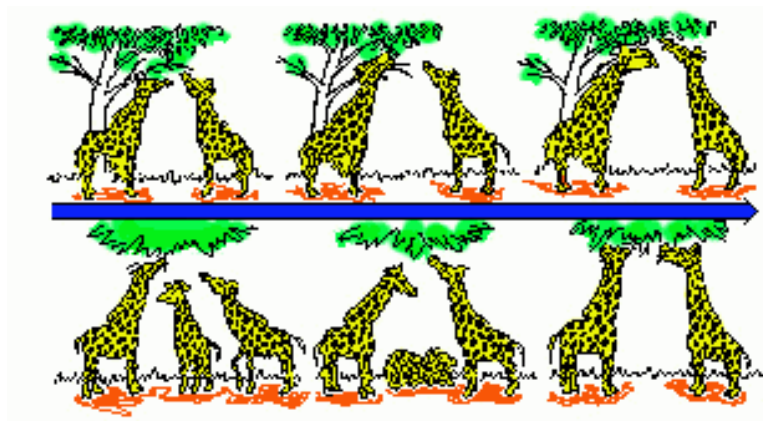




学習と確率の探索

東京大学大学院
情報理工学系科学研究科
電気情報学専攻
伊庭斉志



新しいAI

■ 古典的AI

- 論理、TMS、探索、...
- GOFAI: Good Old Fashioned AI 「古きよき時代のAI」



■ New AI

- 進化計算: GA, GP
- 強化学習
- ニューラルネットワーク
- 複雑系という知能
- 人工生命

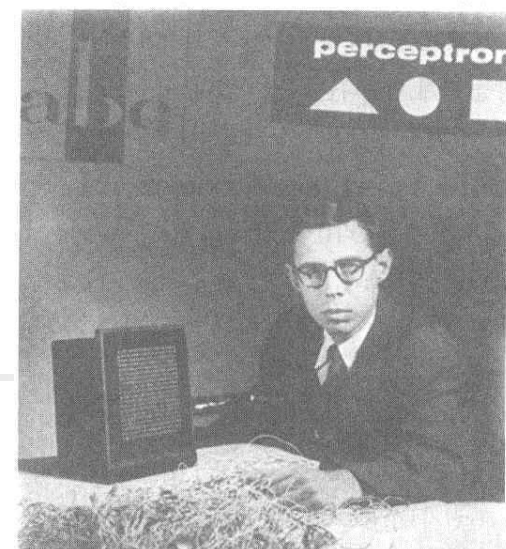




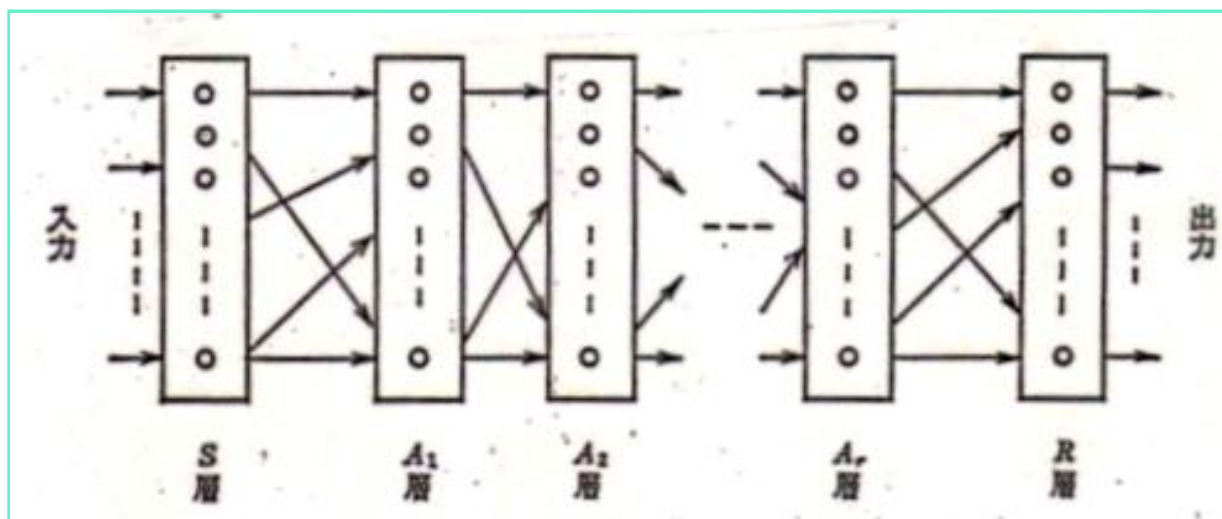
確率的探索法

- 決定的アルゴリズム
 - 従来のアルゴリズム e.g., ソート、ハッシュ ...
- 確率的アルゴリズム : 非決定的なアルゴリズム
 - ニューラル・ネットワーク ⇒ 脳・神経回路網のモデル
 - 焼きなまし法 ⇒ 物理現象のモデル
 - 強化学習 ⇒ パブロフの犬のモデル
 - 遺伝的探索法 ⇒ 進化現象のモデル
- 物理的あるいは生物的現象をモデルにしている.

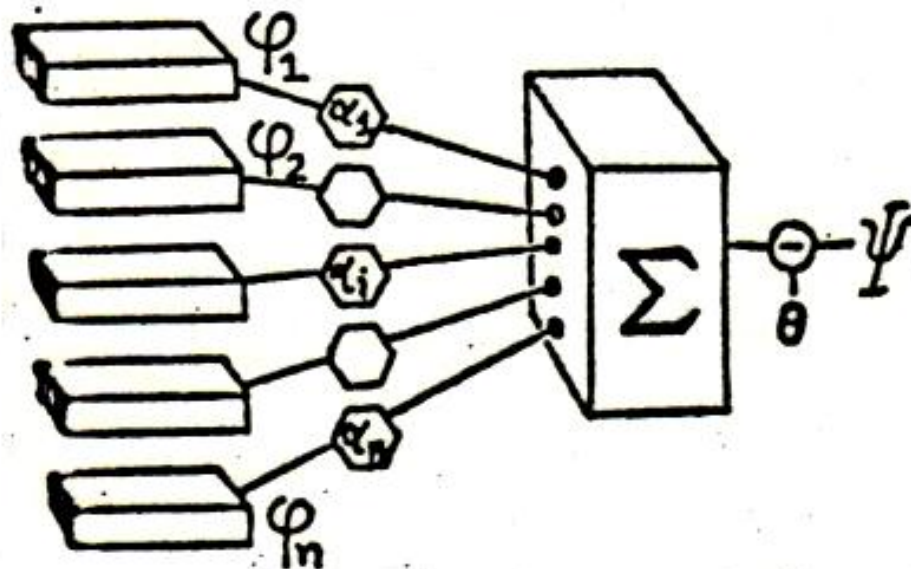
パーセプトロン



- Frank Rosenblatt (1961)
- 学習能力のある神経素子を構成要素とする多層の層状回路
- 脳の学習モデルとして注目される



単純パーセプトロン



「 $\sum_{\varphi \in \Phi} \alpha_{\varphi} \varphi(X) > \theta$ のとき, かつそのときに限り $\psi(X)=1$ 」

$$\alpha_{\varphi_1} \varphi_1(X) + \dots + \alpha_{\varphi_n} \varphi_n(X) > \theta$$

論理関数の集合を $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$ とする.

単純パーセプトロンの初期の成果

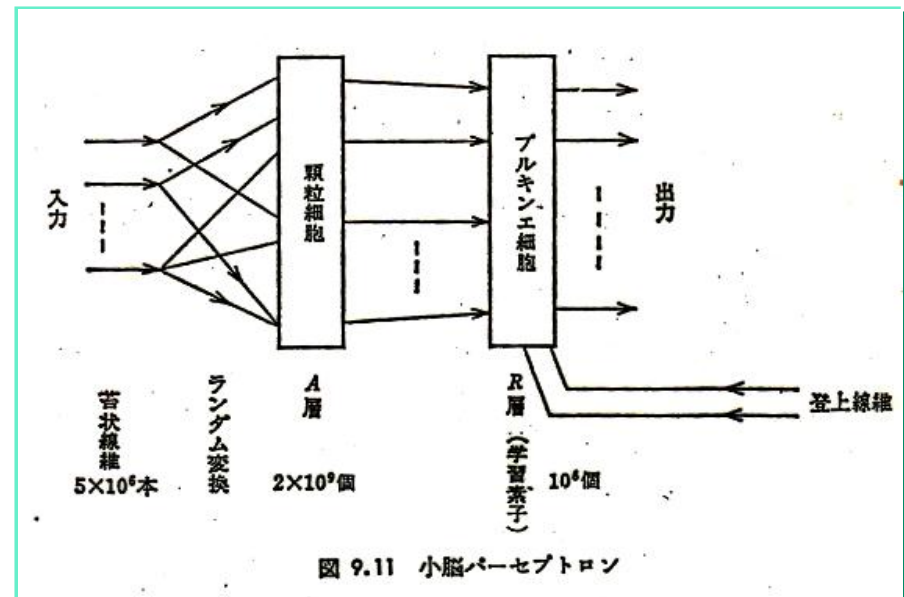
- 収束性定理

- 線形分離可能性を仮定

- Nilsson(1965), Minsky & Papert(1969)

- 小脳のパーセプトロン説

- Marr & Albus





線形分離可能性

- 線形モデルによる2値分類

$$y(\mathbf{x}) = f(\mathbf{w}^T \phi(\mathbf{x}))$$

\mathbf{x} : サンプル

$\phi(\mathbf{x})$: 特徴ベクトル

\mathbf{w} : 重みベクトル

$$f(a) = \begin{cases} +1, & a \geq 0 \\ -1, & a < 0 \end{cases}$$

バイアス要素 $\phi_0(\mathbf{x}) = 1$

重みベクトルと特徴ベクトルの内積をとって、それがゼロ以上であれば +1（正例と判定）、そうでなければ -1（負例と判定）を返す関数



パーセプトロン学習アルゴリズム

1. 重みベクトルを要素0で初期化
2. 学習データからランダムにサンプルを選択
3. 分類が間違っていたら以下の式で重みベクトルを更新
 - 正例の場合 $\mathbf{w} \leftarrow \mathbf{w} + \phi(\mathbf{x})$
 - 負例の場合 $\mathbf{w} \leftarrow \mathbf{w} - \phi(\mathbf{x})$
4. すべてのサンプルを正しく分類できるまで 2 に戻り繰り返す

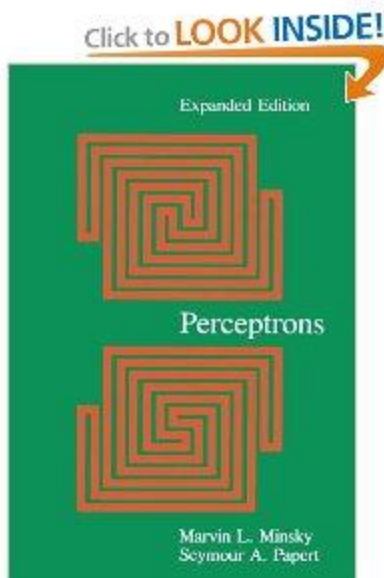


パーセプトロンの収束定理

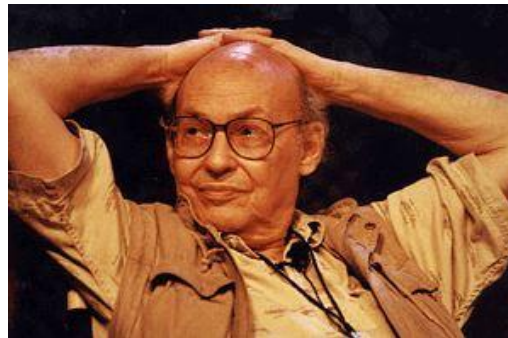
- 正例と負例が**線形分離可能**であるならば、前述のアルゴリズムにより、それらを分離する超平面を有限のステップで見つけられることが保証されている。
- データが線形分離可能でないならば、パーセプトロン学習は収束しない
- 線形分離可能である場合でも、ステップ数が非常に大きくなることもある

連結性の判断はできるか？

定理 0.8: 「任意の幾何学的図形が連結されているかどうかを判定する有限直径パーセプトロンは、存在しない。すなわち、有限直径のパーセプトロンは、 ϕ 連結 を計算することはできない。」



Marvin Minsky & Seymour Papert (1969).
Perceptrons, MIT Press, Cambridge, MA.



Marvin Minsky, 1927/8/9 -



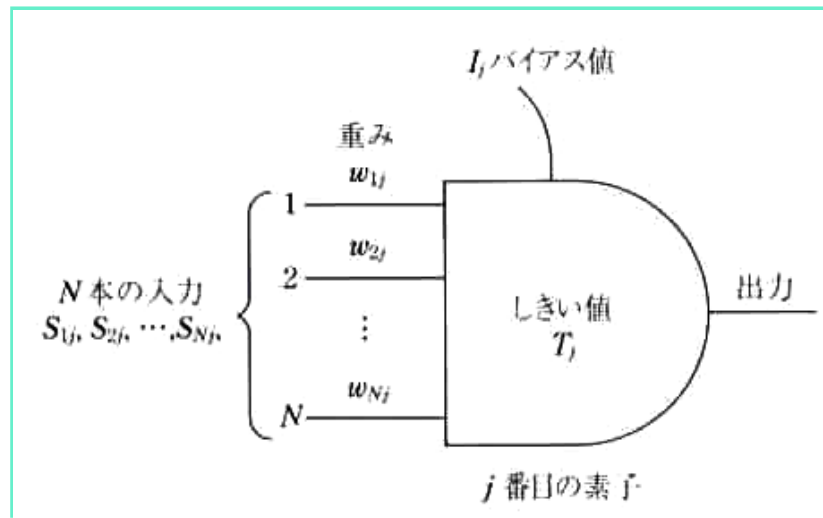
単純パーセプトロンの限界

- 識別できる対象に限界がある
 - 線形分離可能性から来る制約
 - X-orの判別ができない
 - 有限直径パーセプトロンから来る制約
 - 強連結性の判断ができない
 - パリティ(偶奇性)の判別が不能
 - このため膨大な数の素子を必要とするので実用的でない

課題: X-OR, Parityができないことを平易な言葉で説明する。

ニューラルネットワーク

- ノード＝神経回路網の素子（ニューロン）
- 線形のしきい値による判別関数



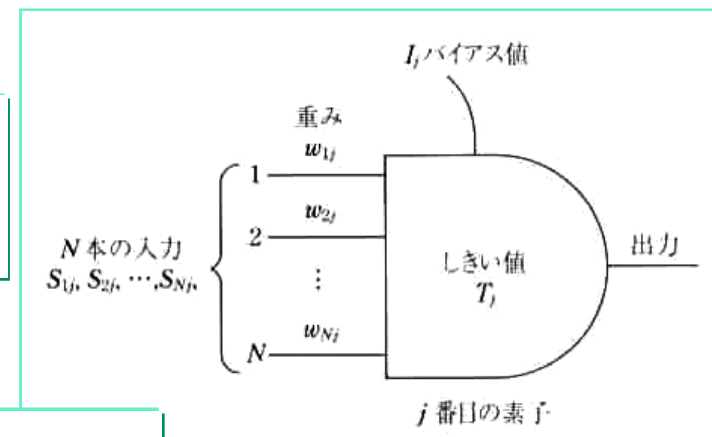
- j 番目の素子への N 個の入力 $S_{1j}, S_{2j}, \dots, S_{Nj}$
- おのおのの重みが $w_{1j}, w_{2j}, \dots, w_{Nj}$

ニューラルネットワーク

- バイアス値 I_j : 常に入力わる.
- 得られた入力(重み付き和+バイアス値)がしきい値(T_j) より大きい場合に1, そうでない場合0 を出力する.

- つまり、

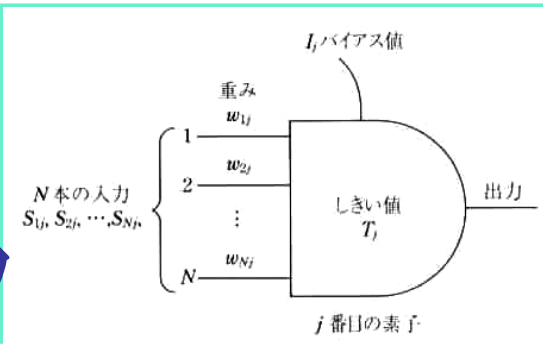
$$U_j = \sum_{i=1}^N S_{ij} w_{ij} + I_j - T_j$$



- に対して、

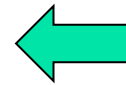
$$\text{出力値}_j = \begin{cases} 1 & U_j > 0 \text{ のとき} \\ 0 & U_j \leq 0 \text{ のとき} \end{cases}$$

ニューラルネットワーク



- より一般的には、
- 関数 f としては、

$$\text{出力値}_j = f(U_j)$$



$$U_j = \sum_{i=1}^N S_{ij} w_{ij} + I_j - T_j$$

1. ステップ関数

$$f(x) = \begin{cases} 1 & x > 0 \text{ のとき} \\ 0 & x \leq 0 \text{ のとき} \end{cases}$$

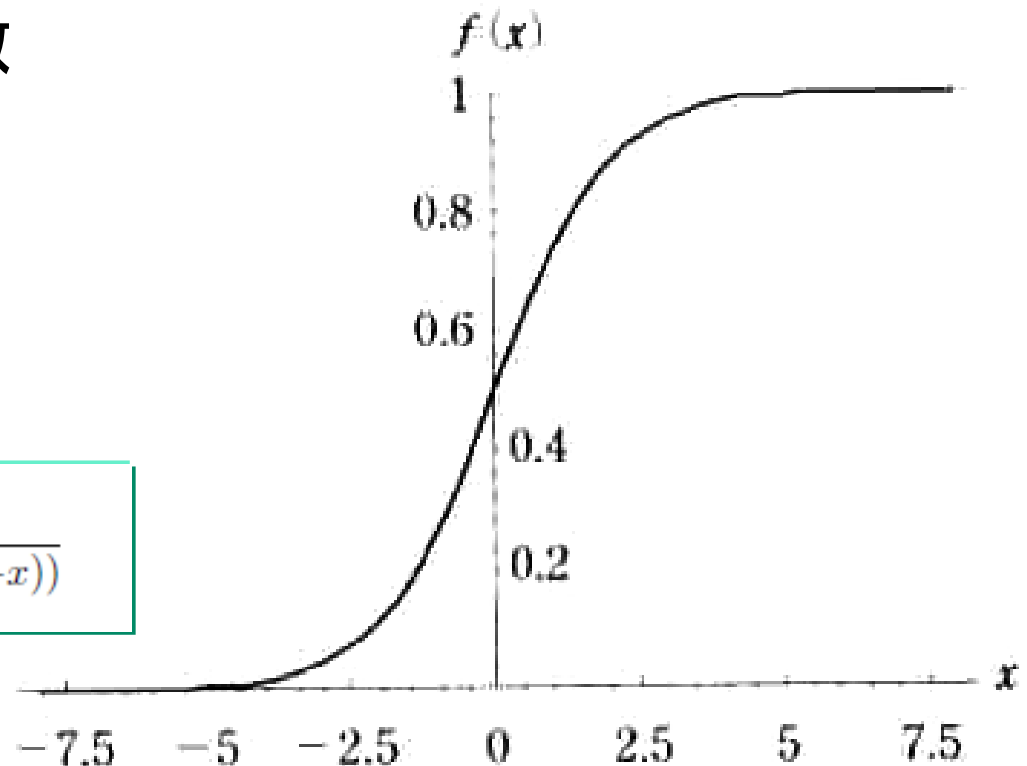
2. シグモイド関数

$$f(x) = \frac{1}{2}(1 + \tanh(x/2)) = \frac{1}{(1 + \exp(-x))}$$

シグモイド(sigmoid) 関数

- $x \rightarrow +\infty$ で値 1.0 に近づき, $x \rightarrow -\infty$ で値 0.0 に近づく.
- S 字型の応答特徴

$$f(x) = \frac{1}{2}(1 + \tanh(x/2)) = \frac{1}{(1 + \exp(-x))}$$





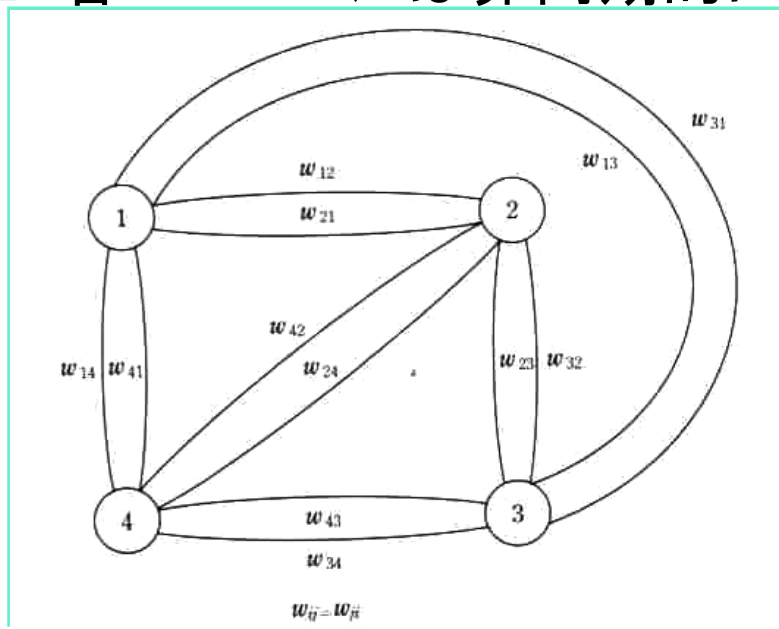
ニューラルネットワークの構造

- ユニット(ニューロン)の間の結合方法
- 1. 相互結合型(ホップフィールド型ネットワーク)
 - 情報の流れは双方向的であり, 各ユニットは出力を伝えた相手のユニットからも情報を受け取る. このようなネットワークでは, 自分の出力が別のユニットを経て, 再び自分への入力として戻ってくる(フィードバック).
- 2. 階層型
 - ユニット間の情報(信号)の流れは一方通行で, 入力層のユニットへの入力が決まれば, 順番に隣接する層のユニットへ情報が伝播していく.

ホップフィールド型ネットワーク

- 1982 年にホップフィールド (J.J.Hopfield)
- 相互結合ネットワーク
 - 1. 相互結合は対称的である. つまり $w_{ij} = w_{ji}$ が成立する
 - 2. 各ニューロンは非同期的に動作する

■ 例:



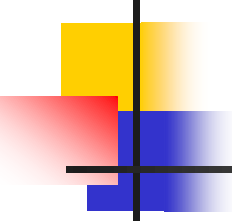


ホップフィールド型ネットワーク

- N 個のユニットから構成されるネットワークにおいて、時刻 t に i 番目のユニットが他の $N - 1$ 個のユニットから受け取る信号の総和 $u_i(t)$ は次のように表される。

$$u_i(t) = \sum_{j \neq i} w_{ij} v_j(t) + \theta_i$$

- $v_j(t)$ はユニット j の出力で、1 または 0 のどちらかの値をとる
- θ_i はユニット i のしきい値
- w_{ij} はユニット間の相互結合係数（結合の重み）であり、ユニット j からユニット i への信号の伝達効率を表す。この値が正の場合は興奮性の結合を、負の場合は抑制性の結合を示す。

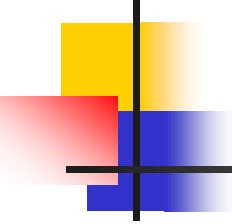


ホップフィールド型ネットワーク

- 次の時刻 $t+1$ でのユニット i の出力値 $v_i(t+1)$ は、入力の総和 $u_i(t)$ が正(または0)ならば1, 負ならば0に更新される.

$$v_i(t+1) = \begin{cases} 1 & u_i(t) > 0 \text{ のとき} \\ 0 & u_i(t) \leq 0 \text{ のとき} \end{cases}$$

- この計算はニューロンのモデル(他のニューロンからの信号の総和がしきい値を越えたときに興奮して発火する)に基づいている



ホップフィールド型ネットワーク

- 次の時刻 $t+1$ でのユニット i の出力値 $v_i(t+1)$ は、入力の総和 $u_i(t)$ が正(または0)ならば1, 負ならば0に更新される.

$$v_i(t+1) = \begin{cases} 1 & u_i(t) > 0 \text{ のとき} \\ 0 & u_i(t) \leq 0 \text{ のとき} \end{cases}$$

- この計算はニューロンのモデル(他のニューロンからの信号の総和がしきい値を越えたときに興奮して発火する)に基づいている



ホップフィールド・ネットワークの動作規則

- **Step1** ネットワークの中からランダムに1つのユニット*i*を選び出す.

- **Step2** ユニット*i*への入力の総和 $u_i(t)$ を

$$u_i(t) = \sum_{j \neq i} w_{ij} v_j(t) + \theta_i$$

にしたがって求める.

- **Step3** $u_i(t)$ の値に従ってユニット*i* の出力 $v_i(t + 1)$ を更新する.

$$v_i(t + 1) = \begin{cases} 1 & u_i(t) > 0 \text{ のとき} \\ 0 & u_i(t) \leq 0 \text{ のとき} \end{cases}$$

- **Step4** *i* 以外のユニットの出力は変化させない.
- **Step5** **Step1** へ戻る.



ネットワークのエネルギー

$$E(t) = -\frac{1}{2} \sum_{i \neq j} w_{ij} v_i(t) v_j(t) - \sum_i \theta_i v_i(t)$$

- ホップフィールドは、上で述べた規則に従ってネットワークを動作させると、ネットワークのエネルギーは単調に減少していくことを示した

ホップフィールド・ネットワークによる連想記憶

- ネットワークに極小点として記憶させたい記憶パターンの数を M 個とし, その m 番目のパターンを $(V_m[0], \dots, V_m[N-1])$ としよう.
- つまり記憶パターンはユニットの出力値の組で表される N 次元ベクトルで表される (M はユニットの総数).
- このとき, ネットワークのユニット間結合係数 w_{ij} としきい値 θ_i を次のように設定する.

$$w_{ij} = \sum_{m=1}^M (2V_m[i] - 1) \sum_j (2V_m[j] - 1) \quad (i \neq j \text{ のとき})$$

$$w_{ii} = 0$$

$$\theta_i = 0$$



ホップフィールド・ネットワークによる連想記憶

- さらに, 記憶パターンが次の制約条件式を満足しているとしよう.

$$2 \sum_j V_m[j] V_n[j] - \sum_j V_n[j] = C_n \delta_{mn}$$

- ただし C_n は正定数であり, $1 \leq n, m \leq M$ である. このとき, ネットワークのパラメータを

$$\begin{aligned} w_{ij} &= \sum_{m=1}^M (2V_m[i] - 1) \sum_j (2V_m[j] - 1) \quad (i \neq j \text{ のとき}) \\ w_{ii} &= 0 \\ \theta_i &= 0 \end{aligned}$$

に従って設定すると, M 個の記憶パターンがエネルギーの極小点になることが示される.

ホップフィールド・ネットワークによる文字パターン認識

- 問題: 式
$$u_i(t) = \sum_{j \neq i} w_{ij} v_j(t) + \theta_i$$
 に従ってネットワークのパラメータを設定した場合には, 記憶パターンだけでなく, 記憶パターンの各成分を反転したパターンも極小点となることを証明せよ.

Hint:

$$\begin{aligned} w_{ij} &= \sum_{m=1}^M (2V_m[i] - 1)(2V_m[j] - 1) \\ &= \sum_{m=1}^M \{2(1 - V_m[i]) - 1\} \{2(1 - V_m[j]) - 1\} \end{aligned}$$



Nクイーン問題をホップフィールド・ネットワークで解く

- ネットワークは $N \times N$ の正方行列として表される
- 2つのニューロン (i, j) と (k, l) を考える.
 - ただし $i, j, k, l \in \{1, 2, \dots, N\}$.
- 2つのニューロンが結合しているのは次の条件のいずれかが成立するとき
 - 1. 同じ列 ($i = k$) のとき.
 - 2. 同じ列 ($j = l$) のとき.
 - 3. 同じ対角線 ($i + j = k + l$ または $i - j = k - l$) のとき.
- 自分自身に結合するニューロンはない.
- すべての結合は対称である.

そしてこれらの結合の重みは負に設定する. 他の場合はニューロン間の結合はなく, 重みはゼロとする.

Nクイーン問題をホップフィールド・ネットワークで解く

- ニューロン(i, j) の出力を V_{ij} と書く.
- 最終状態で $V_{ij} = 1$ となっていれば, クイーンが(i, j) におかれていることを意味する.
- ネットワークエネルギー E

• A, B, C, n_- は正定数である.
• N 個のクイーンの制約条件を満たす正しい配置のものが**極小のエネルギー**になるように定義されている.

なぜか???

$$\begin{aligned} 2E = & A \sum_{i=1}^n \sum_{j=1}^n \left[\left(\sum_{k=1, k \neq j}^n V_{ik} \right) V_{ij} + \left(\sum_{k=1, k \neq i}^n V_{kj} \right) V_{ij} \right] \\ & + B \sum_{i=2}^n \sum_{j=1}^{n-1} \left[\left(\sum_{k=i-j+1, k \neq i}^n V_{k, k-i+j} \right) V_{ij} \right] \\ & + B \sum_{i=1}^n \sum_{j=1}^n \left[\left(\sum_{k=1, k \neq i}^{n+i-j} V_{k, k-i+j} \right) V_{ij} \right] \\ & + B \sum_{i=1}^n \sum_{j=n-i+1}^n \left[\left(\sum_{k=i+j-n, k \neq i}^n V_{k, i+j-k} \right) V_{ij} \right] \\ & + B \sum_{i=1}^{n-1} \sum_{j=1}^{n-i} \left[\left(\sum_{k=1, k \neq i}^{i+j-1} V_{k, i+j-k} \right) V_{ij} \right] \\ & + C \left(\sum_{i=1}^n \sum_{j=1}^n V_{ij} - n_- \right)^2 \end{aligned}$$

Nクイーン問題をホップフィールド・ネットワークで解く

- エネルギーとニューロン x の入力ポテンシャル

$$U_x = -\partial E / \partial V_x \quad (x = 1, \dots, m)$$

- N クイーンの場合

$$\begin{aligned} -U_{ij} = & A \left(\sum_{k=1, k \neq j}^n V_{ik} + \left(\sum_{k=1, k \neq i}^n V_{kj} \right) + R_{ij} \right. \\ & \left. + S_{ij} + C \left(\sum_{i=1}^n \sum_{j=1}^n V_{ij} - n \right) \right) \end{aligned}$$

ただし、

$$R_{ij} = \begin{cases} B \sum_{k=i-j+1, k \neq i}^n V_{k, k-i+j} & i-j > 0 \text{ のとき} \\ B \sum_{k=1, k \neq i}^{n+i-j} V_{k, k-i+j} & i-j \leq 0 \text{ のとき} \end{cases}$$
$$S_{ij} = \begin{cases} B \sum_{k=i+j-n, k \neq i}^n V_{k, i+j-k} & i+j > n \text{ のとき} \\ B \sum_{k=1, k \neq i}^{i+j-1} V_{k, i+j-k} & i+j \leq n \text{ のとき} \end{cases}$$



Nクイーン問題をホップフィールド・ネットワークで解く

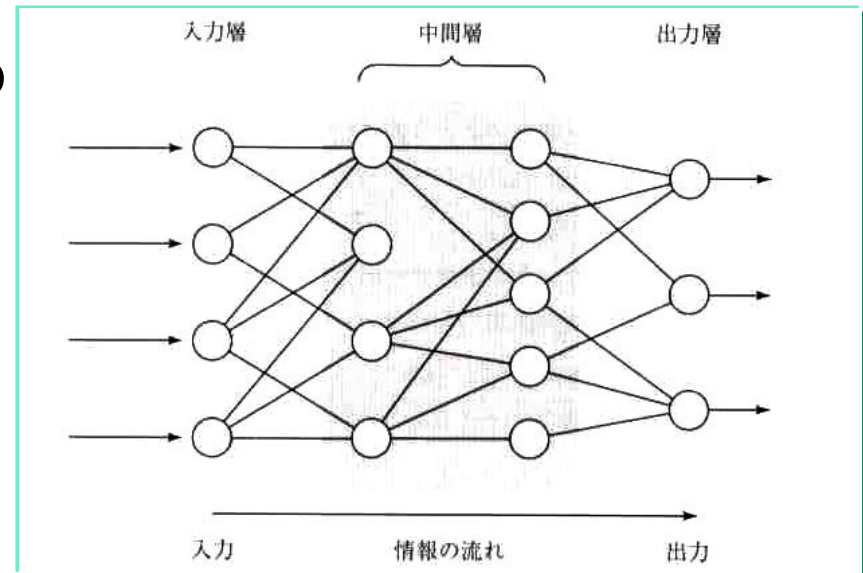
- よって、 N クイーン問題を解くネットワークの動作は次のようになる.
- **Step1** 初期状態の出力値 V_{ij}^0 ($i, j = 1, \dots, N$) を任意に(ランダムに)設定する(0 か1 の値をとる). 初期状態でのエネルギー E_0 を求める.
- **Step2** ニューロン(i, j)をランダムに選ぶ. 他のニューロンの出力をもとに U_{ij} の値を求める. そして次の式によって V_{ij} を変更する.

$$V_{ij} = \begin{cases} 1 & U_{ij} > 0 \text{ のとき} \\ 0 & U_{ij} \leq 0 \text{ のとき} \end{cases}$$

階層型が多層ネットワーク

- 以下の3つの層からなる

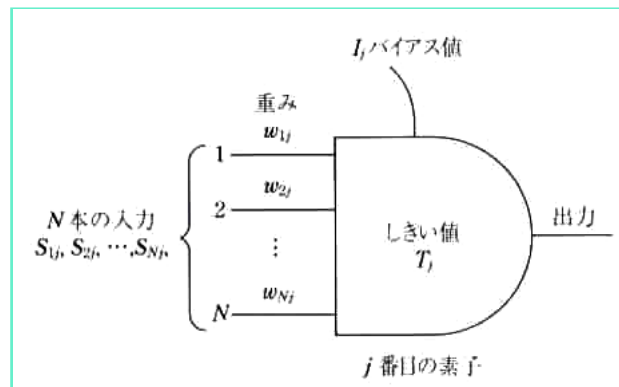
- 入力層
- 中間層(隠れ層)
- 出力層



- 入力からの情報は, 入力層⇒中間層⇒出力層の順に伝達されて出力値が得られる.

階層型が多層ネットワーク

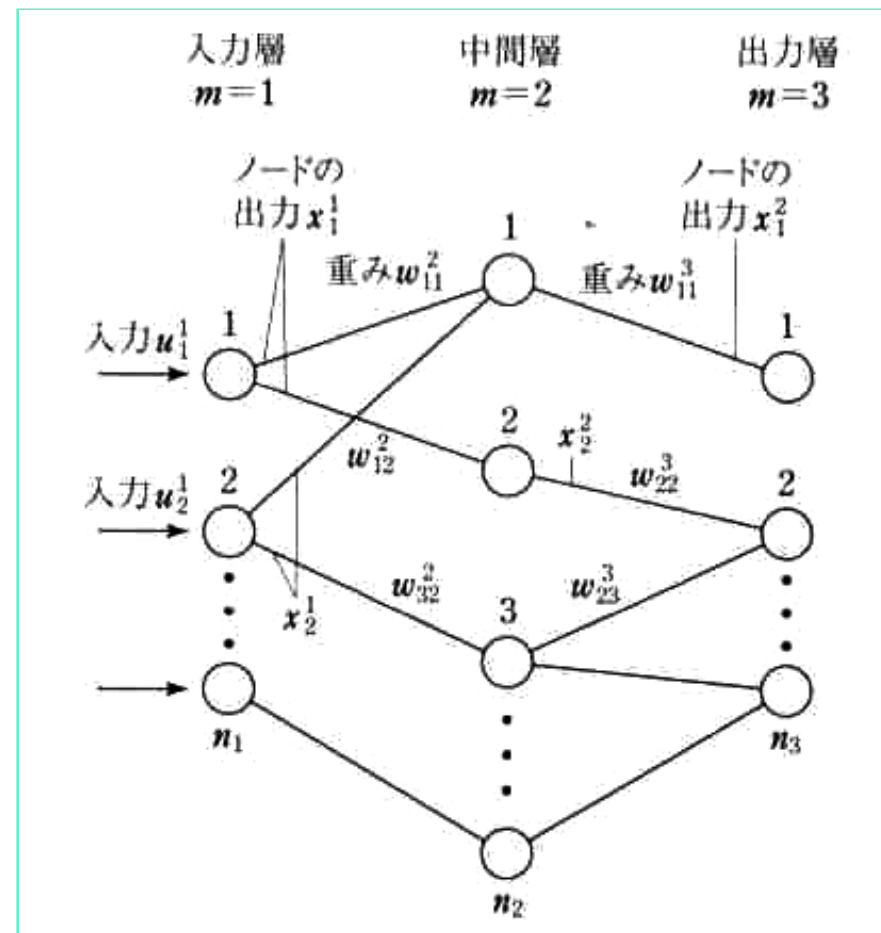
- 各層でのニューラル素子の動作は前に説明した原理に基づいている.



- 1986 年代の半ばRumelhartらは, 出力層での誤差を利用して中間層の細胞の特性を変化させるバックプロゲーション(Back Propagation, BP) 学習アルゴリズムを提案した.

階層型が多層ネットワーク

- 例: 3 層のネットワーク
- 第 m 層の細胞数を n_m 個, 細胞への入力を u^m_{ji} , 出力を x^m_{ji} , 第 m 層への結合行列を $W^m = \{w^m_{ij}\}$ と表す.
- ここで w^m_{ij} は第 $m - 1$ 層の素子 j から第 m 層の素子 i への結合係数(重み)である.

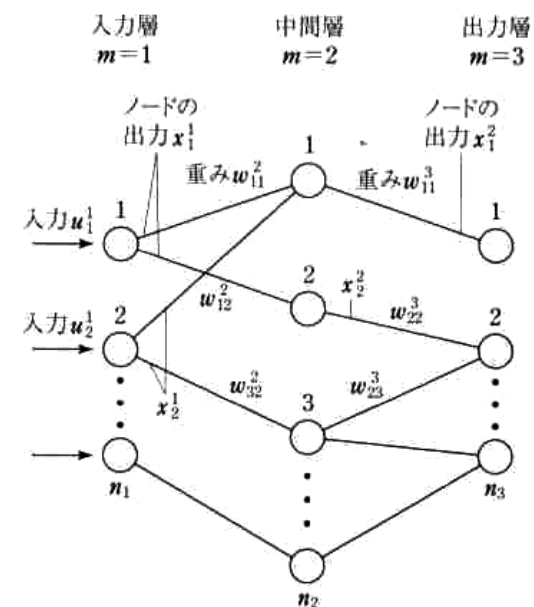


階層型が多層ネットワーク

- このとき各素子の動作は,

$$u_i^m = \sum_{j=1}^{n_{m-1}} w_{ij}^m x_j^{m-1}$$
$$x_i^m = f(u_i^m) \quad (m = 2, 3, i = 1, \dots, n_m)$$

- f は出力を計算する関数であり、ステップ関数やシグモイド関数がかかる

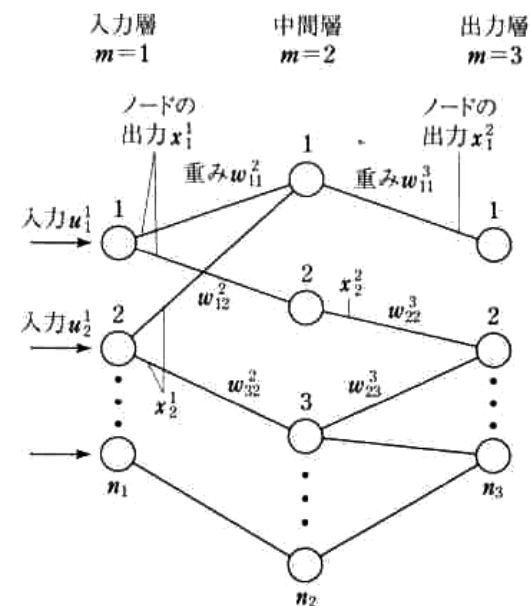


階層型が多層ネットワーク

- このとき各素子の動作は,

$$u_i^m = \sum_{j=1}^{n_{m-1}} w_{ij}^m x_j^{m-1}$$
$$x_i^m = f(u_i^m) \quad (m = 2, 3, i = 1, \dots, n_m)$$

- f は出力を計算する関数であり、ステップ関数やシグモイド関数が用いられる





階層型が多層ネットワーク

- 学習の目標

- 入力($x^1_{1'} \cdots, x^1_{n1}$)が与えられたときの最終層の出力($x^3_{1'} \cdots, x^3_{n3}$)を教師信号($d_{1'} \cdots, d_{n3}$)に近づけること

- そこで出力の2乗誤差

$$E = \frac{1}{2} \sum_{i=1}^{n_3} (x^3_i - d_i)^2$$

を最急降下法により最小化する.

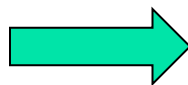
階層型の多層ネットワーク

- 最急降下法による最小化

$$E = \frac{1}{2} \sum_{i=1}^{n_3} (x_i^3 - d_i)^2$$

$$\frac{\partial E}{\partial x_i^3} = x_i^3 - d_i$$

$$\frac{\partial E}{\partial x_i^2} = \sum_{k=1}^{n_3} \frac{\partial E}{\partial x_k^3} \frac{\partial x_k^3}{\partial u_k^2} \frac{\partial u_k^2}{\partial x_i^2}$$



$$\frac{\partial E}{\partial x_i^2} = \sum_{k=1}^{n_3} \frac{\partial E}{\partial x_k^3} f'(u_k^2) w_{ik}^2$$

- よって、出力層と中間層の重みに関する二乗誤差の最急降下学習は ($m=2,3$)

$$\begin{aligned} \Delta w_{ij}^m &= -\mu \cdot \frac{\partial E}{\partial w_{ij}^m} \\ &= -\mu \cdot \partial E / \partial x_i^m \cdot \partial x_i^m / \partial u_i^m \cdot \partial u_i^m / \partial w_{ij}^m \\ &= -\mu \cdot \partial E / \partial x_i^m \cdot f'(u_i^m) \cdot x_j^{m-1} \end{aligned}$$

μ は学習のスピード
を決める係数

階層型の多層ネットワーク

- 以下の学習原理は4層以上のネットワークにも同じように適用できる

$$\begin{aligned}\Delta w_{ij}^m &= -\mu \cdot \frac{\partial E}{\partial w_{ij}^m} \\ &= -\mu \cdot \partial E / \partial x_i^m \cdot \partial x_i^m / \partial v_i^m \cdot \partial v_i^m / \partial w_{ij}^m \\ &= -\mu \cdot \partial E / \partial x_i^m \cdot f'(v_i^m) \cdot x_j^{m-1}\end{aligned}$$

- シグモイド関数を使うとその微分は

$f'(x) = f(x)(1 - f(x))$ と簡単に計算できる

- このとき学習則は

$$\Delta w_{ij}^m = -\mu \cdot \frac{\partial E}{\partial x_i^m} \cdot v_i^m \cdot (1 - v_i^m) \cdot x_j^{m-1}$$

出力の誤差が結合を通じて逆向きに伝播していくので、**バックプロパゲーション**と呼ばれる。

バックプロパゲーションを用いた学習

- 例1：1ビットの加算を求めるニューラル・ネットワーク
- 3層ニューラルネットワークを用いて加算器の学習を実験する.
- 各層の素子数
 - 入力層が2 ($D0, D1$ に対応する)
 - 中間層が4
 - 出力層が2 (x, y に対応)
- 学習係数 $\mu = 0.25$

$D0$	$D1$	x	y
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

バックプロパゲーションを用いた学習

Step1 ネットワークの重みをランダムに初期化する (0から1の範囲とする).

Step2 $Epoch := 0$ とする.

Step3 $Trial := 0$ とする.

Step4 ランダムに訓練データを取りだす. その入力をもとに出力値を計算する.

Step5 式 (333) に基づいて重みを調整する.

Step6 $Trial := Trial + 1$ とする.

$$\Delta w_{ij}^m = -\mu \cdot \frac{\partial E}{\partial x_i^m} \cdot u_i^m \cdot (1 - u_i^m) \cdot x_j^{m-1}$$

Step7 $Trial < MaxTrial$ なら Step4 に戻る.

Step8 $Epoch := Epoch + 1$ とする.

Step9 $Epoch < MaxEpoch$ なら Step3 に戻る.



バックプロパゲーションを用いた 学習

- 例2: 時系列データの予測
- 時系列予測問題(Time Series Prediction)
- 実用的価値も高く古くから研究されている.
 - 例えば株価予測、降水量の予測など
- 一般に予測すべき時系列データの振舞いが非線形で複雑な場合が多く、時系列予測問題では近似関数の形式とパラメータをいかに導出するかがポイントになる.

バックプロパゲーションを用いた学習

- 次の式に従う時系列データ $x(1), x(2), x(3), \dots$ を予想しよう

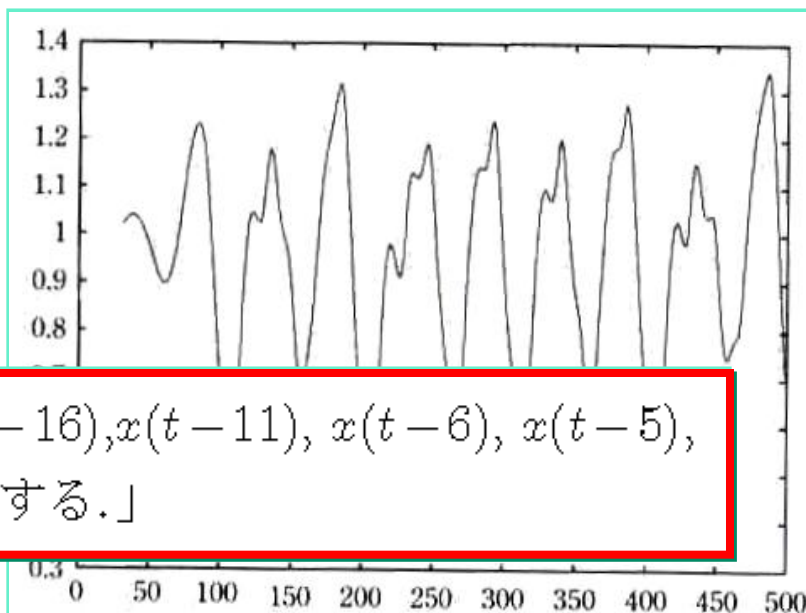
$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} - bx(t)$$

$$a = 0.2, b = 0.1, \tau = 1.7$$

- Mackey–Glass 微分方程式と呼ばれる

- $x(t)$ は約3.5 フレージング・タイムスケールの非線形振動を示す。予測問題

「 $x(t)$ を過去のデータ $x(t-31), x(t-21), x(t-16), x(t-11), x(t-6), x(t-5), x(t-4), x(t-3), x(t-2), x(t-1)$ から予測する。」



バックプロパゲーションを用いた学習

Step1 ネットワークの重みをランダムに初期化する（0から1の範囲とする）.

Step2 $Epoch := 0$ とする.

Step3 $Trial := 0$ とする.

Step4 ランダムに訓練データを取りだす. その入力をもとに出力値を計算する.

Step5 式 (333) に基づいて重みを調整する.

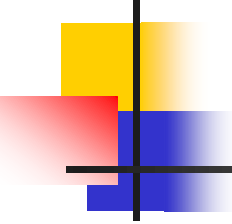
Step6 $Trial := Trial + 1$ とする. $\Delta w_{ij}^m = -\mu \cdot \frac{\partial E}{\partial x_i^m} \cdot u_i^m \cdot (1 - u_i^m) \cdot x_j^{m-1}$

Step7 $Trial < MaxTrail$ なら Step4 に戻る.

Step8 テストデータに対して得られたネットワークで予測を行う.

Step9 $Epoch := Epoch + 1$ とする.

Step10 $Epoch < MaxEpoch$ なら Step3 に戻る.



ボルツマンマシン

- 1984 年 ヒントン(G.E.Hinton) らによって提案されたニューラルネットワーク・モデル
- ホップフィールド・ネットワークの動作規則を確率的に拡張したもの
- 入力から得られた $u_i(t)$ から出力値 $v_i(t+1)$ を決めるときに確率による判定を導入する.



ボルツマンマシン

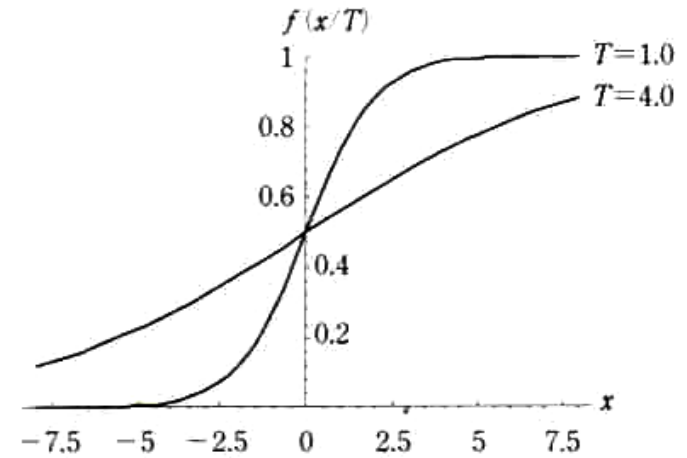
- ユニット i の次の時刻での出力値 $v_i(t+1)$ は, $u_i(t)$ の値から計算される次式の確率 P に従って1 に設定される($1-P$ の確率で0 に設定される).

$$P[v_i(t+1) = 1] = f(u_i(t)/T)$$
$$f(x) = \frac{1}{2}(1 + \tanh(x/2)) = \frac{1}{(1 + \exp(-x))}$$

- $f(x)$ はシグモイド関数であり, $x \rightarrow +\infty$ で値1.0に近づく, $x \rightarrow -\infty$ で値0 に近づく.

ボルツマンマシン

$$P[v_i(t+1) = 1] = f(u_i(t)/T)$$
$$f(x) = \frac{1}{2}(1 + \tanh(x/2)) = \frac{1}{(1 + \exp(-x))}$$



- パラメータ T は「ネットワークの温度」
- T の値によって関数 $f(x/T)$ の形が変化する.
 - T が大きい (温度が高い) と $f(x/T)$ は x の値の違いに鈍感になる.
 - $T \rightarrow +\infty$ の極限では x の値によらず $f(x/T) = 0.5$ になる
 - T が小さい (温度が低い) と $f(x/T)$ は x の値に敏感になる
 - $T \rightarrow 0$ の極限では $f(x/T)$ は階段関数に収束する.

このことから、温度 T が 0 の極限においてボルツマン・マシンは **ホップフィールド・ネットワーク** に一致する。

$$P[v_i(t+1) = 1] = f(u_i(t)/T)$$

$$f(x) = \frac{1}{2}(1 + \tanh(x/2)) = \frac{1}{(1 + \exp(-x))}$$

ボルツマンマシン

- $v_i(t+1) = 1$ の状態に遷移した場合のネットワークのエネルギーの変化
 - 状態遷移の前後で出力が変化するのはユニット*i*だけなので、遷移前後のエネルギーの変化量は次の式で与えられる。

$$E(t+1) - E(t) = -\{1 - v_i(t)\}u_i(t)$$

証明せよ!!

- この式から $u_i(t)$ が正の場合には、状態遷移によってエネルギーが減少する(または変化しない)ことが分かる。



ボルツマン・マシンの動作規則

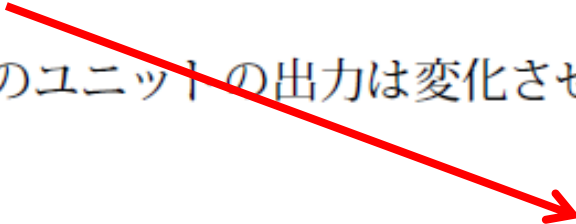
Step1 ネットワークの中からランダムに1つのユニット i を選び出す.

Step2 選ばれたユニット i への出力の総和 $u_i(t)$ を求める.

$$u_i(t) = \sum_{j \neq i} w_{ij} v_j(t) + \theta_i \quad (342)$$

Step3 式 (340) に基づく確率 P でユニット i の出力 $v_i(t+1)$ を1に設定する.

Step4 i 以外のユニットの出力は変化させない. Step1 へ戻る.


$$\begin{aligned} P[v_i(t+1) = 1] &= f(u_i(t)/T) \\ f(x) &= \frac{1}{2}(1 + \tanh(x/2)) = \frac{1}{(1 + \exp(-x))} \end{aligned}$$



ボルツマン分布関数

- ホップフィールド・ネットワークでは, 状態はエネルギー関数の極小点を与える1つの状態に収束した.
- それに対してボルツマン・マシンでは状態遷移に確率を導入したために, ある確率でネットワークの各状態が出現するような平衡状態に収束する.
- そして平衡状態に収束してからも同じ動作規則で状態遷移を繰り返し, 各状態の出現した回数の時間平均を計測すると, 実現確率分布を求めることができる.
- この確率分布は次のようなボルツマン分布に一致することが示されている.

$$Q(E_n) = (1/Z)\exp(-E_n/T)$$

$$Z = \sum_n \exp(-E_n/T)$$



ボルツマン分布関数

- 実現確率 $Q(E_n)$ は温度 T が高いとエネルギーの小さな違いに鈍感になり, 逆に温度 T が低いと敏感になることがわかる.
- そして温度 $T \rightarrow 0$ の極限では最小エネルギーの状態の実現確率が1.0 になり, 他の状態の実現確率はすべて0 になる.
- このことをうまく利用すると, ネットワークの状態をエネルギー関数の最小値に収束させることができる.

$$Q(E_n) = (1/Z)\exp(-E_n/T)$$
$$Z = \sum_n \exp(-E_n/T)$$



ボルツマン分布関数

- つまり, はじめは高い温度から出発してネットワークを動作させ平衡状態に到達させる. そのあとで平衡状態を崩さないように徐々に温度を下げていき, 最終的に温度0 の極限に到達させる方法が提案された.

$$Q(E_n) = (1/Z)\exp(-E_n/T)$$

$$Z = \sum_n \exp(-E_n/T)$$

この方法は金属材料などを加熱して, 徐々に冷却して内部の欠陥を取り除く「焼き鈍し」に似ていることから「シミュレーションによる**焼き鈍し(Simulated Annealing)**」と呼ばれている.



焼き鈍し法

課題: この式が成り立つことを分かりやすく説明せよ。単に論文の証明の書きうつしは駄目。高校程度の数学で分かるように平易に説明すること。

- 重要なのが温度を下げるスケジュール
- あまり速い速度で温度を下げると極小点に取り残される.
- ギーマンら(S.Geman & D.Geman)は, 温度 $T(t)$ を次の条件に従って下げていけばネットワークの状態を必ず最小値に収束させられることを証明した.

$$T(t) \geq \frac{c}{\log(1+t)}$$

- t はネットワークの動作規則の適用回数 (**Step1** の繰り返し数) を表しており, c は定数である.

巡回セールスマン問題

- N 都市の巡回セールスマン問題を解くのに用いる $N \times N$ 個のニューロンを各々 $V_{X,i}$ と書く ($X = A, B, \dots$; $i = 1, 2, \dots$).
- ただし X は都市名, i は訪れる順番である.
- このときネットワークのエネルギー関数は

$$\begin{aligned} E = & A/2 \sum_X \sum_i \sum_{j \neq i} V_{X,i} V_{X,j} \\ & + B/2 \sum_i \sum_X \sum_{X \neq Y} V_{X,i} V_{Y,i} \\ & + C/2 \left(\sum_X \sum_i V_{X,i} - n \right)^2 \\ & + D/2 \sum_X \sum_{Y \neq X} \sum_i d_{XY} V_{X,i} (V_{Y,i+1} + V_{Y,i-1}) \end{aligned}$$

簡単のため,

$$V_{Y,n+j} = V_{Y,j}$$

$$V_{Y,j-n} = V_{Y,j}$$

などのように N を法として添え字を計算する.

ネットワークのエネルギー関数

- この関数の最小値が問題の解に対応している.
- ただしA,B,C,Dは正の定数である

$$\begin{aligned} E = & A/2 \sum_X \sum_i \sum_{j \neq i} V_{Xi} V_{Xj} \\ & + B/2 \sum_i \sum_X \sum_{X \neq Y} V_{Xi} V_{Yi} \\ & + C/2 \left(\sum_X \sum_i V_{Xi} - n \right)^2 \\ & + D/2 \sum_X \sum_{Y \neq X} \sum_i d_{XY} V_{Xi} (V_{Y,i+1} + V_{Y,i-1}) \end{aligned}$$

■ 各項の意味

- 第1項: 各行が1を2つ以上含まないとき, ゼロとなる.
- 第2項: 各列が1を2つ以上含まないとき, ゼロとなる.
- 第3項: 全体で1の数が N 個のとき, ゼロとなる.
- 第4項: 経路の距離を合計した値に相当する.

巡回セールスマン問題

- ニューロン(X, i)と(Y, j)の結合の重み $T_{Xi,Yj}$ は次のようになる.

- 各項の意味

- 第1項: 各行内の抑制結合.
- 第2項: 各列内の抑制結合.
- 第3項: 大域的抑制.
- 第4項: 距離データ項.

$$\begin{aligned} T_{Xi,Yj} = & -A\delta_{X,Y}(1 - \delta_{i,j}) \\ & -B\delta_{i,j}(1 - \delta_{X,Y}) \\ & -C \\ & -Dd_{X,Y}(\delta_{j,i+1} + \delta_{j,i-1}) \end{aligned}$$

$$\delta_{ij} = \begin{cases} 0 & i \neq j \text{ のとき} \\ 1 & i = j \text{ のとき} \end{cases}$$

- さらに、各ニューロンのしきい値としては、

$$I_{Xi} = +C \times n$$

- 常に興奮するほうにバイアスをかけている