップフィールドモデルの最も簡単な例である。

n 個の 2 値ユニットを考える $x_i = 0, 1 (i = 1, 2, \dots, n)$ 。このネットワークの状態は 2^n 個存在し、幾何学的には n 次元超立方体の頂点に対応する。時刻 $t (= 0, 1, 2, \dots)$ におけるユニット i への入力を $u_i(t)$, 出力を $x_i(t)$ とすれば時刻 $t+1$ での出力 $x_i(t+1)$ は、結合荷重 w_{ij} としきい値 $\theta_i(t)$ を用いて次のように表現される

$$x_i(t+1) = \begin{cases} 1, & \text{if } u_i(t) > 0 \\ x_i(t), & \text{if } u_i(t) = 0 \\ 0, & \text{if } u_i(t) \leq 0 \end{cases} \quad (59)$$

$$u_i(t) = \sum_{j=1}^n w_{ij} x_j(t) - \theta_i(t) \quad (60)$$

すなわち、各ユニットの i は、他のユニット j からの入力 $x_j(t)$ の重み付き和 $\sum_{j=1}^n w_{ij} x_j(t)$ がしきい値 $\theta_i(t)$ より大きければ 1 を出力し、小さければ 0 を出力する。

7.3.1 エネルギー関数

ホップフィールドは、ネットワークの状態を表す次のエネルギー関数を導入し、

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i x_j + \sum_{i=1}^n \theta_i x_i, \quad (61)$$

式 (59) と式 (60) で定義される状態変化規則に従ってネットワークを動作させたとき式 (61) で定義されるエネルギー関数が必ず減少することを示した (図 23 も参照のこと)。

このことを確かめるために、ネットワークの状態が変化したときにエネルギー関数がどのように変化するかを調べてみる。エネルギー関数を k 番目のユニットに関する項とそれ以外の項に分けて、変形すると

$$E = -\frac{1}{2} \sum_{i \neq k}^n \sum_{j \neq k}^n w_{ij} x_i x_j + \sum_{j \neq k}^n \theta_j x_j - \frac{1}{2} x_k \sum_j w_{kj} x_j - \frac{1}{2} x_k \sum_i w_{ik} x_i + \theta_k x_k \quad (62)$$

いま、ある時刻 t から $t+1$ の間に k 番目のユニットの出力が

$$x_k(t) \rightarrow x_k(t+1) \quad (63)$$

に変化したものとする。このとき $\Delta x_k = x_k(t+1) - x_k(t)$ は 1 か -1 である。このような Δx_k の変化による状態変化に伴うエネルギー関数の変化 ΔE_k は、非同期的変化の仮定により k 番目のユニット以外は変化しないので

$$\begin{aligned} \Delta E_k &= -\frac{1}{2} \left(\sum_{i=1}^n w_{kj} x_j + \sum_{i=1}^n w_{ik} x_i \right) \Delta x_k + \theta_k \Delta x_k \\ &= - \left(\sum_{j=1}^n \frac{w_{kj} + w_{jk}}{2} x_j \right) \Delta x_k + \theta_k \Delta x_k \end{aligned} \quad (64)$$

となる。 $w_{ij} = w_{ji}$ であることに注意して

$$\Delta E_k = - \left(\sum_{j=1}^n w_{kj} x_j - \theta_k \right) \Delta x_k \quad (65)$$

と表される。右辺のカッコ内は u_k であるから

$$\Delta E_k = -u_k \Delta x_k \quad (66)$$

と書くことができる。式 (59) の状態変化規則から $\Delta x_k > 0$ のときは x_k が $0 \rightarrow 1$ に変化したことを表しているので $u_k > 0$ であるから $\Delta E_k < 0$ である。反対に $\Delta x_k < 0$ のときは x_k が $1 \rightarrow 0$ に変化したことになるので $u_k < 0$ だから $\Delta E_k < 0$ である。 $\Delta x_k = 0$ のときは $\Delta E_k = 0$ なので、全ての場合について

$$\Delta E_k \leq 0 \quad (67)$$

となる。

7.3.2 連想記憶

ホップフィールドは彼のモデルが連想記憶に適用できることを示している。ネットワークが記名すべきパターンベクトルを $0, 1$ ではなく $-1, 1$ をとるものとする⁸。ネットワークに記憶させたいパターン数を P 個、 s 番目のパターンを $x^s = (x_1^s, x_2^s, \dots, x_n^s)$ ($s = 1, 2, \dots, P$) とする。パターン s を記憶するとは、そのパターンに対するエネルギー関数を最小化することに相当する。しきい値を 0 としたときパターン x^s に関するエネルギー関数

$$E^s = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij}^s x_i^s x_j^s \quad (68)$$

⁸0, 1 の値をとる n 次元ベクトルは $w_{ij} = \sum_{s=1}^P (2x_i^s - 1)(2x_j^s - 1)$ にすればよい。

を最小化するもっとも簡単な方法は、 E^s が $x_i^2 x_j^2$ に依存するように w_{ij}^s を設定するばよい⁹。 $w_{ij}^s = x_i^s x_j^s$ とすれば

$$E^s = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij}^s x_i^s x_j^s = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (x_i^s)^2 (x_j^s)^2 \quad (69)$$

となる (相関行列)。

すべてのパターンについての結合係数は、

$$w_{ij} = \sum_{s=1}^P w_{ij}^s = \sum_{s=1}^P x_i^s x_j^s \quad (70)$$

によって近似的に求めることができる。記憶すべきパターンが似ていたり、パターンベクトルの次元数 n に対してパターン数 P の数が多すぎると正しき記憶できないことがある。このパターン間の相互干渉のことをクロストークという。ホップフィールドは記憶できるパターン数はユニット数の 15% 程度であることを示した。

相関行列を用いてホップフィールドネットの結合強度を決定する方法に対し、一般化逆行列 (generalized inverse matrix) の概念を導入し、クロストークを生じさせないようパターンを直交化して記憶する方法が提案されている。一般化逆行列にはいくつかの定義があるが、ムーア–ペンローズ Moore–Penrose の定義を用いることにすれば、

$$Z^+ = (Z^T Z)^{-1} Z^T \quad (71)$$

記名するパターンを $[\xi^1, \xi^2, \dots, \xi^P]$ のように並べてできる $N \times P$ 行列を X としたとき W は一般化逆行列を用いて以下のように求められる。

$$W = X X^+ = X (X^T X)^{-1} X^T \quad (72)$$

なおネットワークの状態変化を行なうには、 W を結合強度行列として (59) 式を用いればよい。

図 25, 図 26 は “浅”, “川”, “伸”, “一” の 4 文字をホップフィールドモデルによって想起させた例である。画像復元の様子を示した。

8 自己組織化

網膜と第一次視覚野との間には連続的な 1 対 1 対応が存在するのはよく知られた事実である (レティノトピー retinotopy)。鼓膜の周波数選択特性と第一次聴覚野の間にも対応関係が存在する (トノトピー tonotopy)。同様に体表面の感覚と体制感覚野の間にも対応関係が見られる (ソマトトピー somatopy)。

⁹ x_i は -1 か 1 だが、 x_i^2 は常に正

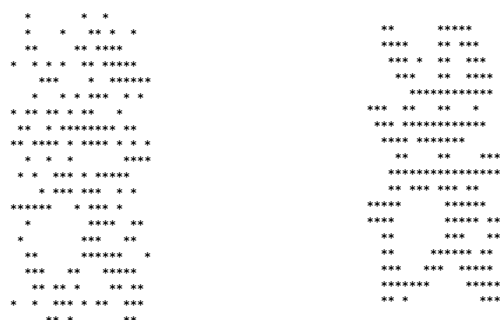


図 25: ホップフィールドモデルによる連想記憶の例。左が初期状態で 20×20 ピクセルで構成された「浅」の字の 100 画素分をランダムに反転させた画像。右は収束した結果

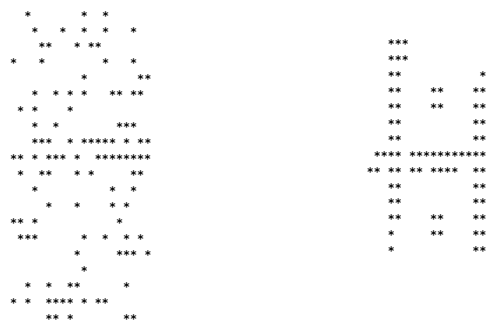


図 26: ホップフィールドモデルによる連想記憶の想起失敗例。左が初期状態で 20×20 ピクセルで構成された「一」の字の 100 画素分ランダムにを反転させた画像。右は結果。クロストークの影響が見られる。

すなわち感覚器官と第一次感覚野との間の神経結合は、類似した刺激に対して皮質上の同じような位置に対応する受容野を持つことが知られている。このような 2 つの神経場間の連続的な結合関係のことをトポグラフィックマッピング (topographic mapping) と言う。

視覚野のトポグラフィックマッピングについては、さらに細かいことが分っていて任意の視覚位置に対して、眼優位性 (ocular dominance)、方位選択性 (orientation selectivity)、色などの情報が処理されるように規則正しく配列されており、ハイパーコラム (hypercolumn) 構造という。ハイパーコラムは、2 次元的な皮質表面上に、2 次元の網膜位置、方位、視差情報 (立体視)、色情報処理などの多次元情報をなるべく効率よく処理しようとする生体情報処理の機構を表していると言える。このような構造は、大まかな構造は遺伝子によって決定されるが、細かい構造については神経回路の自己組織化 (self organization) によって達成されると考えられてる。一例としてリンズカー (1986a,b,c) の行なったシミュレーションを紹介する。

8.1 リンスカーのシミュレーション

リンスカーは図 27 のような数層の細胞で構成されるモデルを考えた。

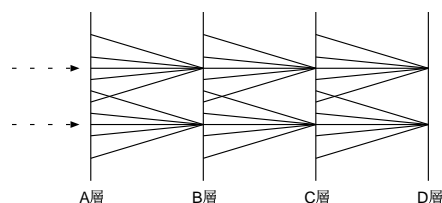


図 27: リンスカーのネットワーク。局所的なフィードフォワード結合を持つ 2 次元に分布した素子からなる。各素子に 5 つの入力があるように描かれている。実際のシミュレーションでは数百の結合が用いられた。

各層では、ユニットが 2 次元的に広がっている。B 層の各細胞は A 層のユニットから入力を受け取る。同様にして、B 層から C 層、C 層から D 層へと結合されている。

結合強度の変化はヘップ則の変形 (表 5) で入力と出力ともに活動度が高い場合には結合を強める他に、両方とも活動が低ければ結合の強さを弱めるような学習が行われた。

A 層における入力はランダムであった。その結果、A から B への結合では、結合の強さは正の大きな値になるか、負の大きな値になるかのいずれかであった。B 層では、隣り合う細胞が受け取る A 層の細胞の範囲は重なりあっている。したがって、この二つの細胞は A 層におけるパターンの似た部分を見ており、ある時刻に一方の細胞の活動度が高ければ、もう一方の細胞の活動度も高い可能性が高いと考えられる。すなわち、B 層において、近くに位置する 2 つの細胞の活動度の間には相関が生じている。

C 層の細胞の最終的な結合を調べると、中心に興奮性の領域が集まり、周辺には抑制性の領域が集まっていた。このことはランダムな結合とヘップ則だけを使って、オン中心型細胞が生じることが示されたことになる。さらに D 層には方位選択性の細胞が現れた。

リンスカーの提案したネットワークを追試するのは、非常に簡単なプログラムで可能である。にもかかわらず、第一次視覚野での細胞の特徴をよくとらえた優れたデモンストレーションと言える。すなわち、リンスカーのシミュレーションは初期視覚情報処理過程で見出された特徴分析器と定性的には同じものであると見なすことができよう。リンスカー (Linsker, 1988) はこの研究から更に進めて、情報理論との関連から最大情報量保存原理を提唱している。しばしば infomax と呼ばれるこの原理は統計的学習理論との関連への道を開いたという意味においても評価されるべき研究である。

なお、視覚野に見られる方位選択性を持つ細胞の受容野特性が形成されるか

モデル	Hebb 項		非 Hebb 項	備考
	シナプス後	シナプス前		
マルスバーク	発火頻度 η_j	発火頻度 η_k	拘束条件 $\sum_k \rho_{jk} = \text{一定}$	差分方程式の一ステップ毎に次のように規格化 $\rho_{jk} \rightarrow \frac{\rho_{jk}}{\sum_{k'} \rho_{jk'}}$
甘利	発火頻度 η_j	発火頻度 η_k	減衰項 $-\frac{1}{\tau} \rho_{jk}$	抑制性のニューロンの学習方程式と連立する
リンスカー	膜電位 ζ_j	膜電位 ζ_k	拘束条件 $-1 \leq \rho_{jk} \leq +1$	ρ_{jk} はシナプス伝達効率を表す $\rho_{jk} > 0$ 興奮性シナプス $\rho_{jk} < 0$ 抑制性シナプス
ミラー	膜電位 ζ_j	発火頻度 η_k	拘束条件 $0 \leq \rho_{jk} \leq +1,$ $\sum_{k'} \rho_{jk'} = \text{一定}$	解空間が $\sum_{k'} \rho_{jk'} = \text{一定}$ に制限するような射影演算子を方程式に含める
田中	膜電位 ζ_j	伝達物質放出量 $\rho_{jk} \eta_k$	拘束条件 $\rho_{jk} (1 - \sum_{k'} \rho_{jk'}) + \epsilon_{jk}$	非 Hebb 項はシナプス前後の栄養因子によるシナプスの安定化メカニズムを表す

Table 5: 代表的なモデルにおけるヘッブ則の変形。田中 (1988) より

についての数学的な研究も多数行われている (たとえば田中 (1988), Mackay & Miller(1990), Yamazaki & Asakawa(1999), Yamazaki(2000) など)。

8.2 主成分分析

自己組織化は、多次元刺激として与えられる情報を、その刺激が持つ規則性に従って 2 次元の皮質上へ照射する対応問題と見なすことができる。すなわち入力層の空間多次元多様体から 2 次元部分空間への写像である。このような立場から、統計学の用語を用いて説明を試みる。

通常の意味での主成分分析では列方向に変数 (p)、行方向にデータ数 (n)、となる行列を仮定する ($X = (x_{ij}), (1 \leq i \leq n, 1 \leq j \leq p)$) ことが多いが、ここではニューラルネットワークとの関連を考え通常の意味でのデータ行列の転置を入力データ行列 X とする ($X = (x_{ij}), (1 \leq i \leq p, 1 \leq j \leq n)$) 主成分分析とは適当な線形結合

$$\mathbf{y}_1 = w_{11}\mathbf{x}_1 + w_{12}\mathbf{x}_2 + \cdots + w_{1p}\mathbf{x}_p \quad (73)$$

$$= \mathbf{X}\mathbf{w}_1, \quad (74)$$

によって合成変量 y を最大化するベクトル w を見つけることである。このことは線形数学では固有値問題を解くことと同義である。合成変量 y の分散を最大化するには、データ行列 X の分散共分散行列の最大固有値に対応する固有ベクトルを求めればよい。

8.3 パーセプトロンモデル

上記の主成分分析を神経回路網の立場で表現する。

X は入力データセットで、 p 個のユニットからなる入力層に与えられる n 個のサンプルデータであると考え。これらのユニットから m 個の出力層 U

ユニットに全結合しているモデルを考える。

簡単のため出力層のニューロンが 1 個しかない場合 ($m = 1$) を考えると、 k 番目の入力パターンに対する出力層ユニットの出力は以下の式、

$$y_k = \sum_i^p w_i x_{ki} = (\mathbf{w} \cdot \mathbf{x}_k). \quad (75)$$

に従う、すなわち線形出力ユニットを考える。ここで、 w_i はパターン k が与えられたときの i 番目の入力層ユニット x_{ki} ($1 \leq i \leq p, 1 \leq k \leq n$) と出力層のユニットとの結合係数である。パターン k が与えられたときの i 番目の入力層ユニットから出力層ユニットへの結合強度 w_i が式 (76) のようなヘップ則

$$\Delta w_i = \eta y_k x_{ki}, \quad (76)$$

を用いて更新されたとすると、 w_i の漸化式は以下ようになる

$$\Delta w_i(t+1) = w_i(t) + \Delta w_i = w_i(t) + \eta y_k x_{ki} = w_i(t) + \eta \sum_i^p w_i x_{ki}, \quad (\text{for all } k). \quad (77)$$

w_i をまとめて \mathbf{w} とベクトル表現すれば

$$\Delta \mathbf{w}(t+1) = \mathbf{w}(t) + \Delta \mathbf{w} = \mathbf{w}(t) + \eta (\mathbf{x}_k \cdot \mathbf{w}) \mathbf{x}_k, \quad (78)$$

である。

学習が収束 (収束) した時点での $\Delta \mathbf{w}$ は 0 になる (すなわち結合係数の更新が行なわれない) ことが期待されるので、全入力パターンの平均を考えて

$$\mathbf{0} = \frac{1}{n} \sum_{k=1}^n \Delta \mathbf{w}^{(k)} = \eta \frac{1}{n} \sum_{k=1}^n (\mathbf{w} \cdot \mathbf{x}^{(k)}) \mathbf{x}^{(k)} = \eta \frac{1}{n} \mathbf{X}' \mathbf{X} \mathbf{w} \quad (79)$$

が成り立っていなければならない。ところが $\mathbf{X}' \mathbf{X}$ は、実対称行列であり、固有値はすべて正で固有ベクトルは直交する。すなわちヘップの学習則では有限回の学習によって解が求められない (実際には最大固有値に対応する固有ベクトルの方向に際限無く大きくなっていく)¹⁰。

そこで、式 (76) を修正して

$$\Delta w_i = \eta y_k (1 - y_k) x_{ki} \quad (80)$$

のように変形すると結合強度ベクトルは最大固有値の方向を向き、かつ収束することが Oja(1988) によって証明されている。すなわち、ヘップ則を使った自己組織化アルゴリズムを用いて入力データの統計的な性質をネットワークに学習させることができる。

¹⁰0 以外に解がない

その他にも、ヤングハウスホルダー変換のような操作などによって、ヘップの学習則を修正する方法が提案されている。出力ユニットが j 個 ($1 \leq j \leq m$) の場合は、

$$\Delta \mathbf{w}_{ij} = \eta \mathbf{y}_j \left[\left(x_j - \sum_k x_k w_k \right) \right] \mathbf{x}_i \quad (81)$$

を用いて更新すればよい (Sanger, 1989)。ザンガーの考え方は、線形数学というグラムシュミットの直交化をそのままニューラルネット上で実現したものととらえることができる。

8.4 コホネンの SOM

コホネン (Kohonen, 1997) は、ヘップ則を用いた主成分分析アルゴリズム (8.2 節を参照) と同様に、多次元刺激を 2 次元にマッピングするアルゴリズムを提案している。トリーズマン (Treisman & Gelade, 1980) の特徴地図の形成モデルと捉えることもできるし、入力刺激の分類器としての性質も考えることができる。

コホネンのアルゴリズムは競合学習 (competitive learning) の考え方を採用し、入力データに対して最大出力を与えるニューロン (とその近傍のニューロン) の結合係数のみを変更する学習則である。このように競合する素子の中で最大出力を与える素子のみが勝ち残ってその後の処理に影響を及ぼす方式をウィナー・テイク・オール (winner take all) 方式と呼ぶ。最大出力を与えるニューロン c は、

$$c = \underset{i}{\operatorname{argmin}} \{ |x - w_i| \}, \quad (82)$$

と定義される。上式はすべての結合強度ベクトル w_i のなかで x との距離が最小になる w_i が選択されることを意味する。結合ベクトルの大きさが 1 に正規化されていれば

$$|x - w_c| = \min_i \{ |w_i - x| \}, \quad (83)$$

と等価である。このようにして勝残ったユニットの近傍関数 h_{ci}

$$h_{ci}(t) = h(|r_c, r_i|, t), \quad (84)$$

を定義する。近傍関数には、たとえば次のようなガウシアン関数

$$h_{ci}(t) = \alpha(t) \exp \left(-\frac{|r_c - r_i|^2}{2\sigma^2(t)} \right), \quad (85)$$

が用いられることがある。コホネンの自己組織化地図 SOM (Self Organizing Map) とは、各入力刺激に対して最大出力を与えるユニット c と、その近傍関数 $h_{c,i}(t)$ で定義されるユニット群に対してヘップ則による学習を行い、類似した入力刺激の特徴が類似した場所に投射されるようにしたものである。

コホネンのアルゴリズム以下のようにまとめられる。

- 2次元状に配置した任意の数の素子を用意し、乱数で初期化する。各素子は入力刺激の次元数に応じた入力を受けとり、以下のような方法で各素子の結合係数を修正する。
- 与えられた刺激に対して (82) もしくは (83) で最大値を与えるユニット c を選択する
- 以下のような更新式

$$w_i(t+1) = w_i(t) + h_{ci}(t)[x(t) - w_i(t)], \quad (86)$$

で結合係数を更新する。

- すべての入力データについて 2, 3 を繰り返す
- 学習を収束させるために $\alpha(t)$ を単調減少関数とし、たとえば、以下のように定義する。

$$\alpha(t) = 0.9 \left(1 - \frac{t}{t_{\max}} \right) \quad (87)$$

- 2,3,4 を t_{\max} 回繰り返す。

近傍関数の広がりを決めるパラメータ $\sigma(t)$ を同様に t の関数として徐々に小さくしてゆく方法もある。

図 28 は 2 次元平面の点を入力データとして SOM に学習させた結果である。初期段階では中心付近に集まっていた結合係数が時間と共に徐々にほどこけて 2 次元平面の各点を反映するように広がる様子が見て取れる。コホネンの SOM は ftp://cochlea.hut.fi/pub/som_pak より手に入れることができる。興味のある読者は自身で実行してみることを勧める。

9 おわりに

9.1 本章で紹介した以外の話題

章の始めに書いたとおり、本章は講義録を元に行っている。参考までに紙面の都合で割愛した講義内容を記しておく、ボルツマンマシン (Hinton & Sejnowski, 1986b)、標準正則化理論 (Poggio, Torre & Koch, 1985) のニューラルネットワークの実現、いわゆる shape from X の話題。同期発火仮説に基づく振動子系に於けるトリーズマンの特徴統合理論 (Treisman & Gelade, 1980; Treisman, 1988) のニューラルネットワークの実現と反応時間のモデル。グロスパーグ (Carpenter & Grossberg, 1987) の適応共鳴理論 (ART: Adaptive Resonance Theory)。相互抑制回路を用いた短期記憶のニューラルネットワークモデルと力学系のアトラクタ。脳損傷とニューラルネットワークの破壊実験 (Plaut & Shallice(1993), Hinton & Shallice(1991)) などであっ

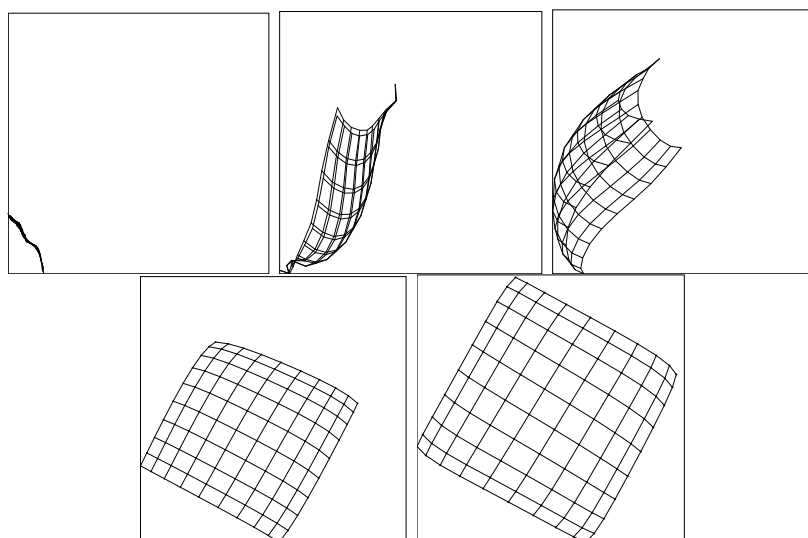


図 28: コホネンの SOM による二次元データの学習結果。左上より右下に向かって $t=20,100,200,400,800$ である。学習の進行に従って徐々に 2 次元格子が解けて行く。シミュレーションは ftp://cochlea.hut.fi/pub/som_pak のプログラムによった。

た。これらの話題は紙面の都合で割愛せざるを得なかったが、興味のある読者は章末の文献にあたって知識を深められたい。

9.2 神経回路網モデルのすすめ

元来、心理学と神経回路網モデルは密接な関係にある。本章で紹介したヘップは心理学者であるし、ローゼンブラットのパーセプトロン論文は *Psychological Review* に掲載されたものである。さらに、1986 年以降大ブームとなったバックプロパゲーション法の発案者ルーメルハートとマククリーランドも心理学者である。

加えて、日本には多数の優れた神経回路網研究者がいる。Biological Cybernetics, IEEE transaction of Neural Network, Neural Computation などの専門誌にはほぼ毎号日本人著者による論文が掲載されている。日本の神経回路網研究は世界のトップクラスなのである。甘利の数理解析、中野のアソシアトロン、福島の新コグニトロンなど、この分野のパイオニアを始めとして枚挙に暇が無い。書店にはこれらの研究者によって書かれたすぐれた教科書が多数並んでいて、世界のトップレベルの知識が母国語で手に入るのである。

このような有利な状況を利用した心理学研究が多いになされるべきであ

ろう。

文献

- 坂本, 石黒 & 北川 (1983). 情報量統計学. 東京: 共立出版.
- 甘利俊一, 村田昇 & Muller, R. (1997). 学習の数理モデル—汎化能力と過学習—. In 外山敬介 & 杉江昇 (Eds.), 脳と計算論 chapter 3, (pp. 37–53). 東京: 朝倉書店.
- 甘利俊一 (1978). 神経回路網の数理. 東京: 産業図書.
- 中野馨 (1979). アソシアトロン —連想記憶のモデルと知的情報処理. 東京: 昭晃堂.
- 田中繁 (1988). シナプス可塑性の数理モデル. In 甘利俊一 (Ed.), ニューラルネットの新展開 chapter 第6編第2章, (pp. 147–172). サイエンス社.
- Ackley, D. H., Hinton, G. E. & Sejnowski, T. J. (1985). A learning algorithm for boltzmann machines. *Cognitive Science*, 9, 147–169.
- Carpenter, G. A. & Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing* (pp. 54–115).
- Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D. & Plunkett, K. (1996). *Rethinking Innateness: A connectionist perspective on development*. Cambridge, MA: MIT Press. (邦訳「認知発達と生得性」, 乾, 今井, 山下訳, 共立出版).
- Hebb, D. (1949). The organization of behavior. In J. Anderson & E. Rosefeld (Eds.), *Neurocomputing* chapter 4. New York: MIT press.
- Hinton, G. E. & Sejnowski, T. J. (1986a). Learning and relearning in boltzmann machines. In J. L. McClelland, D. E. Rumelhart & T. P. R. Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Volume 1 chapter 7, (pp. 282–317). Cambridge, MA: MIT Press.
- Hinton, G. E. & Sejnowski, T. J. (1986b). Learning and relearning in boltzmann machines. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Volume 1 chapter 7, (pp. 282–317). Cambridge, MA: MIT Press.

- Hinton, G. E. & Shallice, T. (1991). Lesioning an attractor network: Investigations of acquired dyslexia. *Psychological Review*, 98(1), 74–95.
- Hopfield, J. J. & Tank, D. W. (1985). Neural computation of decisions in optimization problems. *Biological Cybernetics*, 52, 141–152.
- Kohonen, T. (1997). *Self-Organizing Maps Second Edition*. Springer. (邦訳「自己組織化マップ」、徳高、岸田、藤村訳、1996、シプリンガー・フェアラーク).
- Linsker, R. (1986a). From basic network principles to neural architecture: Emergence of orientation columns. *Proc. Natl. Acad. Sci. USA, Neurobiology*, 83, 8779–8783.
- Linsker, R. (1986b). From basic network principles to neural architecture: Emergence of orientation-selective cells. *Proc. Natl. Acad. Sci. USA, Neurobiology*, 83, 8390–8394.
- Linsker, R. (1986c). From basic network principles to neural architecture: Emergence of spatial-opponent cells. *Proc. Natl. Acad. Sci. USA, Neurobiology*, 83, 7508–7512.
- Linsker, R. (1988). Self-organization in a perceptual network. *IEEE transactions*, 1, 105–117.
- Mackay, D. J. C. & Miller, K. D. (1990). Analysis of linsker’s application of hebbian rules to linear networks. *Network*, 1, 257–297.
- Marr, D. (1982). *Vision*. W. H. Freeman: MIT Press.
- McCulloch, W. S. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115–133. in Neurocomputing, Anderson, J.A. and Rosefeld, E., Neurocomputing, MIT press, chapt 2, 1988.
- Minsky, M. & Papert, S. (1988). *Perceptrons, Expanded Edition* (2 Ed.). Cambridge, MA: MIT Press. パーセプトロン, 中野馨, 坂口豊訳, パーソナルメディア, 1993.
- Murata, N., Yoshizawa, S. & Amari, S. (1994). Network information criterion — determining the number of hidden units for an artificial neural network model. *IEEE Transactions on Neural Networks*, 5(6), 865–872.
- Oja, E. (1988). A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15, 267–273.

- Plaut, D. C. & Shallice, T. (1993). Deep dyslexia: A case study of connectionist neuropsychology. *Cognitive Neuropsychology*, *10*(5), 377–500.
- Poggio, T., Torre, V. & Koch, C. (1985). Computational vision and regularization theory. *Nature*, *317*, 314–319.
- Reed, R. D. & Marks II, R. J. (1999). *Neural Smithing: supervised learning in feedforward artificial neural networks*. Cambridge, MA: MIT Press.
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*, 386–408. In J. A. Anderson and E. Rosenfeld (Eds.) *Neurocomputing* (1988), MIT Press.
- Rumelhart, D. E., Smolensky, P., McClelland, J. L. & Hinton, G. E. (1986). Schemata and sequential thought processes in pdp models. In J. L. McClelland, D. E. Rumelhart & T. P. R. Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Volume 2 chapter 14, (pp. 7–57). Cambridge, MA: MIT Press.
- Sanger, T. (1989). Optimal unsupervised learning in a single-layer linear feed-forward neural network. *Neural Networks*, *2*, 459–473.
- Treisman, A. (1988). Feature and objects: The fourteenth bartlett memorial lecture. *The quarterly Journal of Experimental Psychology*, *40A*, 201–237.
- Treisman, A. & Gelade, G. (1980). A feature integration theory of attention. *Cognitive Psychology*, *12*, 97–136.
- Yamazaki, T. (2000). On a mathematical analysis of development of oriented receptive fields in linsker's model. *Research Report C-147, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology*.
- Yamazaki, T. & Asakawa, S. (1999). On a feedforward network model of orientation selectivities in v1 cells. In *Proceedings in European Conferene of Visual Processing 1999*. London: Elseiver.