



HOKKAIDO UNIVERSITY

Title	2004年度 混沌系工学特論講義ノート
Author(s)	井上, 純一; Inoue, Jun-ichi
Description	当講義資料は著者のホームページ http://chaosweb.complex.eng.hokudai.ac.jp/~j_inoue/ からダウンロードできます。 http://chaosweb.complex.eng.hokudai.ac.jp/~j_inoue/
Issue Date	2004
Doc URL	https://hdl.handle.net/2115/370
Rights(URL)	https://creativecommons.org/licenses/by-nc-sa/2.1/jp/
Type	learning object
File Information	konton2004_1.pdf, 第1回講義ノート



混沌系工学特論 配布資料 #1

担当：井上 純一 (情報科学研究科棟 8-13)

URL : http://chaosweb.complex.eng.hokudai.ac.jp/~j_inoue/

平成 16 年 10 月 25 日

～ 一般的な注意事項 ～

- この講義ではレポート問題を出します。講義ノート中に出てくる **問 #** がその問題です。これらの問題を各自考えてレポートにて提出して下さい。何題解いて提出するかは自由です。提出期限は最終講義まで。最終講義終了時に回収します。解答例は最終講義時に配布します。
- 講義ノート中に現れる **付録 (興味のある者は自習せよ)** は講義時間の関係で説明を省略する可能性の高い部分です。
- 各週に配布する講義ノートは必ずしもその週の講義分とは限りません。説明の終わらなかった部分は翌週に回しますので、講義ノートは忘れずに持参してください。

目次

1 脳の高次機能 — 連想記憶 —	1
1.1 ノイマン型コンピュータとの比較	2
1.2 生理学から数理モデルへ	3
1.3 単一ニューロンモデルの性能	6
1.4 連想記憶とは何か？	7
1.5 1つだけパターンを記憶した場合の想起	8
1.6 計算機シミュレーションで「感じ」をつかもう	10
1.7 多数のパターンを埋め込んだ場合の連想記憶	13
1.8 クロストーク項の統計的性質と最大記憶パターン	14

1 脳の高次機能 — 連想記憶 —

この講義でははじめに「連想記憶」と呼ばれる脳の高次機能を簡単な数理モデルを用いて解析し、その動作を調べていくことにする。この題材自体は比較的古いものではあるが、当講義の主題である簡単な構成要素が多数集まったときに、どのような機能が生じ、それがどのように情報処理に役立つのかということの説明する好例としてまずはこれを取り上げる。

1.1 ノイマン型コンピュータとの比較

ここでは脳の数理モデルを扱うわけであるから、ともあれ従来のコンピュータ(ノイマン型)と計算機としての脳の処理様式の比較をすることから始めよう。次に両コンピュータの仕様を載せる。

ノイマン型コンピュータ	脳
単位: プロセッサ	単位: ニューロン
演算速度: $\sim 10^8$ Hz	演算速度: $\sim 10^2$ Hz
シグナル/ノイズ $\sim \infty$	シグナル/ノイズ ~ 1
シグナルスピード $\sim 10^8$ m/sec	シグナルスピード ~ 1 m/sec
コネクション数 ~ 10	コネクション数 $\sim 10^4$
特徴: 直列演算 プログラム & データ 外部プログラミング	特徴: 並列演算 シナプス結合, ニューロン, 閾値 自己プログラミング (自己組織化), 適応
ハードウェアの欠陥が致命的	ハードウェアの欠陥に対してロバスト

以上をまとめると、脳(型コンピュータ)の利点は

- ノイズ, 欠陥に対してロバスト (脳の神経細胞は毎日死滅しているのに, 我々の情報処理能力はさほど変わらない).
- 外部から, C, Fortran, Pascal 等の言語を用いてプログラミングする必要なし \Rightarrow 「学習」(後述) によって外部に適応する能力あり.
- ファジー, 確率的, あるいはノイズを含む情報に対処することができる.

(例)

科学史上の重要人物とそのキーワードの対応表を覚え, それに答えるという状況を考えよう (例えばダーウィン \leftrightarrow 進化論, ニュートン \leftrightarrow 万有引力). このとき $E = mc^3$ と提示されたら, 皆さんはどう答えるだろうか? 多くの人「アインシュタイン」と答えるに違いない. $E = mc^2$ と正しくキーワードが提示されていないにも関わらず, である. これは我々の脳が $E = mc^3$ は何かの間違いで, 出題者は $E = mc^2$ のことを言っているに違いないと類推することができるからである.

- 高度な分散処理
- コンパクトである

一方, 不利な点は

- 一つの素子の処理速度は早くない.
- 一つ一つの素子レベルで見ると非常に誤差動が大きい.

などが挙げられるが, 脳全体としてみると, これらの不利な点を高度に分散化した情報表現と並列演算によって補っていると言える.

こうして見ると通常のコンピュータとはかなりの部分で異なっており, 工学的な応用という立場で見ただけには非常に魅力的な対象であることがわかる. そこで, 次に神経細胞を生理学的に概観し, それをどのように「数理モデル」化するかを見て行くことにしよう.

1.2 生理学から数理モデルへ

脳における実際の神経細胞の写真を図 1 に示す.

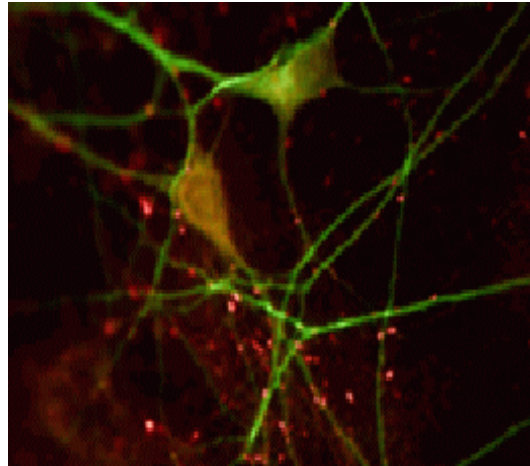


図 1: 実際の神経細胞.

これに対して, この神経細胞の模式図と各部位の名称を図 2 に載せる.

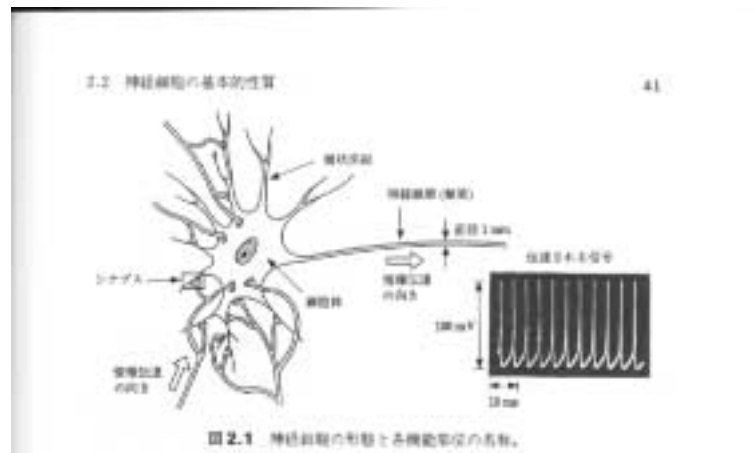


図 2: ニューロンの構成要素とその電圧特性. 「神経細胞が行なう情報処理とそのメカニズム」松本元, 大津展之共編, 培風館 (1991) p. 41 の図 2.1 から抜粋.

これらからわかることを要約すると, 各ニューロンはシナプスを介して他のニューロンに結合しており, 各ニューロンは信号を電気的なパルスとして伝える. 電気的なパルスの生成過程は次の様式に従う. 各シナプスはその結合部分に K^+ , Na^+ 等の正のイオンを透過させるチャンネルと Cl^- 等の負のイオンを透過させるイオンチャンネルを有し, このチャンネルを調節することにより, シナプス内部と外部の電位差 (通常は -70 mV 程) を調節することができる. 一般的には, 図のようにこの電位差がある値 (約 -55 mV) を越えると高さ 100 V 幅 1 ms の大きさのパルスが生成され, このパルスは軸索上を電気的な信号 (イオンの流れ) として伝わっていく. その後, パルスを生成したニューロンは約 10 ms の「不応期」を経て, また, 電位

差がある閾値を越えるとパルスを生成する.

これを素朴に数理モデル化すると図3のようになる.

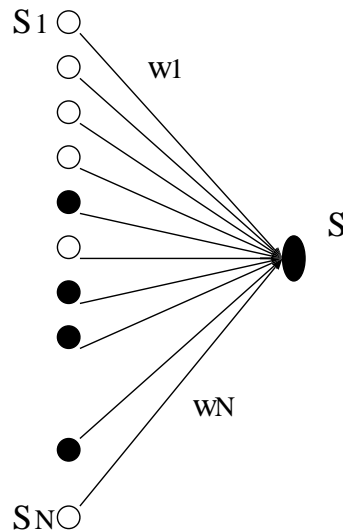


図3: 神経素子(ニューロン)の数理モデル. 各素子 $S_i (i = 1, \dots, N)$ は $S_i = 1$ (発火) あるいは $S_i = 0$ (静止) のどちらかの状態をとる. 注目するニューロン S の状態はこれらのニューロンからの重みつき和 $\sum_i w_i S_i$ がある閾値 θ を越えれば $S = 1$ 越えなければ $S = 0$ として決定される.

ここで, 各ニューロンは $S_i = 1$ (発火) か $S_i = 0$ (静止) の2つの状態のいずれかをとり, それぞれの値は注目するニューロンに重みつき和

$$h \equiv w_1 S_1 + w_2 S_2 + \dots + w_N S_N \quad (1)$$

として入力される. S はこの入力 h に対して

$$h > \theta : S \rightarrow 1 \quad (2)$$

$$h < \theta : S \rightarrow 0 \quad (3)$$

で決定される.

さて, ここでは注目する一つのニューロンの状態の決定がどのように行なわれるのかについて見たわけであるが, 図4のように, 他数のニューロンがつながっている場合にはどうなるであろうか?

この場合には, 任意のニューロン i, j 間の結合(シナプス結合)を w_{ij} とし, また i 番目のニューロンに関する閾値を θ_i とすると, 時刻 $t + 1$ での i 番目のニューロンの状態が, 時刻 t でのニューロン $S_j (j \neq i)$ の状態を用いて

$$w_{i1} S_1 + w_{i2} S_2 + \dots + w_{iN} S_N > \theta_i : S_i(t + 1) = 1 \quad (4)$$

$$w_{i1} S_1 + w_{i2} S_2 + \dots + w_{iN} S_N < \theta_i : S_i(t + 1) = 0 \quad (5)$$

で決定されるとして自然に拡張される. ところで, ここでは何も断りを入れずに「時刻」というものを導入したが, これはニューロン集団の状態変化が1ステップだけのものではなく, 上記のルールに従って

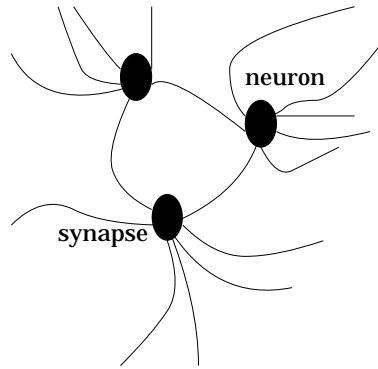


図 4: ニューロン同士をつなげる.

$S_i(0) \rightarrow S_i(1) \rightarrow S_i(2) \rightarrow S_i(t) \rightarrow S_i(t+1) \rightarrow \dots$ と変遷して行くものであるためである. 後に詳しく述べるが, 実はこのダイナミックスの固定点が「記憶」に相当する.

このように各ニューロンが時間的に変化するものであるとすると, その集団の時間変化の様式には大きく分けて次の 2 通りが考えられる.

- 状態変更則 (4),(5) が全てのニューロンに同時に適用される
 … 同期式状態変更 (paralell dynamics).
- ある時刻に 1 つだけのニューロンが状態更新の対象となる
 … 非同期式状態変更 (sequential dynamics).

現実の脳では各ニューロンが互いに「申し合わせて」状態を一斉に更新するとは考えづらいので後者の様式に従っているのではないかと予想される. 一方, 工学的な応用では前者をとる場合が比較的多い.

こうして, 例えばある時刻でのニューロン集団の状態 $\{S_i(t)\}$ はニューロン数が 9 の場合

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
t=0	1	1	0	1	0	0	1	0	0
t=1	1	0	0	1	0	1	1	1	1
t=2	0	0	1	1	1	0	0	0	1
t=3	1	0	0	1	1	1	1	0	1
t=4	0	1	1	1	0	0	1	0	1

となり, これを (発火; $S_i = 1$), (静止; $S_i = 0$) で表し, ニューロンの番地を

1 2 3
 4 5 6
 7 8 9

で表せば

t=0 t=1 t=2 t=3 t=4

のように表現される.

1.3 単一ニューロンモデルの性能

次の節からいよいよ「連想記憶」に入るが、この節ではその前に単一ニューロンの数理モデル — パーセプトロンの素子としての性能について少し触れておきたい。つまり、「構成要素が多数集まって実現される情報処理」の中の「構成要素」について調べてみようというわけである。ここでは簡単のため、2入力1出力閾値ありの神経素子

$$S = \Theta(w_1x + w_2y - \theta) \tag{6}$$

$$\Theta(x) = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \end{cases} \tag{7}$$

このとき、問題は w_1, w_2, θ を適当に選ぶことにより、論理命題 AND, OR, NOT が実現できるかどうか、という点である。答えを言ってしまうとこれらは可能である。下の表を参照されたい。

AND : $w_1 = w_2 = 1, \theta = \frac{3}{2}$ とすればよい。

x	y	$x \wedge y$	$x + y - \frac{3}{2}$	S
0	0	0	-3/2	0
0	1	0	-1/2	0
1	0	0	-1/2	0
1	1	1	1/2	1

OR : $w_1 = w_2 = 1, \theta = \frac{1}{2}$ とすればよい。

x	y	$x \vee y$	$x + y - \frac{1}{2}$	S
0	0	0	-1/2	0
0	1	1	1/2	1
1	0	1	1/2	1
1	1	1	1/2	1

NOT : $w_1 = -1, \theta = \frac{1}{2}$ とすればよい。

x	\bar{x}	$\bar{x} + \frac{1}{2}$	S
0	1	1/2	1
1	0	-1/2	0

なお, 排他的論理和 (XOR) は一つのパーセプトロンでは実現できないことがわかっているが, 中間層を導入した層状パーセプトロンにより実現できる.

1.4 連想記憶とは何か?

この節からはいよいよ連想記憶の話題に入る.

まず, 「連想記憶」というのは先の例でいうと「進化論」と聞いたときに「ダーウィン」と答えるようなものである. あるいは, 友人の写真を見せられたときにその友人の名前を答えるという状況を考えても良い. これらの各状況をもう少し一般的に言うと

「ヒント」が与えられたときにそのヒントを足掛かりに正解を見つけ出すこと

と言えそうである. ところで, この「ヒント」とか「正解」とかいうことを数式を用いて表すにはどうしたらよいのだろうか?

我々は神経回路網を用いたいわけであるから, N 次元超格子空間内の一点

$$(\xi_1, \xi_2, \dots, \xi_N) \quad (8)$$

を思い出すべき事項 — パターン — としよう. すると, 記憶すべき事項が p 個あった場合には上つきの添え字 μ を用いて

$$(\xi_1^\mu, \xi_2^\mu, \dots, \xi_N^\mu) \quad \mu = 1, \dots, p \quad (9)$$

とそれぞれのパターンを区別すればよい. このとき, パターン $\mu = 1$ に対する「ヒント」はパターン $(\xi_1^1, \xi_2^1, \dots, \xi_N^1)$ から少し離れた配列

$$(S_1, S_2, \dots, S_N) \quad (10)$$

を選べばよい.

(注) この節から, $\xi_i = \pm 1, S_i = \pm 1$ のバイナリ表現を用いることに注意されたい.¹

次に問題となるのはこの「ヒント」がどれだけ良いヒントであるかということである. これは簡単で, 上記 2 つのベクトルの内積

$$m \equiv \frac{1}{N} \sum_{i=1}^N \xi_i^1 S_i \quad (11)$$

を調べ, これが限りなく 1 に近ければ限りなく良いヒントになるだろうし, 逆にこれが限りなくゼロに近い場合にはヒントとしては意味のないものになるであろう. これで大まかな問題設定はできたので, 後は全節

¹ こうする理由は後に説明する「磁石」では $+1, -1$ を「小磁石」の N, S 極に対応させて数理モデル化するので, ここでもそれに倣っているに過ぎない. 深く考えず, 単に各ニューロン i で $S_i \rightarrow (S_i + 1)/2$ なる変換を施しただけと思えばよい.

までで説明された神経回路網を用いて $(S_1, S_2, \dots, S_N) \rightarrow (\xi_1^1, \xi_2^1, \dots, \xi_N^1)$ という正しい「連想の流れ」が実現できるかどうかという点が問題となる。つまり、全ての i に対して

$$S_i(t+1) = \operatorname{sgn} \left(\sum_{j=1}^N w_{ij} S_j(t) \right) \quad (12)$$

というダイナミクスを考えた場合、十分に時間が経過した後に ($t \rightarrow \infty$ の極限で)

$$\xi_i^1 = \operatorname{sgn} \left(\sum_{j=1}^N w_{ij} \xi_j^1 \right) \quad (13)$$

となり、我々の欲している $(\xi_1^1, \xi_2^1, \dots, \xi_N^1)$ がダイナミクスの固定点になりうるかどうかというところがポイントである。もし $(\xi_1^1, \xi_2^1, \dots, \xi_N^1)$ が無事に固定点になるならば、我々の目論みは成功したことになる。

では、何がこの成功・不成功をきめるのであろう？ (12) で自由に変えることのできそうなのは結合 w_{ij} である。実際この結合の選び方こそが、この神経回路網が連想記憶装置として機能するかどうかの鍵を握っている。次節ではこの w_{ij} の選び方 — Hebb 則 — を紹介しこの方法で果たして連想記憶が成り立つかどうかに関し、1つのパターンを記憶させるもっとも簡単な場合を例にとり、調べてみることにする。

1.5 1つだけパターンを記憶した場合の想起

ここでは一つのパターンを記憶した場合、そのパターンがダイナミクスの固定点にらるような w_{ij} の選び方について話しを進める。結論から先に言うと、任意の神経素子間の結合を

$$w_{ij} = \frac{1}{N} \xi_i^1 \xi_j^1 \quad (14)$$

のように選べば良い。実際にこのように選んだ場合、各 i に対して

$$\begin{aligned} \xi_i^1 &= \operatorname{sgn} \left(\frac{1}{N} \sum_{j=1}^N \xi_i^1 \xi_j^1 \xi_j^1 \right) \\ &= \operatorname{sgn}(\xi_i^1) \end{aligned} \quad (15)$$

となるが、これは成立しているのは明らかであろう。従って、(14) 式のように結合を選べば少なくとも一つのパターンの記憶に関しては神経回路網は連想記憶装置として機能することがわかった。

問1: 1個のパターンを記憶することができることが示されたわけであるが, 2個のパターンはどうだろうか? 1個のパターンの場合を拡張して数理的に議論せよ.

(ヒント)

(14) 式を2個のパターンに拡張して

$$w_{ij} = \frac{1}{N}(\xi_i^1 \xi_j^1 + \xi_i^2 \xi_j^2)$$

としてみる. これを(13)式に代入して, 各 i に対して

$$\xi_i^1 = \operatorname{sgn} \left(\xi_i^1 + \frac{\xi_i^2}{N} \sum_{j=1}^N \xi_j^2 \xi_j^1 \right)$$

が成立すれば, 2個のパターンを記憶し, その内のパターン ξ^1 を想起することができることになる. しかし, これが成立するかどうかは上式右辺括弧内の第2項の大きさを評価しなければわからない. 各 ξ_i^1, ξ_i^2 は独立かつ等確率に ± 1 をとる確率変数だから, この右辺括弧内第2項も確率変数となり, この変数の平均値の回りの揺らぎ — 標準偏差 — を評価し, それが $N \rightarrow \infty$ の極限でゼロになるかどうかを調べればパターン ξ^1 が想起できるかどうか判定できる.

1.6 計算機シミュレーションで「感じ」をつかもう

1つのパターンの記憶が可能であることがわかったわけであるから、2つ以上のパターンを記憶させた場合に任意のパターンが想起できるか否かが知りたくなる。しかし、2個程度ならともかく、例えば思い切って1000個ではどうなのか？等、多数個のパターンを Hebb 則でネットワークに埋め込んだ場合にシステムがどう動くのか、を解析的な議論に乗せるにはまだいくらかのギャップがありそうだ。このようなとき、計算機を用いた簡単な数値シミュレーションを行うとシステムの振る舞いを大雑把にはあるが把握することができる。もちろん、十分慎重にシミュレーションを行えば、解析計算に劣らないほどの信頼度で答えを出すこともできるが、以下に示すシミュレーションの目的は後に行う解析計算を始める前にある程度の「察し」をつける、「感じ」をつかむことであることに注意しておこう。

さて、ここで行うシミュレーションは具体的に (12) 式でみた状態更新式：

$$S_i(t+1) = \operatorname{sgn} \left(\sum_{j=1}^N w_{ij} S_j(t) \right) \quad (16)$$

を結合が次で与えられる一般化 Hebb 則：

$$w_{ij} = \frac{1}{N} (\xi_i^1 \xi_j^1 + \xi_i^2 \xi_j^2 + \cdots + \xi_i^p \xi_j^p) = \frac{1}{N} \sum_{\mu=1}^p \xi_i^\mu \xi_j^\mu \quad (17)$$

に選んで、全ての神経素子 S_i に対し、同時に上記の更新 (16) を行うことである（「同期的状態更新」）。このとき、時刻 t での回路網の状態： $S(t) = (S_1(t), S_2(t), \dots, S_N(t))$ と、今思い出そうとしているパターン： $\xi^1 = (\xi_1^1, \xi_2^1, \dots, \xi_N^1)$ との内積を回路網の神経素子数 N で割ったものを

$$m(t) \equiv \frac{1}{N} S(t) \cdot \xi^1 = \frac{1}{N} \sum_{i=1}^N S_i(t) \xi_i^1 \quad (18)$$

として定義し、これを重なり (overlap) と呼ぶことにしよう。この重なりを状態更新 (16) の各ステップで観察し、重なりが 1 に収束すればパターン： ξ^1 が想起できたとみなすわけである。ここで、今想起しようとするパターン（以下、「想起パターン」と略記する）： ξ^1 としては何でもよいのであるが、ここでは図 5 に示すような 2 次元の画像を選ぶことにする。このパターンは画像処理の分野で良く知られた「標準画像



図 5: ここで思い出そうとするパターン ξ^1 . サイズはやや小さめで 64×50 .

レナ」を閾値を用いて 2 値化したものである。しかし、パターン ξ^1 としては画像に限らず、何でも良いのであり、極端に言えば $\xi^1 = (1, 1, \dots, 1)$ と選んでも良いのである。ここでは視覚的に解りやすくするためにこのような特定の画像を選んだにすぎない。

さて、 ξ^1 以外の $p - 1$ 個のパターン ξ^μ ($\mu \geq 2$) はどのように選ぶかであるが、ここでは各成分にランダムに ± 1 を割り当てたものを採用することにしよう。すると、素子数 N が十分に大きければ任意の 2 つのパターン・ベクトル ξ^α, ξ^β は統計的にみて直交していることに注意されたい ($\xi^\alpha \cdot \xi^\beta = 0$ 。なぜこれが言えるか、各自考えてみるように)。

また、神経回路網の初期条件をどのように選ぶのかも重要であるが、ここでは想起パターン ξ^1 の各成分を確率 p で反転させたものを $S(0)$ に選ぶことにしよう。図 6 に 1 つのパターンを埋め込んだ場合の想起

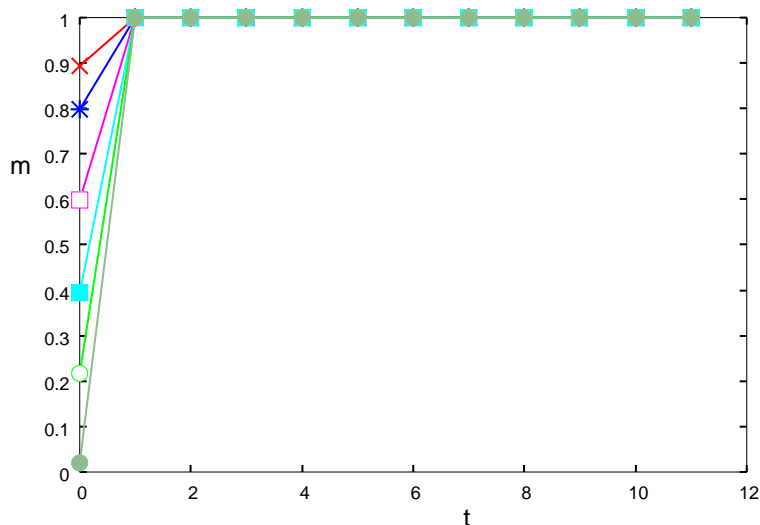


図 6: 1 つのパターンを埋め込んだ神経回路網の想起過程。

過程を示す。このシミュレーションでは神経素子数を $64 \times 50 = 3200$ と選んでいる。この図から明らかにように、たとえどんなに想起パターンから離れた初期状態から状態更新を始めても最後には想起パターンに収束していることがわかる。これは先に我々が見たとおりである。次に、 $p = 300$ 個のパターンを埋め込んだ場合の結果を図 7 に載せる。この場合には想起パターンにある程度近い初期状態からスタートすれば想起パターンに収束するが、想起パターンから遠い初期状態から出発すると想起パターンとは異なる状態 (偽記憶と呼ぶ) に収束してしまうという特性が見られる。

図 8 に初期状態を $m(0) = (1/N)S(0) \cdot \xi^1 = 0.6$ に選んだ場合の想起過程における画像の変遷を載せる。図 9 には初期状態を $m(0) = (1/N)S(0) \cdot \xi^1 = 0.2$ に選んだ場合の想起過程における画像の変遷を載せる。図 9 の場合、時間が経過しても想起パターンには近づかず、全く関係の無い状態に収束することがわかる。初期状態の選び方がまずかったというわけである。

最後に、 $p = 1000$ 個のパターンを埋め込んだ場合の想起過程を図 10 に載せる。この図より、明らかにどんなに想起パターンに近い初期状態からスタートしたとしても、神経回路の状態は想起パターンには近づかないことがわかる。あまりにも多くの情報を詰め込んだために、任意の想起パターンを回復できなくなってしまったわけである。このように Hebb 則に基づいて神経素子を多数つなげたシステムを構成しても、いくらでもパターンが記憶できるというわけではなく、ある限界が存在する。このような限界を詳細に調べることはシステム的设计上意味があることであろう。次節以降ではこの評価法について詳しくみていくことにしよう。

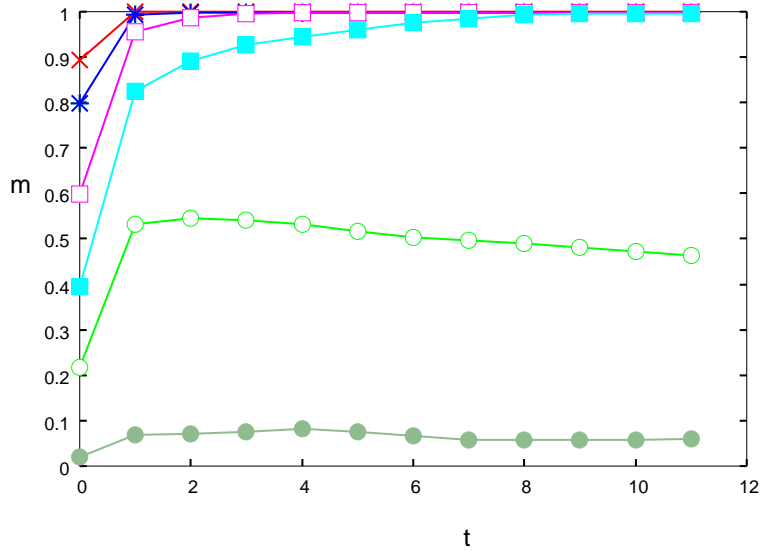


図 7: 300 個のパターンを埋め込んだ神経回路網の想起過程.

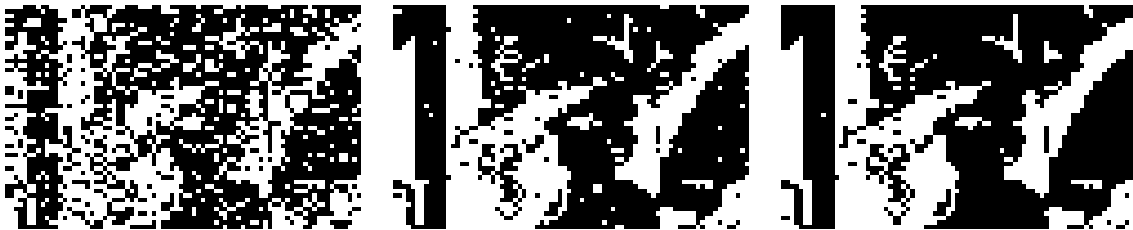


図 8: $p = 300$, 初期状態を $m(0) = (1/N)S(0) \cdot \xi^1 = 0.6$ に選んだ場合の想起過程. 左から $t = 0, 1, 12$ の時刻での回路網の状態.

問 2: ここで示したシミュレーションを実行せよ. 特に, 埋め込むパターン数 p と神経素子数 N の比 p/N と想起可能性との関係を明らかにせよ. なお, 当問題を解いて提出する際にはプログラムソースも添付すること.

(参考)

想起パターンとして, ここで用いたレナの 2 値化画像を用いる者は

http://chaosweb.complex.eng.hokudai.ac.jp/~j_inoue/KONTON2004/konton2004.html

からダウンロードできるので用いても良い (もちろん, 自分の好きな画像を用いても良い). また, 画像のフォーマット, 及び, xv, GIMP 等の画像処理ソフトの使い方は

http://chaosweb.complex.eng.hokudai.ac.jp/~j_inoue/LEC/ENS/inf_ens_2004.html

からの情報を参考にするとよい.

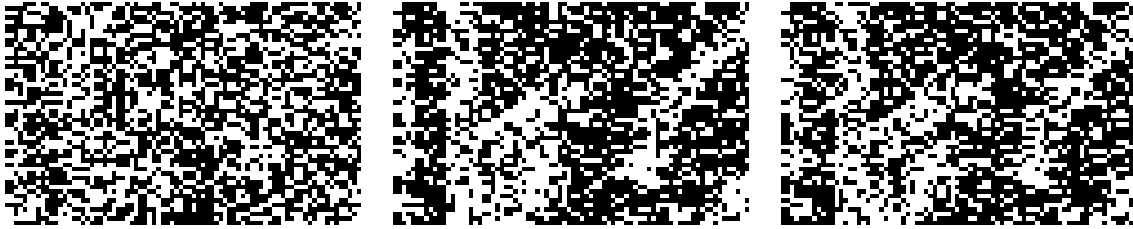


図 9: $p = 300$, 初期状態を $m(0) = (1/N)S(0) \cdot \xi^1 = 0.2$ に選んだ場合の想起過程. 左から $t = 0, 1, 12$ の時刻での回路網の状態.

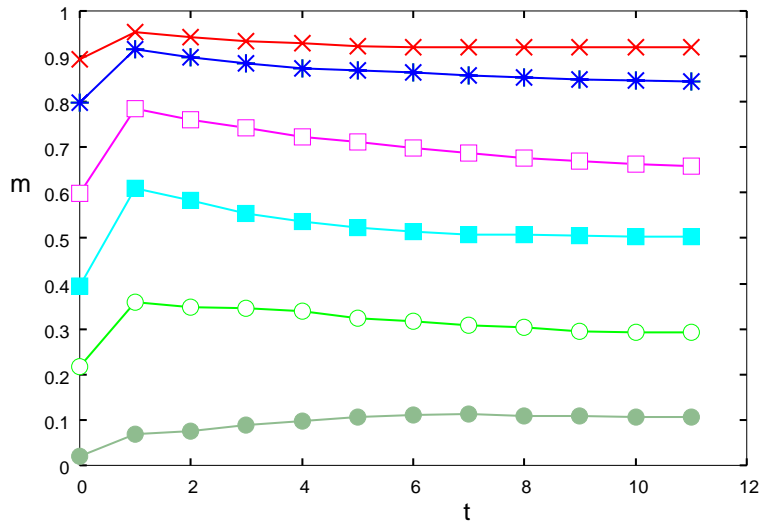


図 10: 1000 個のパターンを埋め込んだ神経回路網の想起過程.

1.7 多数のパターンを埋め込んだ場合の連想記憶

前々節では一つのパターンをネットワークに埋め込んだ (記憶させた) 場合に, そのパターンの想起の可能性を調べた. この節では, 多数個のパターンをネットワークに埋め込んだ場合どうなるかについて考察する.

前々節では Hebb 則 (14) を用いて神経回路網の「結合」に記憶を埋め込んだわけであるから, 前節の計算機シミュレーションの際に採用した拡張 Hebb 則 :

$$w_{ij} = \frac{1}{N}(\xi_i^1 \xi_j^1 + \xi_i^2 \xi_j^2 + \dots + \xi_i^p \xi_j^p) = \frac{1}{N} \sum_{\mu=1}^p \xi_i^\mu \xi_j^\mu \quad (19)$$

をここでもやはり用いて自然な拡張を図ろう. この場合, 注目するパターン $\xi^\nu = (\xi_1^\nu, \xi_2^\nu, \dots, \xi_N^\nu)$ の想起が安定である条件²

$$\xi_i^\nu = \text{sgn}(h_i^\nu) \quad (\forall_i) \quad (20)$$

$$h_i^\nu \equiv \sum_{j=1}^N w_{ij} \xi_j^\nu \quad (21)$$

² 前小節では注目するパターン (想起パターン) を 1 番目のパターンとしたが, ここでは ν 番目のパターンとしている. しかし, これを選んだ理由は特でない. 重要なのは, 注目するパターンとそれ以外のパターンを分離することであり, それ以外のパターンからの寄与が大きくなると, 注目しているパターンの想起が妨害されるということがポイントである.

が成立するかどうかを確かめてみよう. まず, h_i^ν に (19) を代入してみると

$$\begin{aligned} h_i^\nu &= \frac{1}{N} \sum_{j=1}^N \sum_{\mu=1}^p \xi_i^\mu \xi_j^\mu \xi_j^\nu = \xi_i^\nu + \frac{1}{N} \sum_j \sum_{\mu \neq \nu} \xi_i^\mu \xi_j^\mu \xi_j^\nu \\ &= \xi_i^\nu \left\{ 1 + \frac{\xi_i^\nu}{N} \sum_j \sum_{\mu \neq \nu} \xi_i^\mu \xi_j^\mu \xi_j^\nu \right\} \end{aligned} \quad (22)$$

となる. ここで, (22) の最右辺第 2 項³ がゼロならば, 前節で調べたものと全く同じになり, パターン ν は安定である.

従って, 想起が成功か失敗かに関して問題になるのは, 第 2 項の第 1 項へ及ぼす影響であり, 多数のパターンを埋め込んだ際にはこの項を何らかの方法で評価する必要がある.

そこで

$$C_i^\nu \equiv -\xi_i^\nu \frac{1}{N} \sum_j \sum_{\mu \neq \nu} \xi_i^\mu \xi_j^\mu \xi_j^\nu \quad (23)$$

とおき, この C_i^ν について詳しく調べてみることにする.

すぐにわかることは, もし, この C_i^ν が

$$C_i^\nu > 1 \quad (24)$$

を満たすならば (22) 式より, a を正の定数として

$$h_i = -a\xi_i^\nu \quad (25)$$

が成り立ち, 結局, パターン ξ_i^ν の安定条件は

$$\xi_i^\nu = \text{sgn}(-a\xi_i^\nu) \quad (26)$$

となるので満たされない. 従って, 条件 (24) が想起に失敗する条件と言えそうだ. しかし, 多少厄介な問題として, この C_i^ν はただの数ではなくて「確率変数」であるという点がある. 従って, 我々が以下にできることと言えば, この確率変数 C_i^ν の統計的性質 (どのような分布に従う確率変数なのか) を調べ, この変数のとる値が $C_i^\nu > 1$ を満たす確率を具体的に求めていくことである.

1.8 クロストーク項の統計的性質と最大記憶パターン

この節では C_i^ν を精密に評価する.

まず, 各 ξ_i^μ がランダムに ± 1 をとるランダムパターンを考えると, C_i^ν はこれらの和であるから, 中心極限定理より, 平均がゼロ, 分散 σ^2 が

$$\begin{aligned} \sigma^2 &= \left(-\frac{\xi_i^\nu}{N} \right)^2 \sum_j \sum_{\mu \neq \nu} (\xi_i^\mu \xi_j^\mu \xi_j^\nu)^2 \\ &= \frac{1}{N^2} \sum_j \sum_{\mu \neq \nu} 1 = \frac{1}{N^2} \times N \times (p-1) \simeq \frac{p}{N} \end{aligned} \quad (27)$$

³ この項をクロストーク項 (cross talk term) あるいはクロストークノイズ (cross talk noise) と呼ぶ

であるガウス分布 (正規分布) に従う.

つまり, $C_i^v \equiv x$ とおくと, x は

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (28)$$

に従う確率変数である. 従って, ξ_i^v が安定な固定点ではなくなる確率 (想起に失敗する確率) P_{error} は

$$\begin{aligned} P_{\text{error}} &= \text{Prob}(C_i^v > 1) \\ &= \text{Prob}(x > 1) \\ &= \frac{1}{\sqrt{2\pi}\sigma} \int_1^{\infty} e^{-\frac{x^2}{2\sigma^2}} dx \\ &= H\left(\sqrt{\frac{N}{p}}\right) \end{aligned} \quad (29)$$

となる.

ただし,

$$H(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} du e^{-\frac{u^2}{2}} \quad (30)$$

で補誤差関数を定義した (図 11 参照). 従って, 例えば $P_{\text{error}} = 0.01$ を満たすような p , つまり,

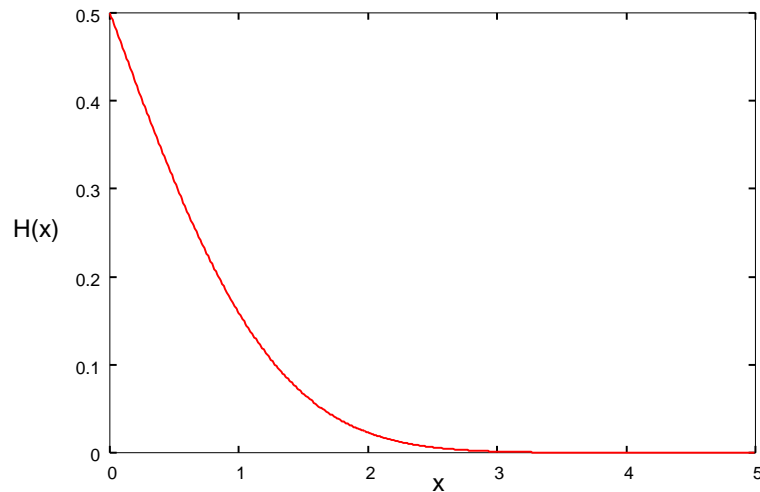


図 11: 補誤差関数 $H(x)$ の形状.

$$H\left(\sqrt{\frac{N}{p}}\right) = 0.01 \quad (31)$$

を満たす p を求めれば, 想起に失敗する確率が 0.01 以内に収まるような最大の埋め込みパターン数 p_{max} が求まるわけである.

実際に補誤差関数の数値テーブルを用いて求めた p_{max} をいくつかの P_{error} について算出したものを表にしておく.

P_{error}	p_{max}/N
0.001	0.105
0.036	0.138
0.01	0.185
0.05	0.37
0.1	0.61

この表より, 許容度を $P_{\text{error}} = 0.01$ に選ぶと

最大記憶パターン数 : $p_{\text{max}} = 0.185N$

までのパターンが安定に埋め込めることになる (つまり, 全素子数の 2 割ほどのパターンが記憶できる).

付録 (興味のある者は自習せよ)

ところで, 以上の議論は i 番目のビット ξ_i^ν にのみ着目し, そのビットの安定性を調べたわけであるが, 実際には $\xi^\nu = (\xi_1^\nu, \dots, \xi_N^\nu)$ の N ビット全てに関する安定性を調べるべきではないだろうか? つまり, この場合

$$\prod_i (1 - P(C_i^\nu > 1)) = (1 - P_{\text{error}})^N \tag{32}$$

が $1 - 0.01 = 0.99$ より大きい, すなわち, 不等式

$$(1 - P_{\text{error}})^N > 0.99 \tag{33}$$

が許容度 0.01 での最大埋め込みパターン数 p_{max} を決定する.

ここで, 2 項定理を用いると

$$\begin{aligned} (1 - P_{\text{error}})^N &= \sum_{r=0}^N {}_N C_r (-P_{\text{error}})^r 1^{N-r} \\ &\simeq {}_N C_0 1 + {}_N C_1 (-P_{\text{error}}) = 1 - NP_{\text{error}} \end{aligned} \tag{34}$$

となるので, 条件 (33) は

$$P_{\text{error}} < \frac{0.01}{N} \tag{35}$$

と書き直せる.

さて, p のオーダーが N より小さく, $N \rightarrow \infty$ の極限で $\sqrt{N/p} \rightarrow \infty$ となる場合を考えよう. このとき, $x \rightarrow \infty$ で誤差関数が

$$1 - \text{erf}(x) \simeq \frac{e^{-x^2}}{\sqrt{\pi}x} \tag{36}$$

で振舞うことに注意すると, 条件 (35) は

$$-\log 2 - \frac{N}{2p} - \frac{1}{2} \log \pi - \frac{1}{2} \log(N/2p) < \log(0.01) - \log N \tag{37}$$

となるから, $N \rightarrow \infty$ のときの主要項のみ残すと

$$p < \frac{N}{2 \log N} = p_{\max} \quad (38)$$

となり, この場合の最大記憶パターン数 p_{\max} は

$$\text{最大記憶パターン} : p_{\max} = \frac{N}{2 \log N}$$

となる⁴.

⁴ p のオーダーが N より小さく, $N \rightarrow \infty$ の極限で $\sqrt{N/p} \rightarrow \infty$ であるとした我々の仮定は正しかったことに注意されたい.