情報意味論(5) パーセプトロン

理工学部管理工学科
櫻井彰人

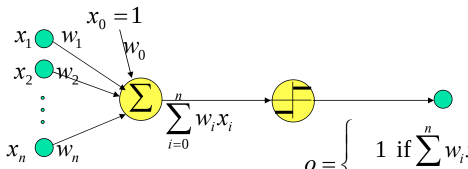
神経回路網 ニューラルネットワーク

- 動物の神経・神経回路にヒントを得る
- 人工の神経素子(neuron)とそのネットワーク
  - 多くの場合はソフトウェアで実現
- 適応する、学習する
- 連続値が扱える

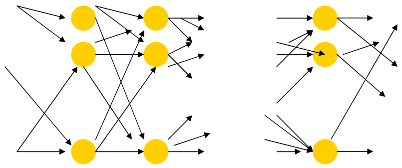
今回

- 概要
- パーセプトロン

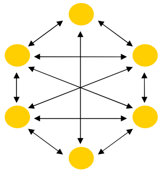
McCullogh-Pitts モデル(1943)



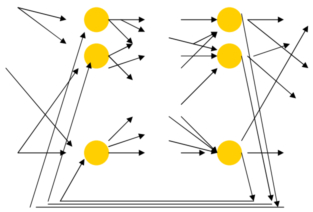
$$o = \begin{cases} 1 & \text{if } \sum_{i=0}^{n} w_i x_i > 0 \\ -1 & \text{otherwize} \end{cases}$$



階層型（フィードフォワード）

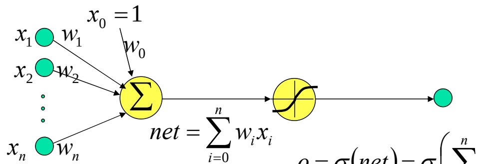


相互結合



再帰型（リカレント）

シグモイド素子

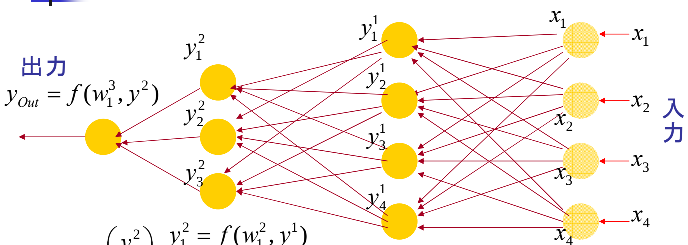


$$net = \sum_{i=0}^{n} w_i x_i$$

$$o = \sigma(net) = \sigma\left(\sum_{i=0}^{n} w_i x_i\right)$$

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

基本的な計算

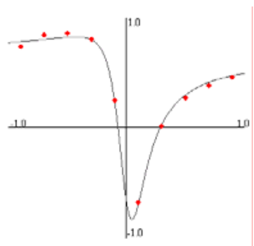


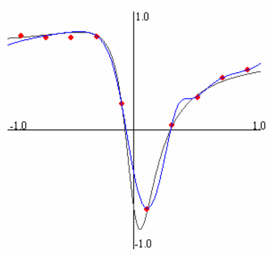
$$y^2 = \begin{pmatrix} y_3^2 \\ y_3^2 \\ y_3^2 \end{pmatrix}, \quad \begin{matrix} y_1^2 = f(w_1^2, y^1) \\ y_2^2 = f(w_2^2, y^1) \\ y_3^2 = f(w_3^2, y^1) \end{matrix}$$

$$y^1 = \begin{pmatrix} y_1^1 \\ y_2^1 \\ y_3^1 \\ y_4^1 \end{pmatrix}, \quad \begin{matrix} y_1^1 = f(w_1^1, x_1) \\ y_2^1 = f(w_2^1, x_2) \\ y_3^1 = f(w_3^1, x_3) \\ y_4^1 = f(w_4^1, x_4) \end{matrix}$$

ニューラルネットの原理

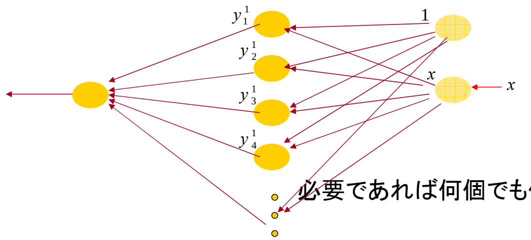
- 一側面：関数近似・回帰である
  - 例：1入力・1出力とする

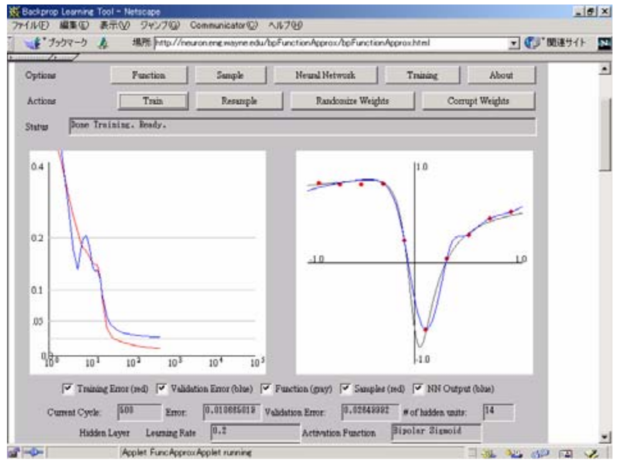




近似定理

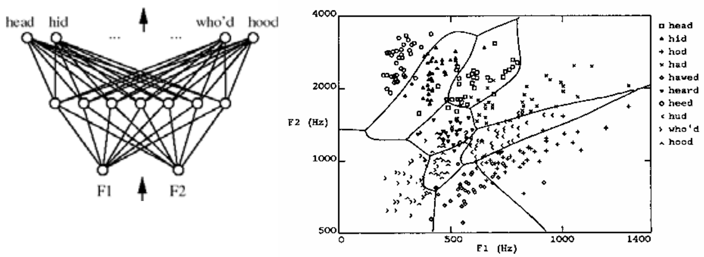
- ニューラルネットワークの中間素子数を必要なだけ用意できるなら、任意の滑らかな関数を任意の精度で近似することができる





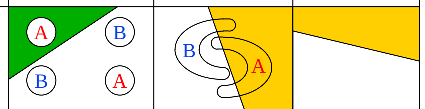
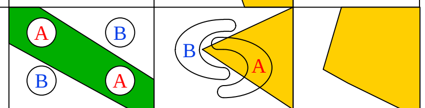
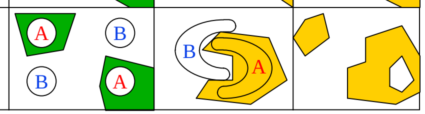
http://neuron.eng.wayne.edu/bpFunctionApprox/bpFunctionApprox.html

音声認識



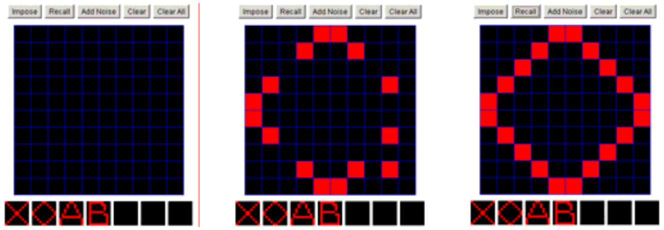
http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-3/www/ml.html

領域と境界面

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

連想

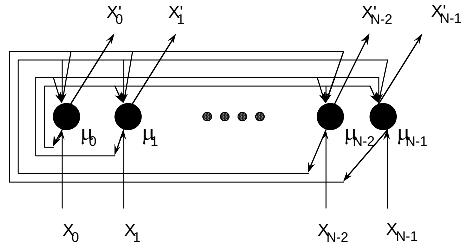
- (A,a), (B,b), (C,c),… というデータを記憶し、(A,?) と聞かれたら ?=a と答える
- 変形した・ノイズの乗った図形から元の図を復元する。



http://www.physics.syr.edu/courses/modules/MM/sim/hopfield.html

Hopfieldネットワーク

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



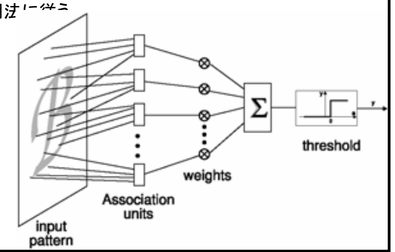
Kohonenマップ

- 説明省略

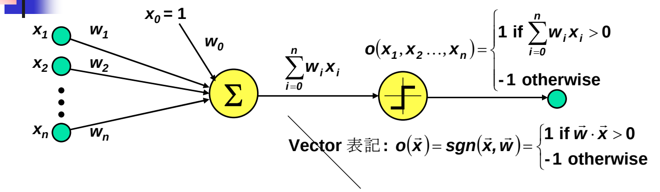
http://rfhs8012.fh-regensburg.de/~saj39122/jfroehl/diplom/e-sample.html

多義語: パーセプトロン

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

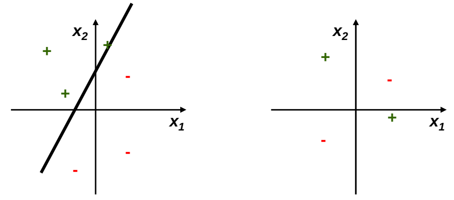


パーセプトロン Perceptron



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

パーセプトロンの決定境界



Example A　　Example B

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

パーセプトロン学習アルゴリズム

- 学習規則 ≡ 訓練規則 training Rule
  - 教師付き学習に特有の話ではない
  - 文脈: モデルの更新
- Hebbの学習則 Hebbian Learning Rule (Hebb, 1949)
  - アイデア: もし2個の素子が両方とも active ("firing")であれば, 結合荷重は増加する
  - $w_{ij} = w_{ij} + r\, o_i\, o_j$, 但し r は学習係数 learning rate で、定数である
  - 神経生理学的に, ほぼ, 指示されている
- パーセプトロン学習アルゴリズム Perceptron Learning Rule (Rosenblatt, 1959)
  - アイデア: 各入力ベクトルに対して出力値が与えられているなら, 荷重を漸進的に更新することにより, 当該出力値が出力できるようになる
  - 2値出力 (Bool値, Boolean-valued) を仮定; 単一パーセプトロン素子
  - $w_i \leftarrow w_i + \Delta w_i$
    $\Delta w_i = r(t - o)x_i$

    但し $t = c(x)$ は目標出力値, o はパーセプトロンの現在の出力値, r は学習係数, 正定数であれば何でも良い. 1でよいので、実は、パーセプトロン学習アルゴリズムでは, r は不要
  - D が線型分離可能 linearly separable であれば, 収束する. r が十分小さいことを条件とする説明もあるがそれは誤り

パーセプトロン学習アルゴリズム

- 単純な勾配降下 Gradient Descent アルゴリズムである
  - このアイデアは, 適当な表現を用いれば, 概念学習にも記号学習にも適用可能
- アルゴリズム *Train-Perceptron* ($D \equiv \{<x,\ t(x) \equiv c(x)>\}$)
  - 荷重 w_i をランダム値に初期化する　　　　// パーセプトロン時は0に初期化してもよい
  - WHILE 正しい出力をしない事例がある DO

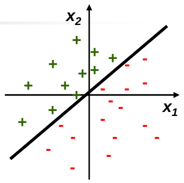
    FOR それぞれの事例 $x \in D$

    　　現在の出力 $o(x)$ を計算

    　　FOR i = 1 to n

    　　　　$w_i \leftarrow w_i + r(t - o)x_i$　　　　**// perceptron learning rule. r is any positive #**
- パーセプトロン学習可能性
  - 復習: $h \in H$ のときのみ学習可能 - i.e., 線型分離可能 linearly separable (LS) functions
  - Minsky and Papert (1969) *Perceptrons* : 元祖パーセプトロンの表現･学習の限界を示した
    - 注: 素子一個では parity (n-変数 XOR: $x_1 \oplus x_2 \oplus \ldots \oplus x_n$) 関数が表現できない, というのは既知
    - e.g., 画像の symmetry, connectedness は(元祖パーセプトロンで)表現できない
    - "*Perceptrons*" のせいで ANN 研究が10年近く遅れたといわれもするが、どこまで真実か。

線型分離



Linearly Separable (LS) Data Set

- 定義
  - $f(x) = 1$ if $w_1x_1 + w_2x_2 + \ldots + w_nx_n \geq \theta$, 0 otherwise
  - θ: 閾値
- 線型分離可能性
  - 注: D が線型分離可能だからといって, 真の概念 $c(x)$ が線型分離可能とは限らない
  - 選言 disjunction: $c(x) = x_1' \vee x_2' \vee \ldots \vee x_m'$ ✓
  - m of n: $c(x)$ = at least 3 of $(x_1', x_2', \ldots, x_m')$ ✓
  - 排他的 exclusive OR (XOR): $c(x) = x_1 \oplus x_2$ ✗
  - 一般の DNF: $c(x) = T_1 \vee T_2 \vee \ldots \vee T_m$; $T_i = l_1 \wedge l_1 \wedge \ldots \wedge l_k$ ✗
- 表現の変換
  - 線型分離可能でない問題を線型分離可能な問題に変換できるか?
  - それは意味のあることなのか? 現実的なのか?
  - 現実問題の重要な部分を占めるのか?

パーセプトロン学習の収束

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