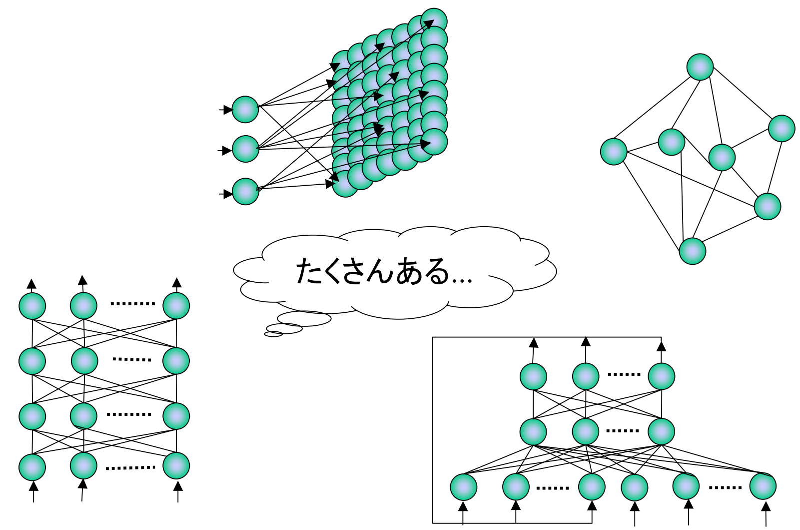
ニューラルネットの話

真部　雄介

ニューラルネットの例



ニューラルネットを見る3つの観点

- **構造（つながり方）**
  - 階層型 or 相互結合型
- **信号の流れ方（計算順序）**
  - フィードバックあり（リカレント） or　フィードバックなし（フィードフォワード）？
- **学習法**
  - 教師あり(supervised)学習 or 教師なし(unsupervised)学習 ?

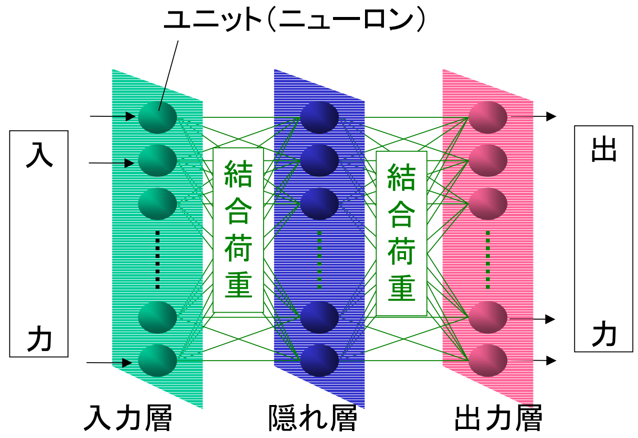
様々なモデルがあってキリがないので...

最も代表的な,
「教師あり学習」「階層型」「フォードフォワード」
のニューラルネットワークを説明します.

ニューラルネットワーク

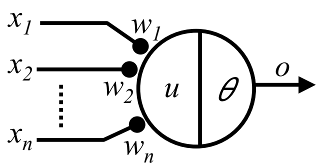
「教師あり学習」「階層型」「フォードフォワード」の

- ニューラルネットワークの例

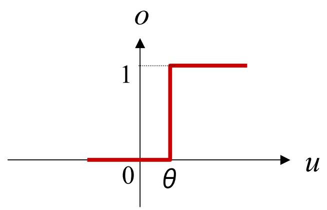


ニューロンの動作（1/2）

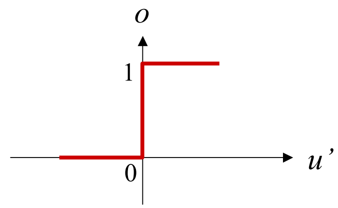
- ニューロンモデル（ McCullochとPitts [1943]）



$$u = \sum_{i=1}^{n} w_i x_i \quad o = \begin{cases} 1, & (u > \theta) \\ 0, & (u \le \theta) \end{cases}$$

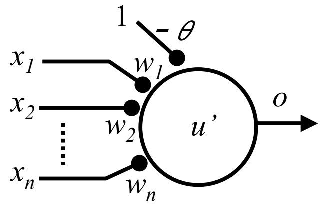


$$u' = \sum_{i=1}^{n} w_i x_i - \theta \quad o = \begin{cases} 1, & (u' > 0) \\ 0, & (u' \le 0) \end{cases}$$



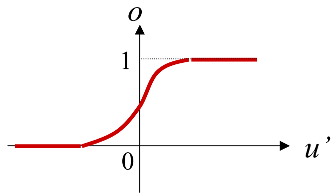
ニューロンの動作（2/2）

- ニューロンモデル改



$$\begin{aligned} u' &= \sum_{i=1}^{n} w_i x_i - \theta \\ &= \sum_{i=1}^{n} w_i x_i + (-\theta) * 1 \\ &= \sum_{i=1}^{n} w_i x_i + \theta \end{aligned}$$

$$o = f(u') = \frac{1}{1 + e^{-u'}}$$

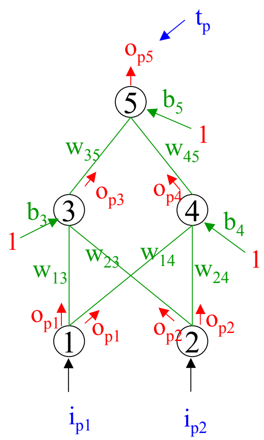


これをいっぱいつなげて構造化すると
ニューラル　ネットワーク

ニューラルネットの計算過程

p: 入出力データパターン数
i_{px}: pパターン目のユニットxへの入力値
t_p: pパターン目の教師データ
w_{xy}: ユニットxy間の結合荷重
o_{px}: pパターン目のユニットxの出力値
b_x: ユニットxのバイアス荷重（閾値）

$o_{p1} = i_{p1}$, $o_{p2} = i_{p2}$
$o_{p3} = f(u3)$, $u3 = o_{p1}*w_{13}+o_{p2}*w_{23}+b_3$
$o_{p4} = f(u4)$, $u4 = o_{p1}*w_{14}+o_{p2}*w_{24}+b_4$
$o_{p5} = f(u5)$, $u5 = o_{p3}*w_{35}+o_{p4}*w_{45}+b_5$
$f(x) = 1/1+\exp(-x)$



$w_{xy} \in W$, $b_x \in B$を調整し、$(i_{p1}, i_{p2}) \in I_p$が入力された時のo_{p5}をt_pに近付けたい！

何ができるのか

「教師あり学習」「階層型」「フォードフォワード」の
ニューラルネットワークは…

任意の複雑な入力と出力の関係性（関数）を
任意の精度で実現することのできる学習モデル

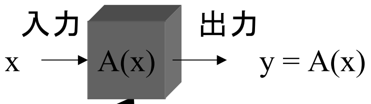
おおざっぱにいうと

関数近似マシン

と思って使っています

関数を作る（1）

- 例）ブラックボックスAをつくる

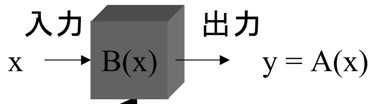


こんな箱Aを作りたい！
x=1のとき，y=2
x=2のとき，y=4
x=3のとき，y=6
x=4のとき，y=8
x=5のとき，y=10

xとyの対応づけを
実現するA(x)とは？

関数を作る（2）

- 例）ブラックボックスBをつくる

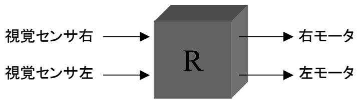


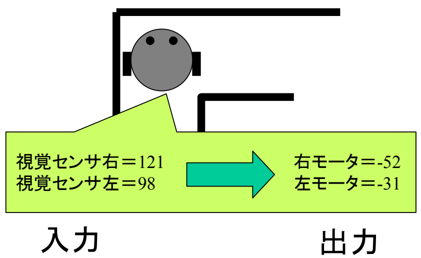
こんな箱Bを作りたい！
x=1のとき，y=0.1
x=2のとき，y=2.9
x=3のとき，y=5
x=4のとき，y=2.5
x=5のとき，y=1.3

xとyの対応づけを
実現するB(x)とは？

世の中は複雑な関数だらけ

- ロボットの行動
  - 環境→センサ→内部処理→モーター→行動

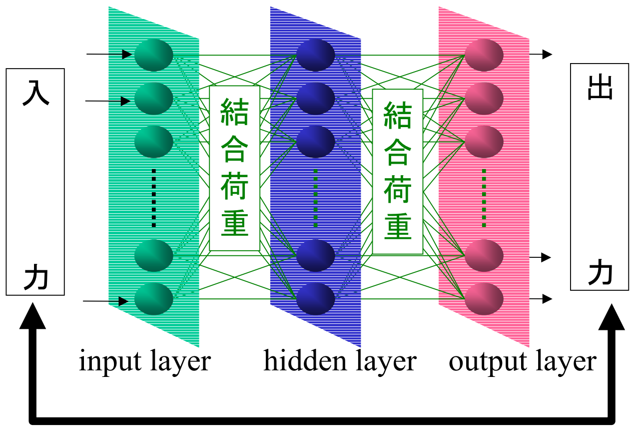




複雑な入力と出力の関数でも対応づけを実現したい

ニューラルネットワークなら可能

ニューラルネットの学習＝関数近似



入力と出力の対応付けを実現すること= 学習
関数を作る（関数近似）

入出力関係（関数）の実現の仕方

- 自分が実現したい入力と出力の関係を表わすデータを用意する
  - 例1）ブラックボックスB
    - （入力，出力）＝ {(1, 0.1), (2, 2.9), (3, 5), (4, 2.5), (5, 1.3)}
      学習データパターン（これは5パターン）

入力データ　教師データ

  - 例2）入力／出力は，もちろん多変数でもOK.
    - （入力,出力）＝ {((1, 2), 1), ((2, 3), 5), ((2, 1), 3), ((3, 4), 5)}
      2入力1出力

- そのデータを使って関数を作る（学習する）
  - 結合荷重を調整して，入出力の対応づけを実現する
  - 誤差逆伝播学習法（バックプロパゲーション学習法）が有名

最急降下法（1/3）

例）$y = f(x) = \frac{1}{2}(x-1)^2$ という関数がある。このyができるだけ小さくなるようなxを求めよ。

数値解法的に求める→最急降下法

最急降下法の手順

①xの初期値を適当に決める。
②その時の傾きを求める。
③傾きに応じてxの値をyが小さくなるように少し修正する。
④修正した値を次のxとして②に戻る。

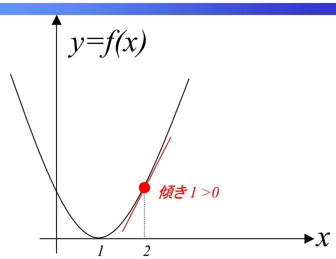
最急降下法（2/3）

① x=2とする

② x=2のときの傾きは、

$y = \frac{1}{2}(x-1)^2 \quad f'(x) = \frac{dy}{dx} = (x-1)$ より

$\boxed{f'(2) = 1}$



③

$x^{new} = x + \Delta x$

$\Delta x = -\eta \frac{dy}{dx}$

とすればyが小さくなる方へxが修正される！ η :学習係数

η =0.3とすると
$x^{new} = 2 - 0.3*1 = 1.7$

x=2のとき	y=0.5
x=1.7のとき	y=0.245

確かにyが小さくなった！

最急降下法（3/3）

④ x=1.7として再度計算。順次計算していくと…

$x=1.7$, 傾き0.7, $x^{new}=1.7-0.3*0.7=1.49$

$x=1.49$, 傾き0.49, $x^{new}=1.49-0.3*0.49=1.343$

$x=1.343$, 傾き0.343, $x^{new}=1.343-0.3*0.343=1.2401$

$x=1.2401$, 傾き0.2401, $x^{new}=1.2401-0.3*0.2401=1.16807$

…..

xが最小値0に近づいていく！

誤差逆伝播学習

ニューラルネットの目的：o_{p5}をt_pに近づけたい！

$$E=\sum_{p=1}^{N}E_p \qquad E_p=\frac{1}{2}\left(o_{p5}-t_p\right)^2$$

E_p：p番目の学習データパターンに対する出力層の出力値と教師データの誤差
E：トータルの誤差

各学習パターンの誤差を小さくする＝ E_pを小さくする

→ $E_p=\frac{1}{2}\left(o_{p5}-t_p\right)^2$ を小さくするようにWを修正する！

誤差逆伝播学習

$$x^{new}=x+\Delta x \qquad \Delta x=-\eta\frac{dy}{dx}$$

$$w_{ij}^{new}=w_{ij}^{old}+\Delta w_{ij} \qquad \Delta w_{ij}=-\eta\frac{\partial E_p}{\partial w_{ij}}$$

例1）w_{45}の修正量は

$$\Delta w_{45}=-\eta\frac{\partial E_p}{\partial w_{45}}=-\eta\frac{\partial E_p}{\partial o_5}\frac{\partial o_{p5}}{\partial u_5}\frac{\partial u_5}{\partial w_{45}}=\underbrace{\boxed{-\eta\left(o_{p5}-t_p\right)f'(u_5)}}_{\text{学習信号}}o_{p4}$$

例2）w_{24}の修正量は

$$\Delta w_{24}=-\eta\frac{\partial E_p}{\partial w_{24}}=-\eta\frac{\partial E_p}{\partial o_{p5}}\frac{\partial o_{p5}}{\partial u_5}\frac{\partial u_5}{\partial o_{p4}}\frac{\partial o_{p4}}{\partial u_4}\frac{\partial u_4}{\partial w_{24}}$$

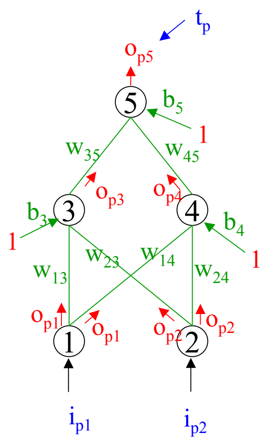
$$=\boxed{-\eta\left(o_{p5}-t_p\right)f'(u_5)}w_{45}f'(u_4)o_{p2}$$

逆伝播！

ニューラルネットの計算過程

p：入出力データパターン数
i_{px}: pパターン目のユニットxへの入力値
t_p：pパターン目の教師データ
w_{xy}：ユニットxy間の結合荷重
o_{px}: pパターン目のユニットxの出力値
b_x：ユニットxのバイアス荷重（閾値）

$o_{p1}=i_{p1}$, $o_{p2}=i_{p2}$
$o_{p3}=f(u3)$, $u3=o_{p1}*w_{13}+o_{p2}*w_{23}+b_3$
$o_{p4}=f(u4)$, $u4=o_{p1}*w_{14}+o_{p2}*w_{24}+b_4$
$o_{p5}=f(u5)$, $u5=o_{p3}*w_{35}+o_{p4}*w_{45}+b_5$
$f(x)=1/1+\exp(-x)$



$w_{xy}\in W$, $b_x\in B$を調整し、$(i_{p1}, i_{p2})\in I_p$が入力された時のo_{p5}をt_pに近付けたい！

まとめると

$$\begin{aligned}w_{ij}^{new}&=w_{ij}^{old}+\Delta w_{ij}\\&=w_{ij}^{old}-\eta\frac{\partial E_p}{\partial w_{ij}}\\&=w_{ij}^{old}-\eta\delta_i o_j\end{aligned}$$

i: 対象としている層のユニット番号
j: 1つ下層のユニット番号

ただし，
$$\delta_i=\begin{cases}(o_i-t_i)f'(u_i) & i\text{が出力ユニット番号のとき}\\ \delta_k w_{ki}f'(u_i) & i\text{が隠れユニット番号のとき}\end{cases}$$
（k: 1つ上層のユニット番号）

$$f(x)=\frac{1}{1+e^{-x}}$$
$$f'(x)=f(x)(1-f(x))$$