

知識工学

第十回 ニューラルネットワーク

コメント



<http://papapac.com/post.php?room=知識情報工学citns>



コメントを投稿したい人

教えてもらった部屋名を入れてね

知識情報工学citns

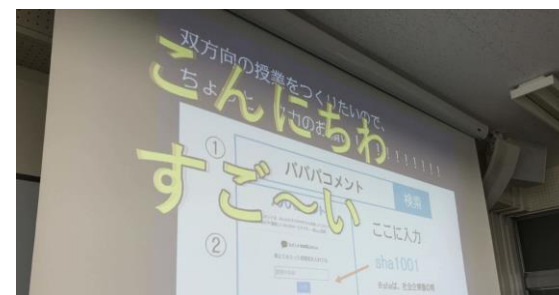
入室



知識情報工学citnsの部屋

コメント送信

コメントは部外者にも見られる可能性があります。個人情報などは送信しないでください。



面白かった
(偶数)



もっと知りたい
(偶数)



面白かった
(奇数)



もっと知りたい
(奇数)

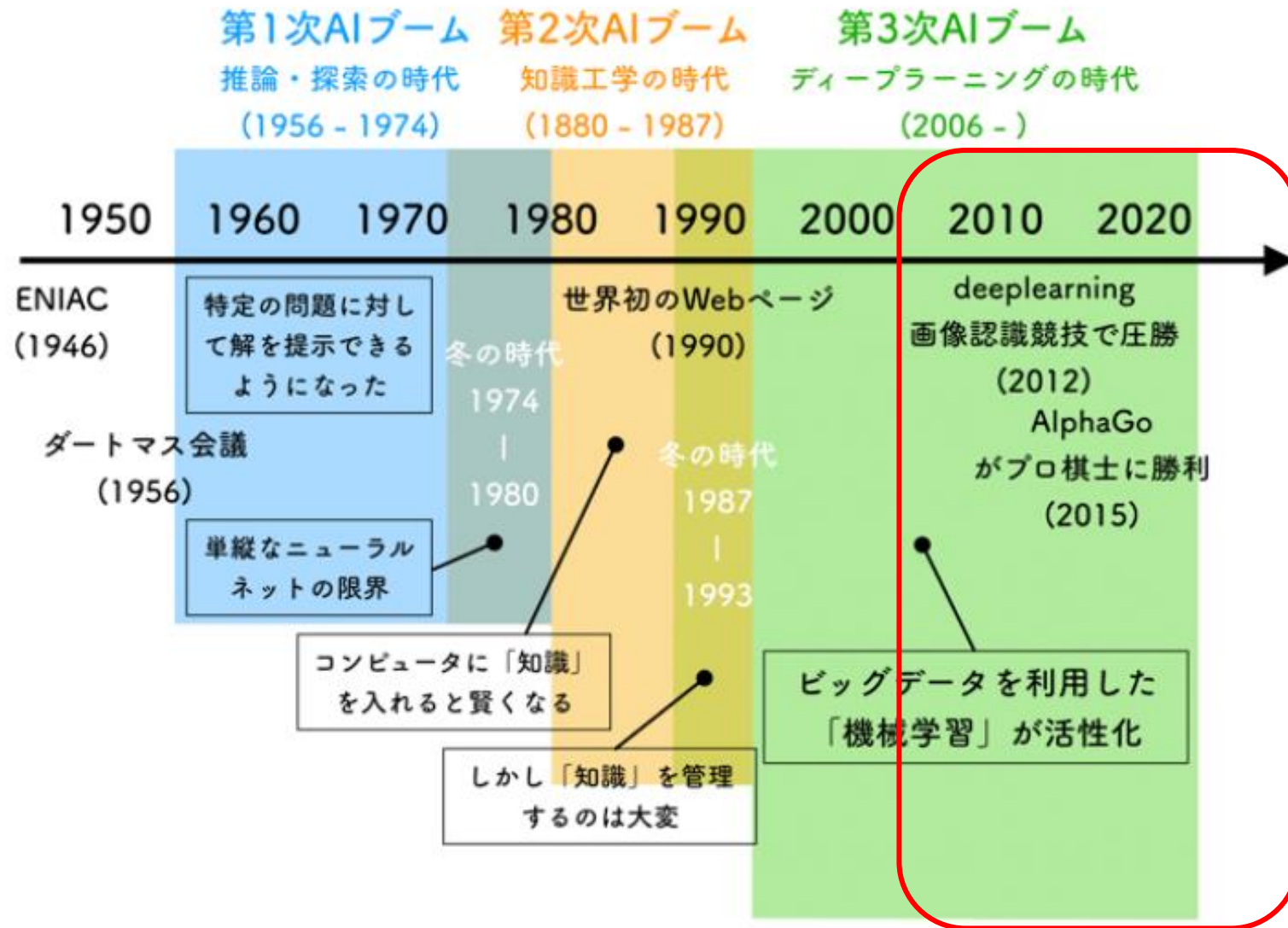
[illegible]

コメント返答

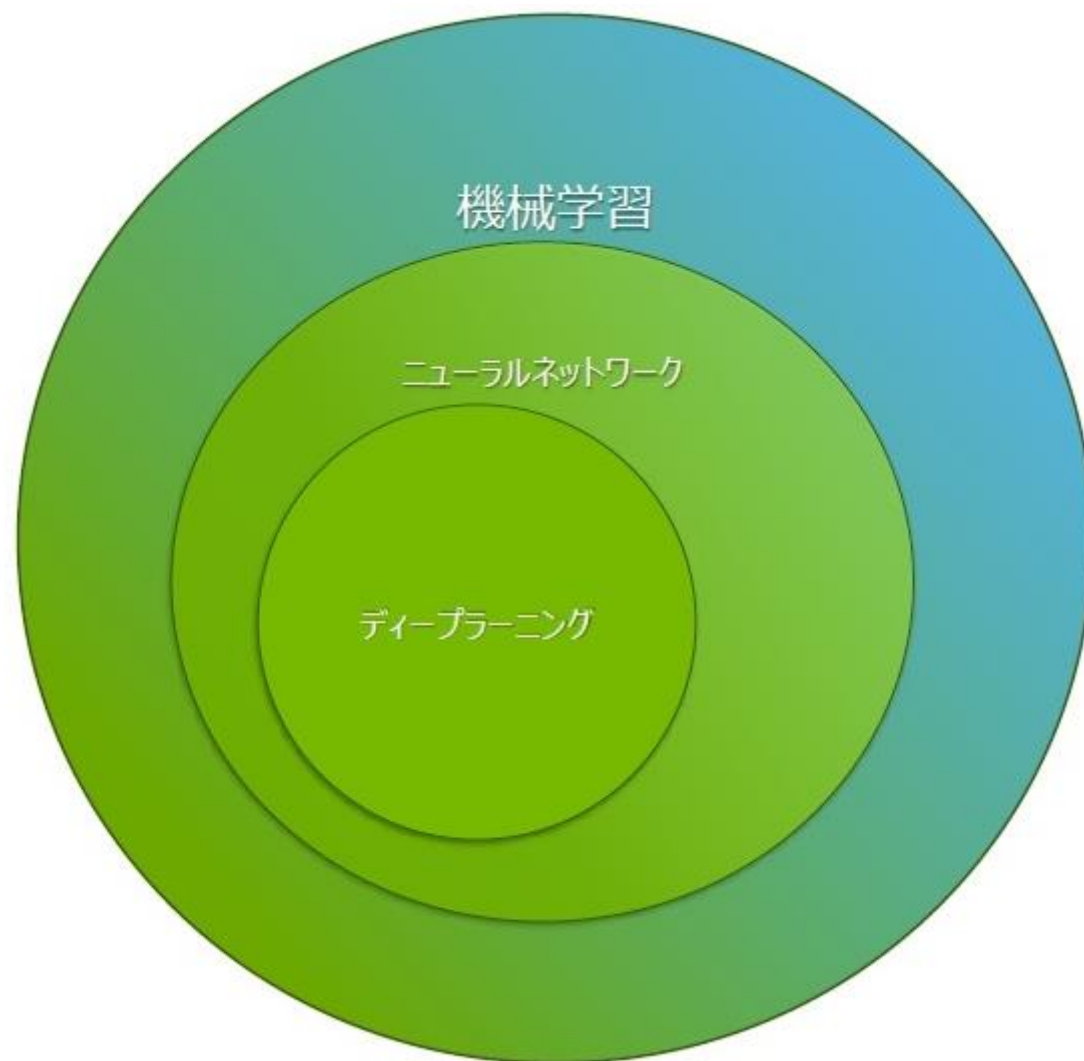
- 強化学習の適用が難しい範囲
 - 正解の評価が難しいもの：会話，気分・感情理解，
 - シミュレーションが難しいもの：教育ロボットが良い授業をするための説明方法
 - 数値の予測や分析：トライ＆エラーの話ではないため，他の手法が適している

今日の内容

- 1970年代の発案当初から、近年まで続く発展の内容
- 画像処理や数値予測の技術
- 最近の応用例

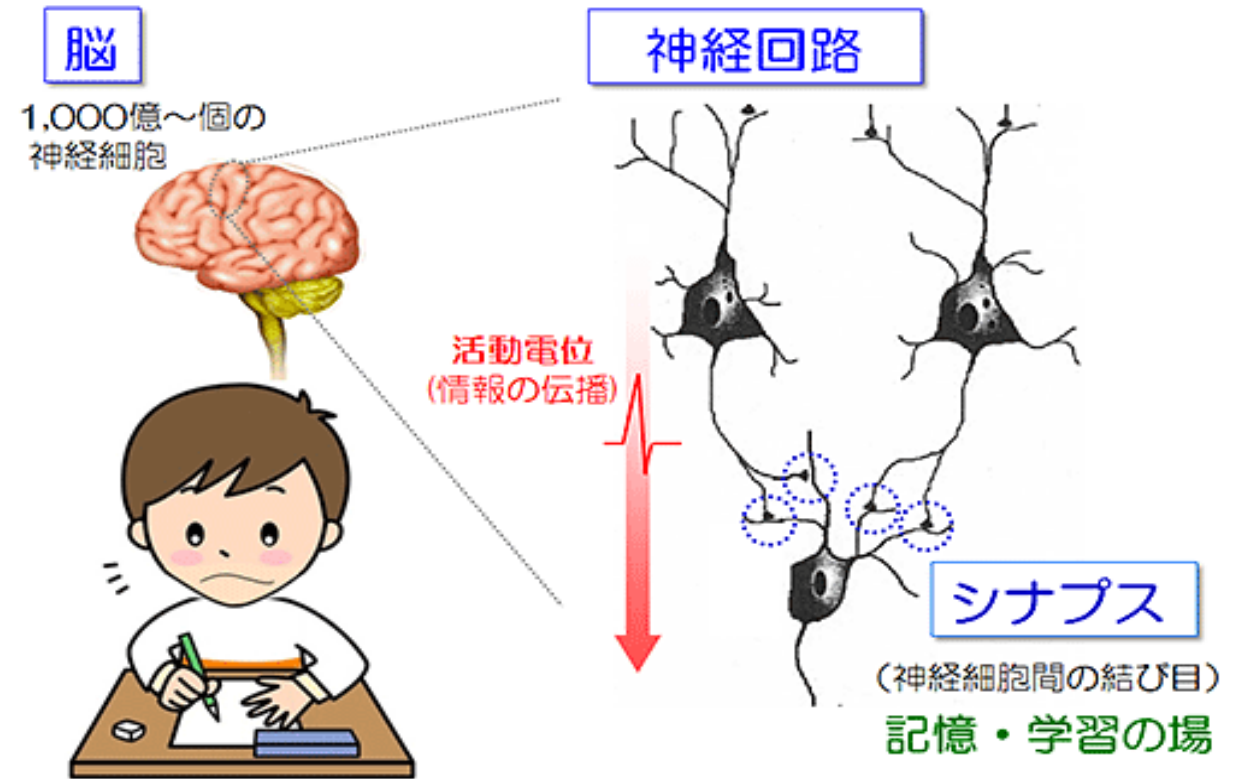


機械学習との関係



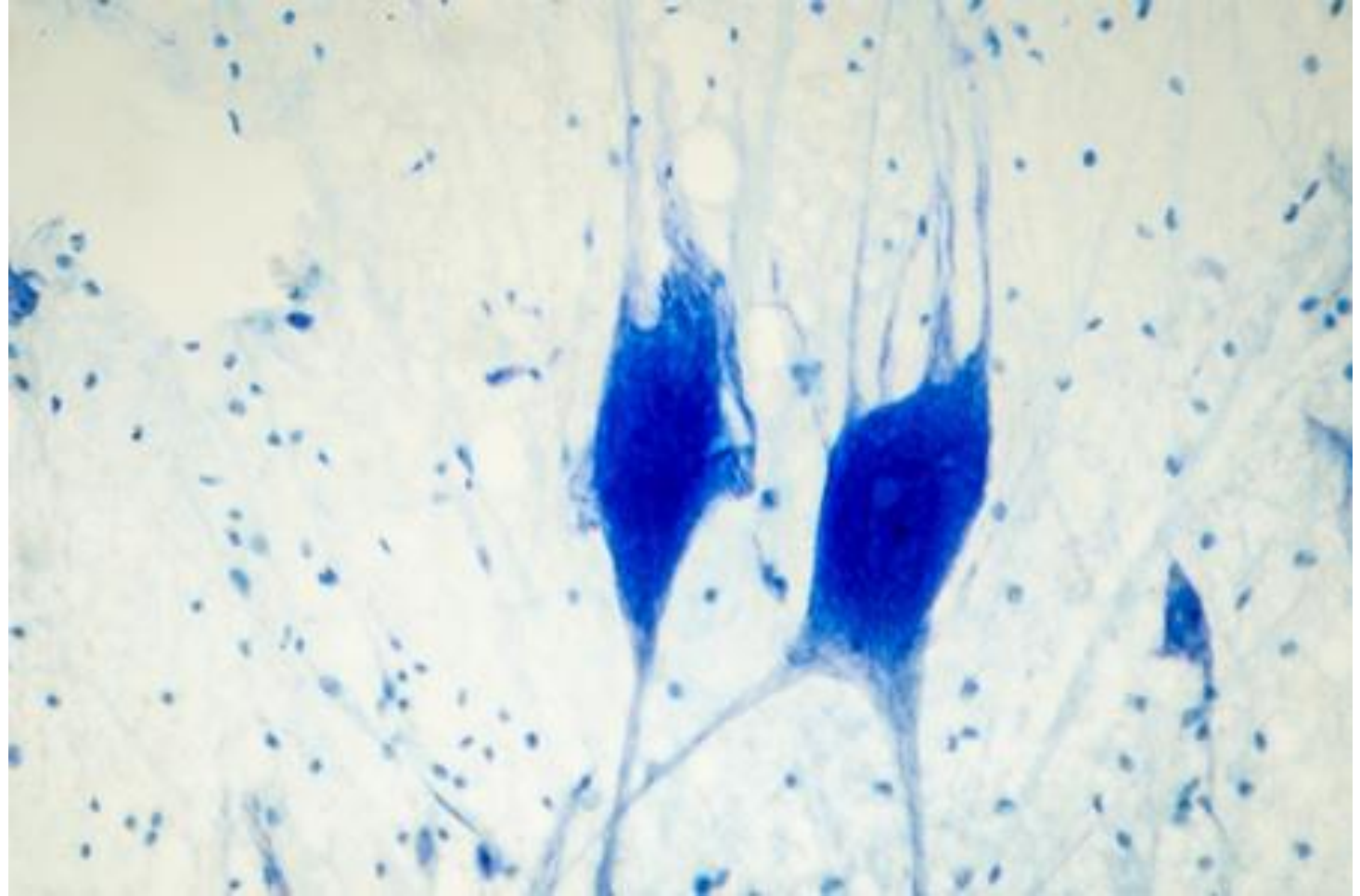
そもそもニューロンとは

- 信号が来たら何らかの計算を行って、閾値以上なら電位を上げる（発火）
- 発火したら、それを信号として他のニューロンに伝える



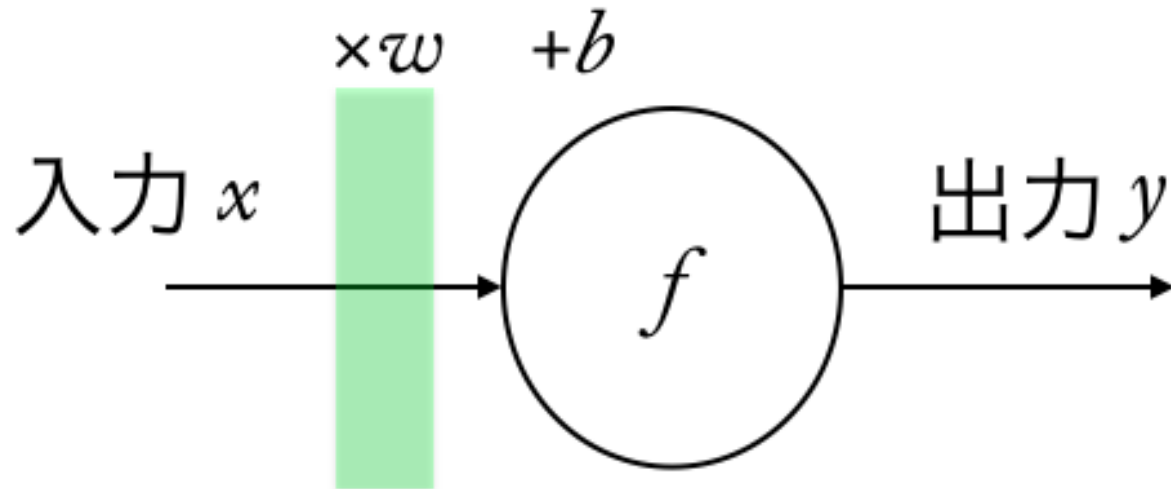
実物

- 見やすいように染色されている



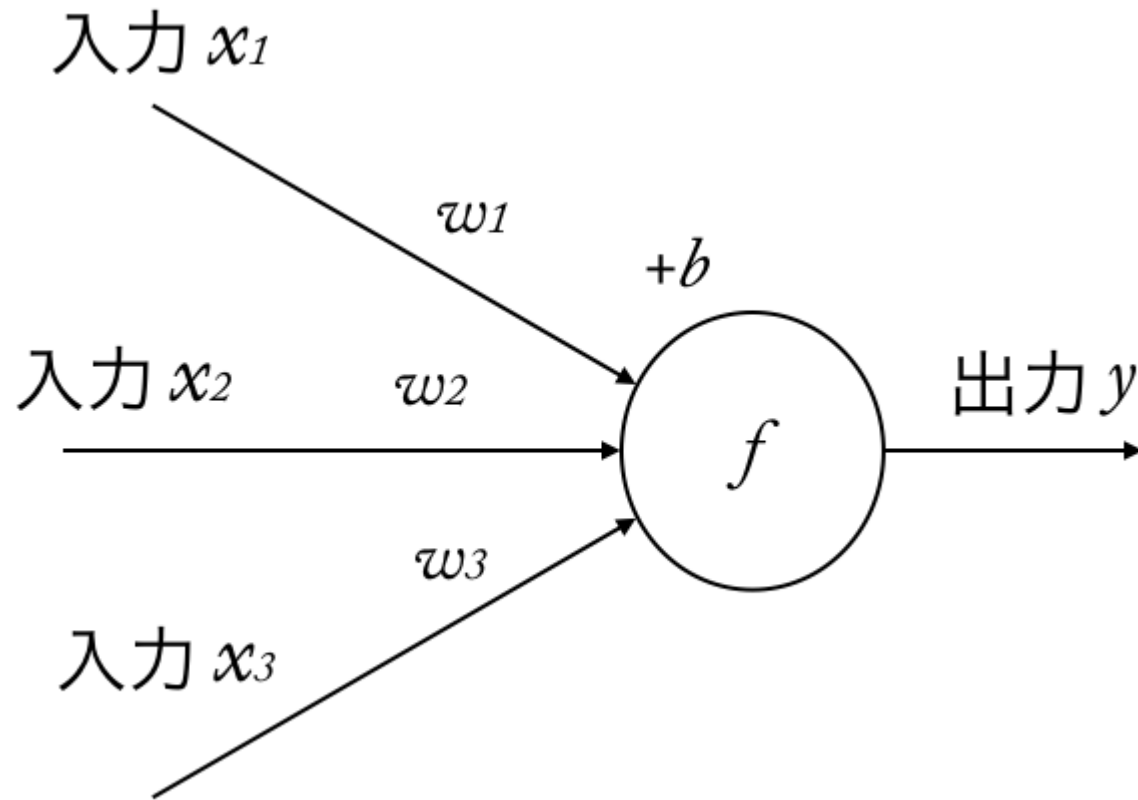
人工的なニューロン（ユニット）

- 繋がっているユニットの出力 x を受け，当該ユニットの出力 y が決まる
- そのつながりがどの程度重要であるかを，重み（ w ）として学習する
- バイアスとして，定数項の b を足す



$$\text{出力 } y = f(\text{入力 } wx + b) \text{ と表せる}$$

複数の入力



入力が複数でも，基本は同じ
 $y = f(w_1x_1 + w_2x_2 + w_3x_3 + b)$

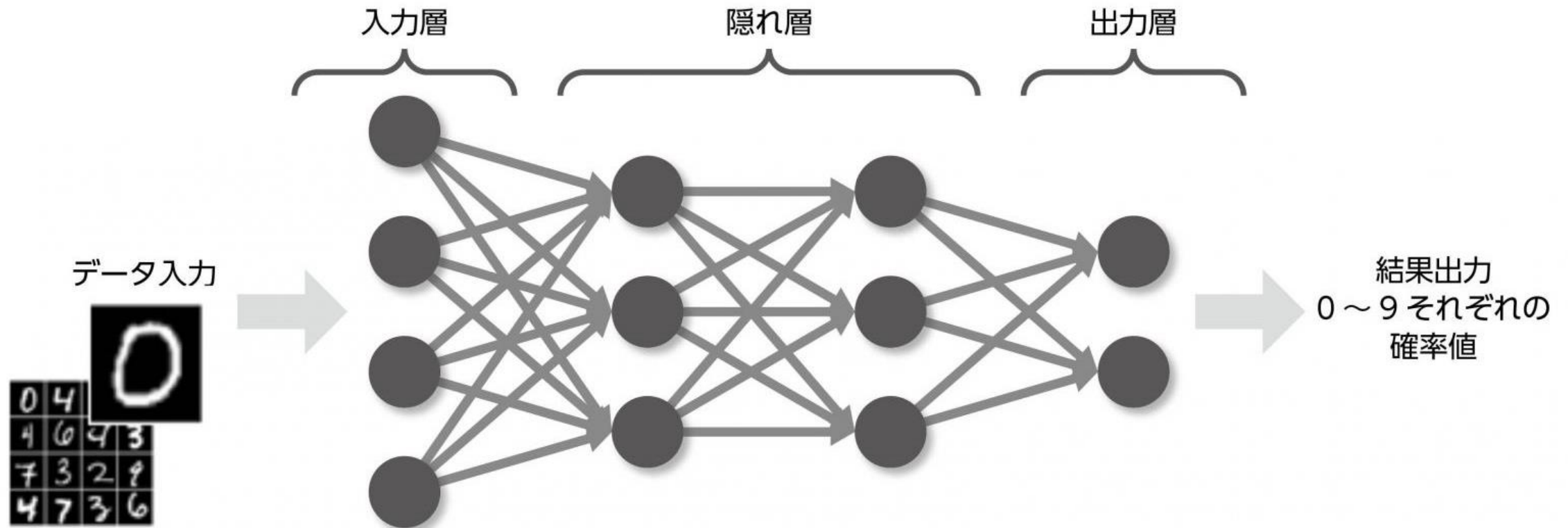
$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

という内積の形で表すと，

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

ニューラルネットワークと深層学習

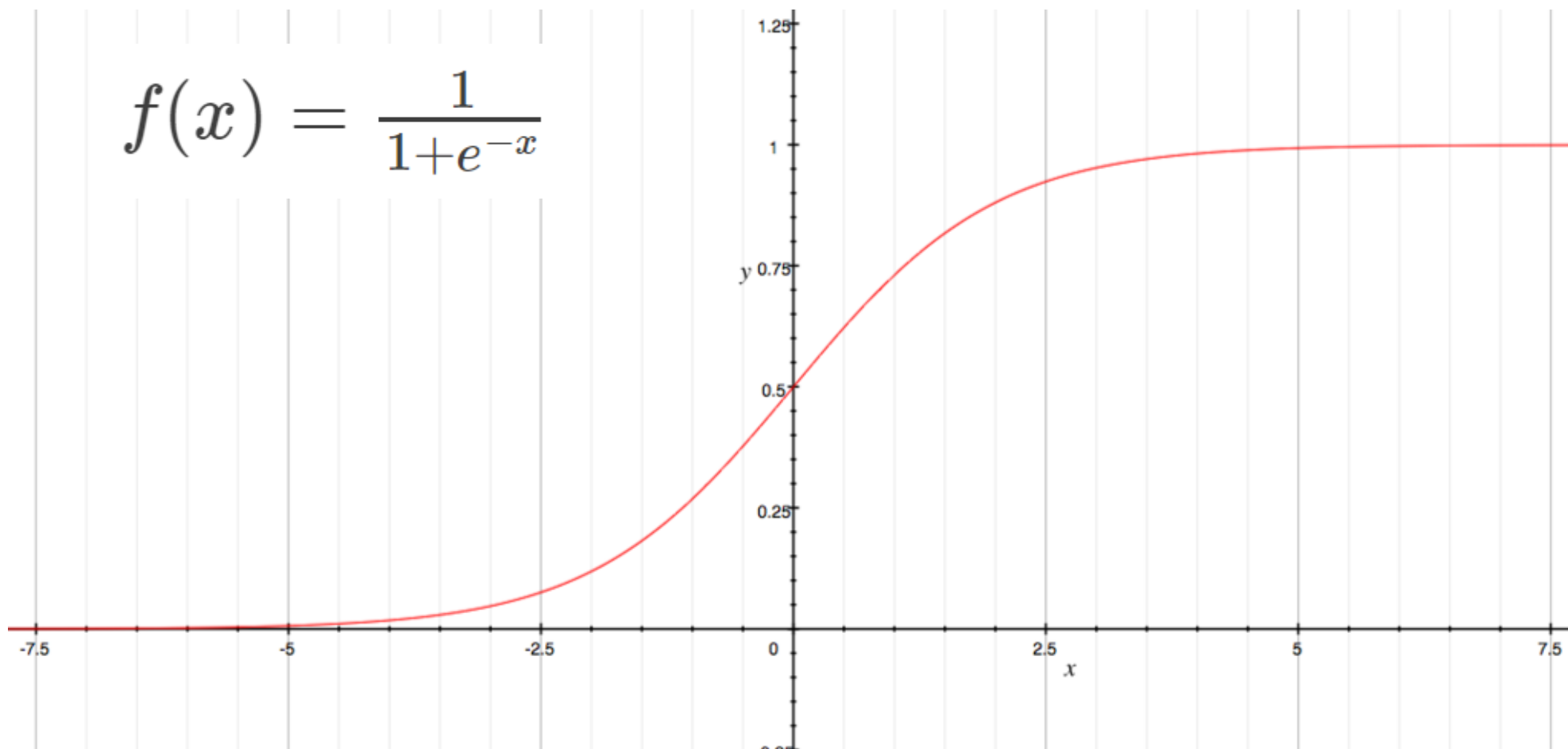
- 入力と出力以外にも、中間にネットワーク（隠れ層）がある。
- 隠れ層が複数ある場合をディープラーニング（深層学習）という。



fという関数の正体

シグモイド関数： $w x + b$ がマイナスに行くほど0に近づき，プラスに行くほど1に近づく．
これが y として出力される．他にも f の種類はあるが，代表的なものがこれ．

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

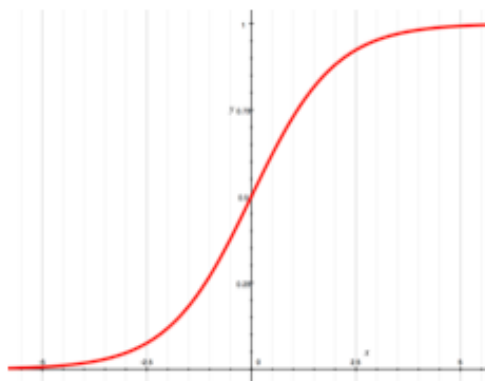


重みwを変えた場合

$$f(x) = \frac{1}{1+e^{-x}} \text{ より, } f(wx + b) = \frac{1}{1+e^{-(wx+b)}}$$

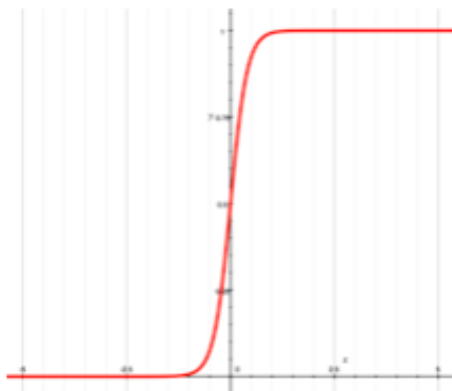
$x=1$, $b=0$ として w を変えると、勾配が変わることがわかる。

$$w=1$$



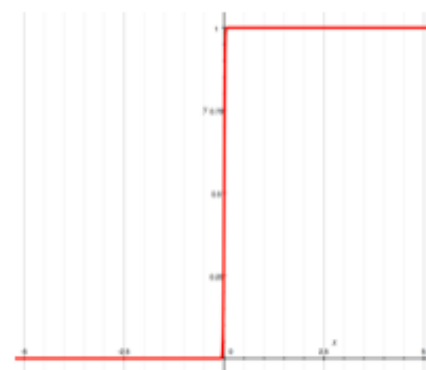
$$y = \frac{1}{1+e^{-x}}$$

$$w=5$$



$$y = \frac{1}{1+e^{-5x}}$$

$$w=100$$

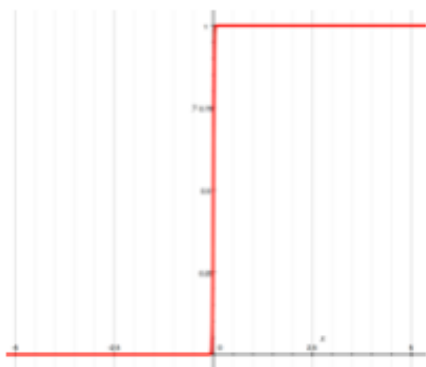


$$y = \frac{1}{1+e^{-100x}}$$

バイアスbを変えた場合

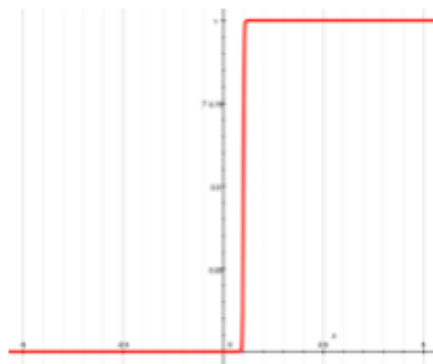
$x=1$, $b=100$ としてbを変えると、段差の位置が変わることがわかる。

$$b=0$$



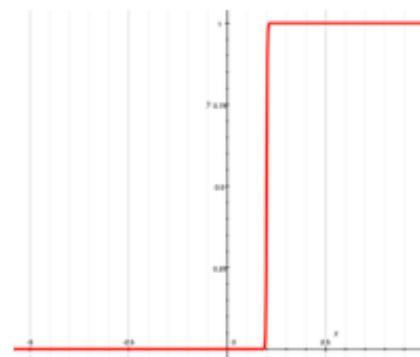
$$y = \frac{1}{1 + e^{-(100x+0)}}$$

$$b=-50$$



$$y = \frac{1}{1 + e^{-(100x-50)}}$$

$$b=-100$$



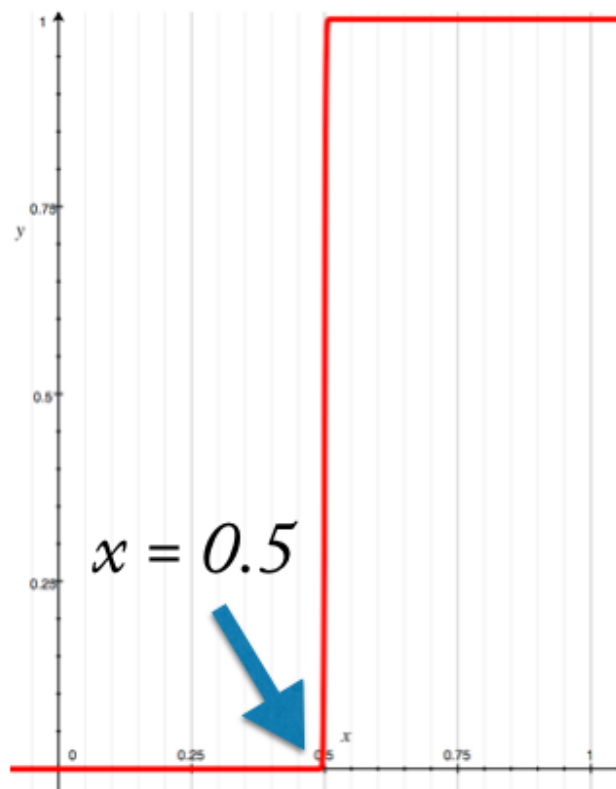
$$y = \frac{1}{1 + e^{-(100x-100)}}$$

段差の位置

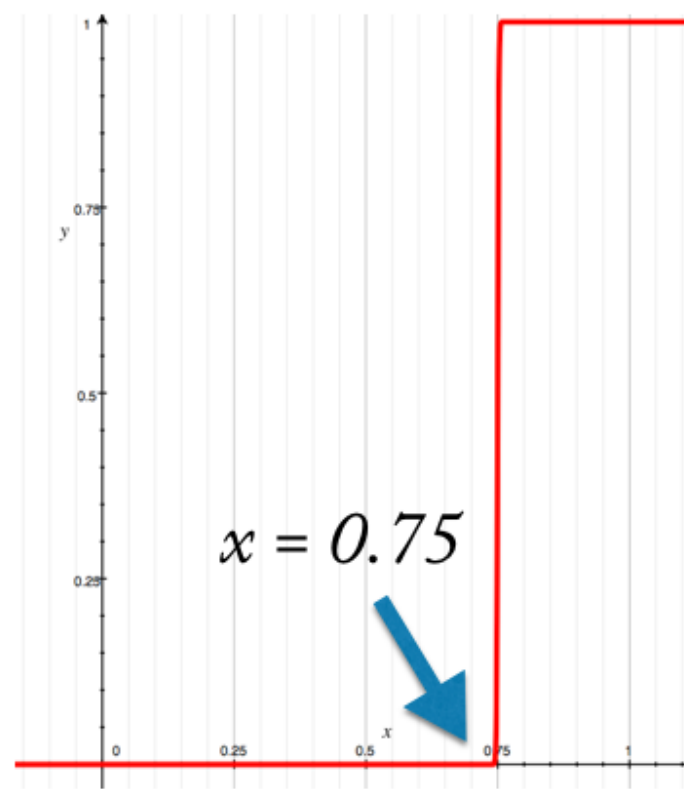
$f(wx + b) = \frac{1}{1+e^{-(wx+b)}}$ の段差は,

$-\frac{b}{w}$ の所に出来る

$$y = \frac{1}{1+e^{-(1000x-500)}}$$



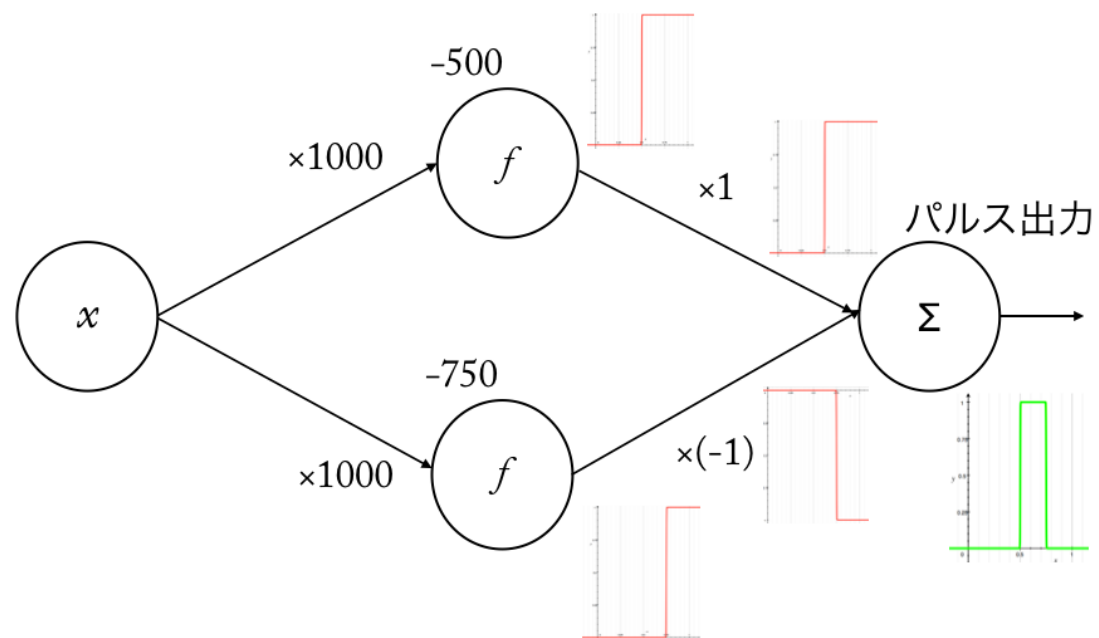
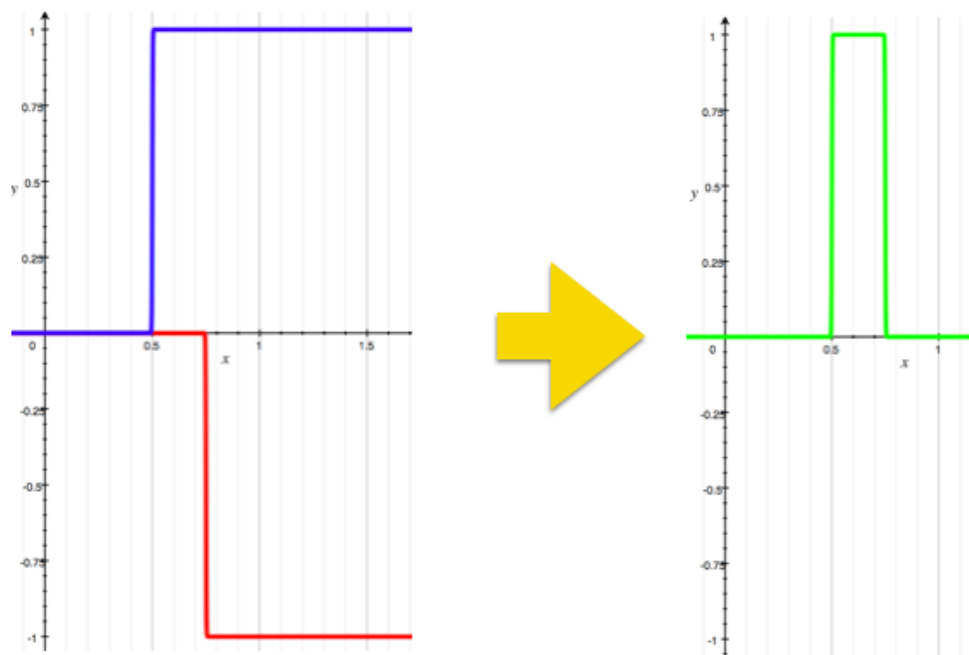
$$y = \frac{1}{1+e^{-(1000x-750)}}$$



任意の波を作成する

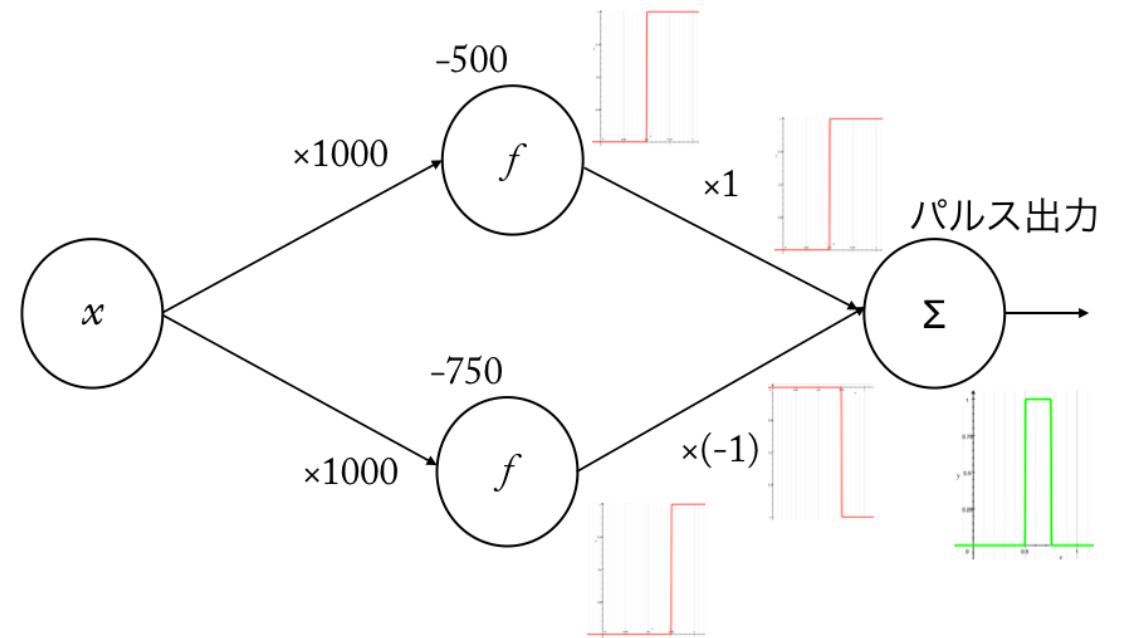
- 2つの波形を合成することで、新しい波形ができる.
- 中間層を増やし、パラメータを調整すれば任意の波に近似できる

$$y = \frac{1}{1+e^{-(1000x-500)}} + y = \frac{1}{1+e^{-(1000x-750)}} \times (-1)$$



学習の方法

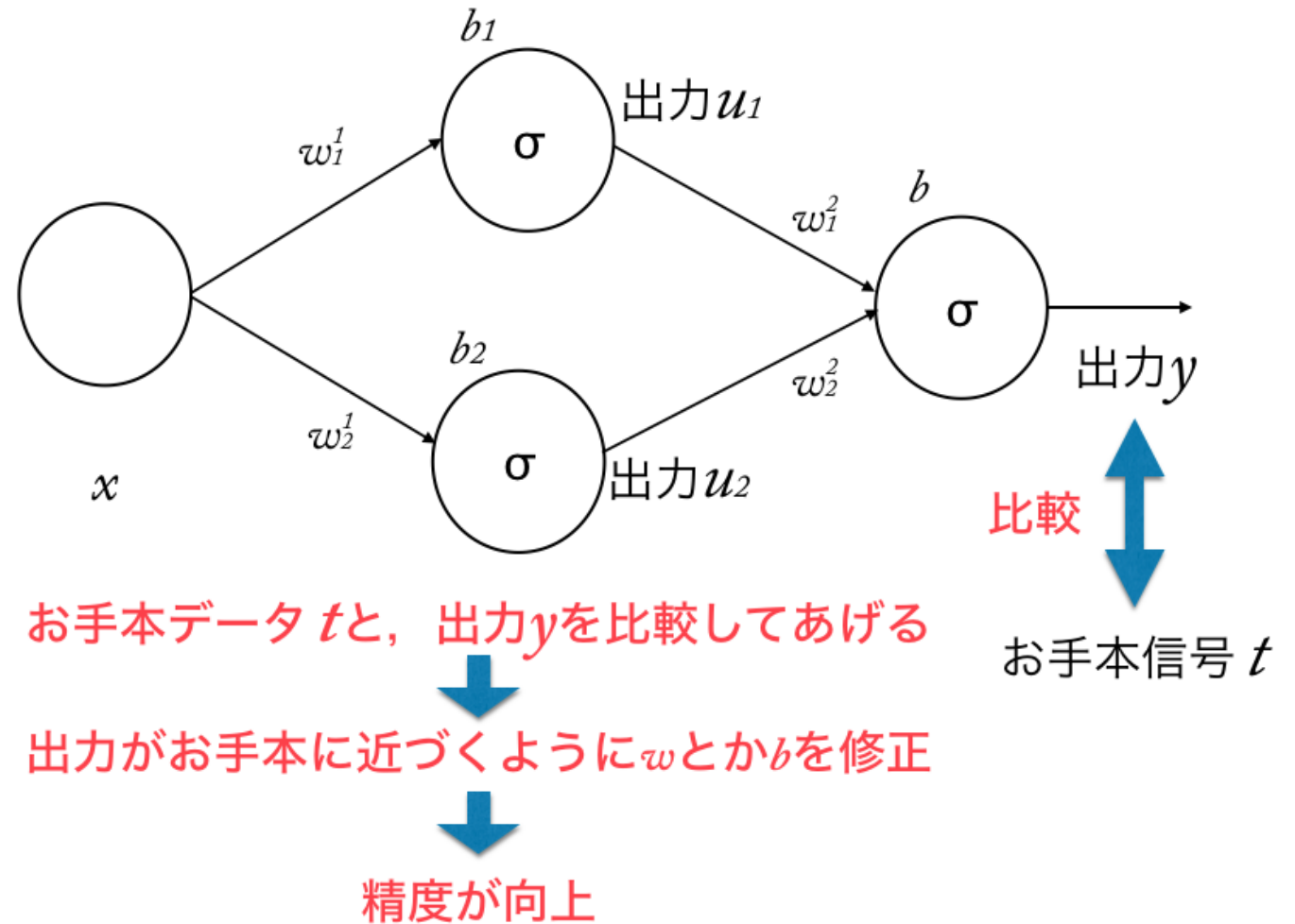
- 人間やセンサーも，内部に何らかの関数があって物事を判別しているはずである．
- 重み w ，バイアス b ，ユニットの数を調整すれば，判別をうまくできるニューラルネットが得られる．



誤差逆伝播法

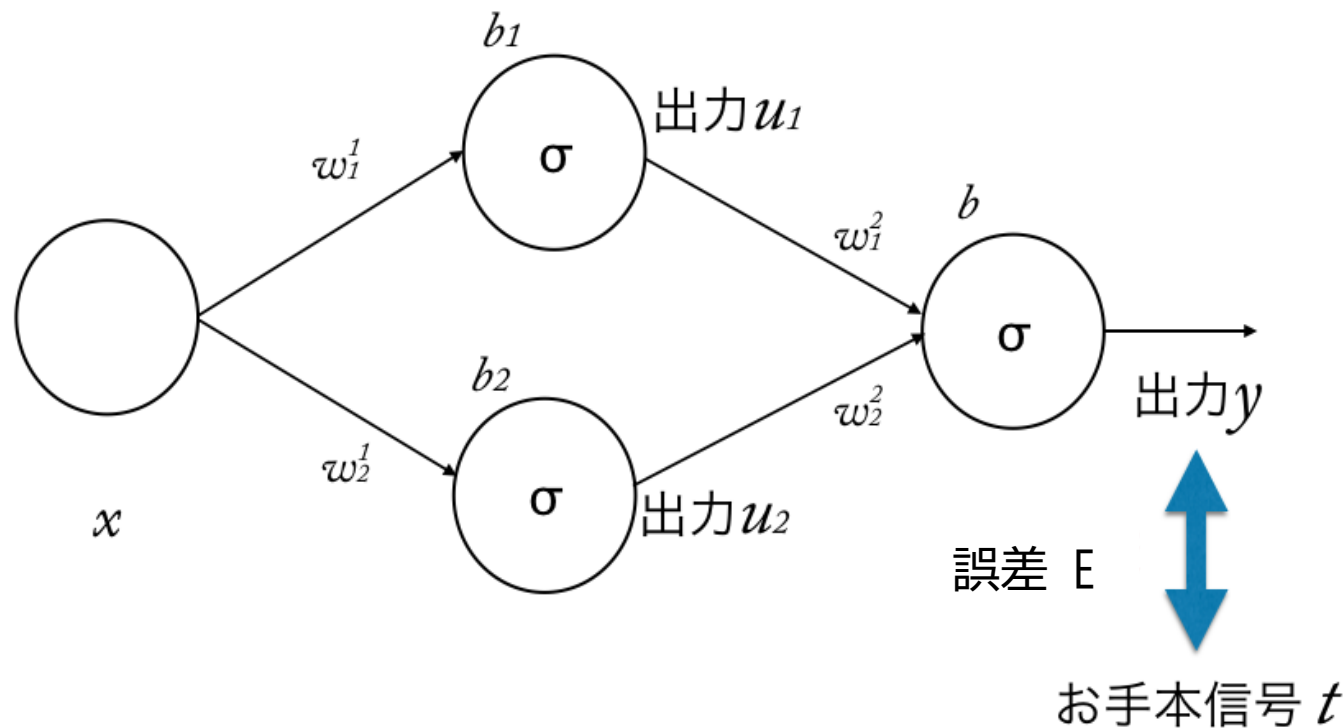
- 出力が間違っていたら、内部パラメータを調整し、出力を確認しながら w と b を調整していけばよい.
- 正解 t との差を考えるが、差にはプラスの場合とマイナスの場合があるため、二乗して考える.
- 結果の誤差が、さかのぼってパラメータ調整に影響するためこの名前がついた.

w^1 は、一層目の重みを示す.



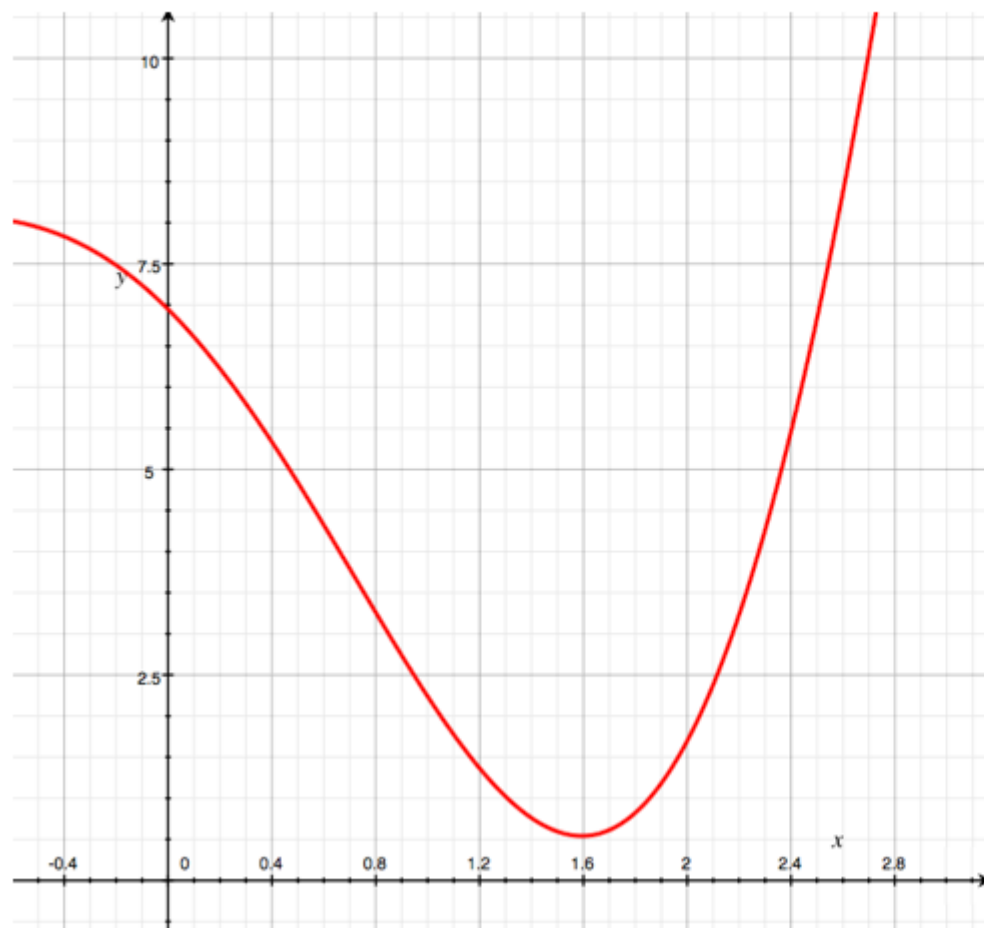
誤差 (Error)

- 誤差を, $E=(y-t)^2$ としたいが, パラメータ調整によるEを以下の形で定義する
- $E = \frac{1}{2}(y-t)^2$
- これを0にできれば, x が与えられたときに正しい t を出力するモデルとなる.



誤差からの修正

誤差 E



$w_1^1 = 1.6$ くらいで誤差が最小



誤差を少なくするために
 w_1^1 を, 1.6に近づけたい

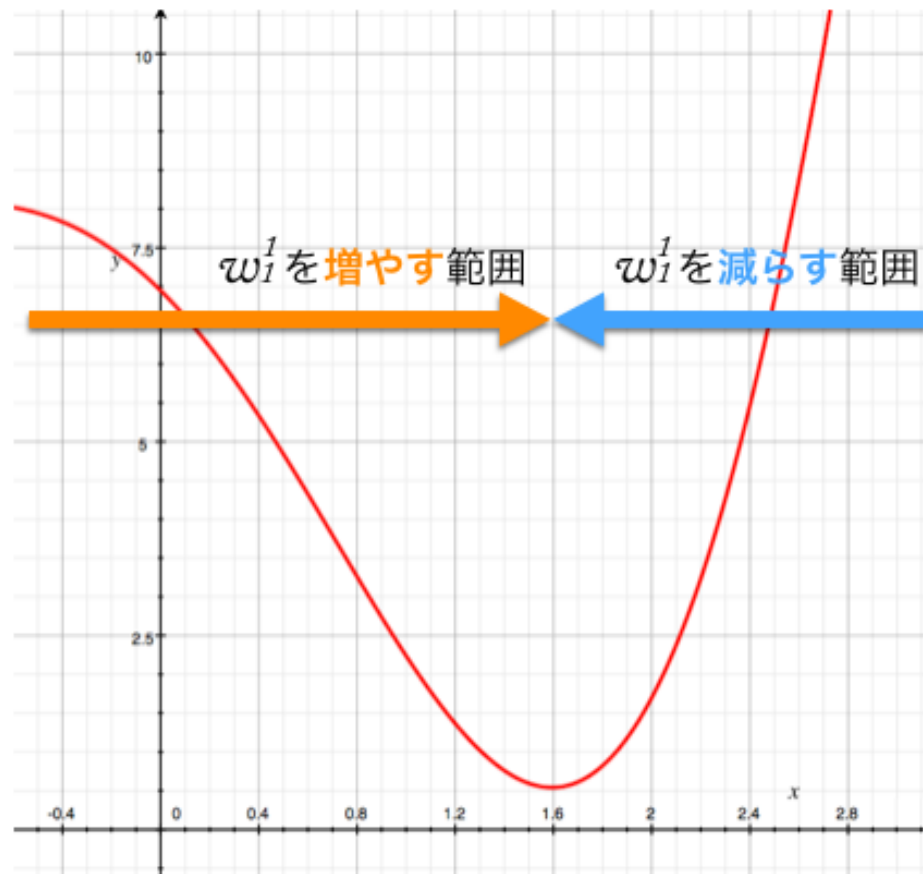
このような極小値を探す
ときのテクニックは？

重み w_1^1

最適な重みを得るために

- 微分で傾きに注目.
- 偏微分 $\frac{\partial E}{\partial w_1^1}$ が、負なら w_1^1 を増やしたい
- 偏微分 $\frac{\partial E}{\partial w_1^1}$ が、正なら w_1^1 を減らしたい

誤差 E



$w_1^1 = 1.6$ くらいで誤差が最小

w_1^1 の値がもし1.6より小さい

→ w_1^1 を増やす

w_1^1 の値がもし1.6より大きい

→ w_1^1 を減らす

パラメータの更新

- 偏微分 $\frac{\partial E}{\partial w_1^1}$ が負なら, w_1^1 を増加させる
- 偏微分 $\frac{\partial E}{\partial w_1^1}$ が正なら, w_1^1 を減少させる
- どの程度増減させるかは, α で調整する

$$\underbrace{w_1^{'1}}_{\text{更新後}} = \underbrace{w_1^1}_{\text{更新前}} - \underbrace{\alpha}_{\text{学習率}} \frac{\partial E}{\partial w_1^1} \quad (\alpha > 0)$$

必要な数学的知識

- 偏微分：ある変数1つに注目し，その他の変数は定数として微分.
 - $g(a, b, c) = a^2 + b^2 + c^2$ のとき， a について偏微分すると，
 - $\frac{\partial g}{\partial a} = 2a$
- 合成関数の微分： y が u の関数で， u が x の関数であるとき， y を x で微分
 - $\frac{dy}{dx} = \frac{dy}{du} \frac{du}{dx}$

連鎖律

- 以下の場合の微分を考える

$$y(w_1, w_2) = (w_1 x_1 + w_2 x_2)^2$$

- これを，2つの関数の合成とみなす

$$y = s^2$$

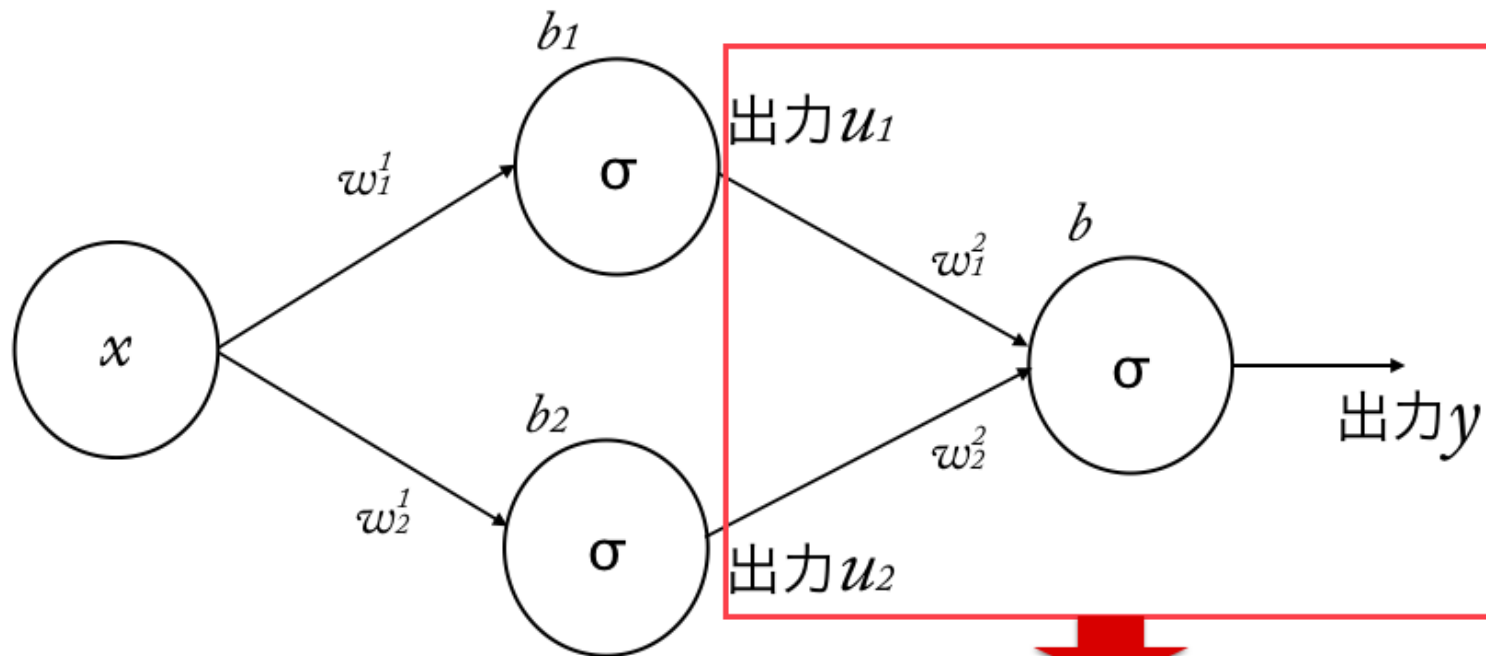
$$s = w_1 x_1 + w_2 x_2$$

- この状態で偏微分をすると，以下のようなになる．

$$\frac{\partial y}{\partial w_1} = \frac{\partial y}{\partial s} \frac{\partial s}{\partial w_1}$$

これが連鎖率．
偏微分に分解して掛け合わせてあげれば良い．

パラメータ w_k^2 の更新



Eにyがあり，yにwがある。
これを連鎖律で表すと右のようになる。
つまり，ほかを固定して w_k^2 だけを動か
し，Eを最小にする狙い。

参考：シグモイド関数の微分

$$\begin{aligned}\sigma(x)' &= \left(\frac{1}{1+e^{-x}}\right)' = \frac{-(1+e^{-x})'}{(1+e^{-x})^2} = \frac{e^{-x}}{(1+e^{-x})^2} = \frac{e^{-x}}{(1+e^{-x})(1+e^{-x})} = \frac{1+e^{-x}-1}{(1+e^{-x})} \cdot \frac{1}{(1+e^{-x})} \\ &= \left(1 - \frac{1}{(1+e^{-x})}\right) \frac{1}{(1+e^{-x})} = (1 - \sigma(x))\sigma(x)\end{aligned}$$

$$\text{誤差: } E = \frac{1}{2}(y-t)^2$$

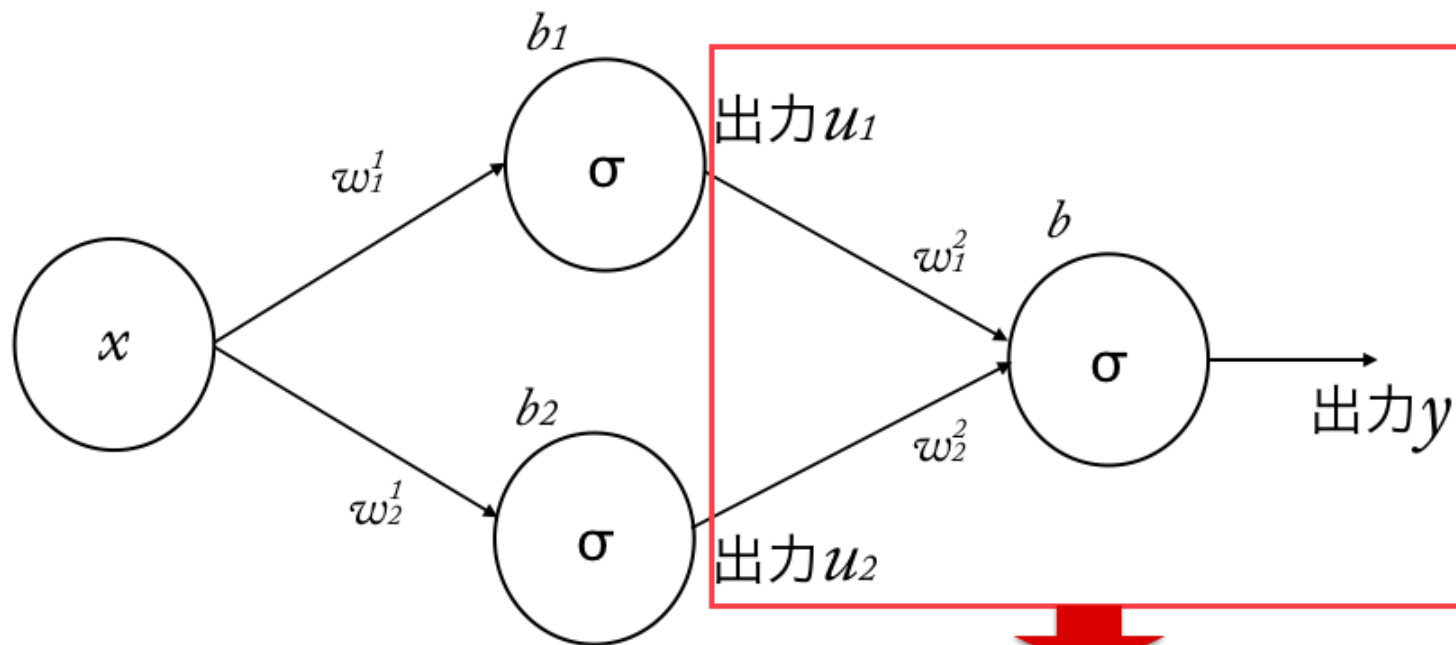
$$\text{出力: } y = \sigma(w_1^2 u_1 + w_2^2 u_2 + b)$$

$$\frac{\partial E}{\partial y} = y - t$$

$$\frac{\partial y}{\partial w_k^2} = (1-y) \cdot y \cdot u_k$$

$$\frac{\partial E}{\partial w_k^2} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial w_k^2} = (y-t) \cdot (1-y) \cdot y \cdot u_k$$

パラメータb の更新



bに関して同様に偏微分する

$$\text{誤差: } E = \frac{1}{2}(y-t)^2$$

$$\frac{\partial E}{\partial y} = y-t$$

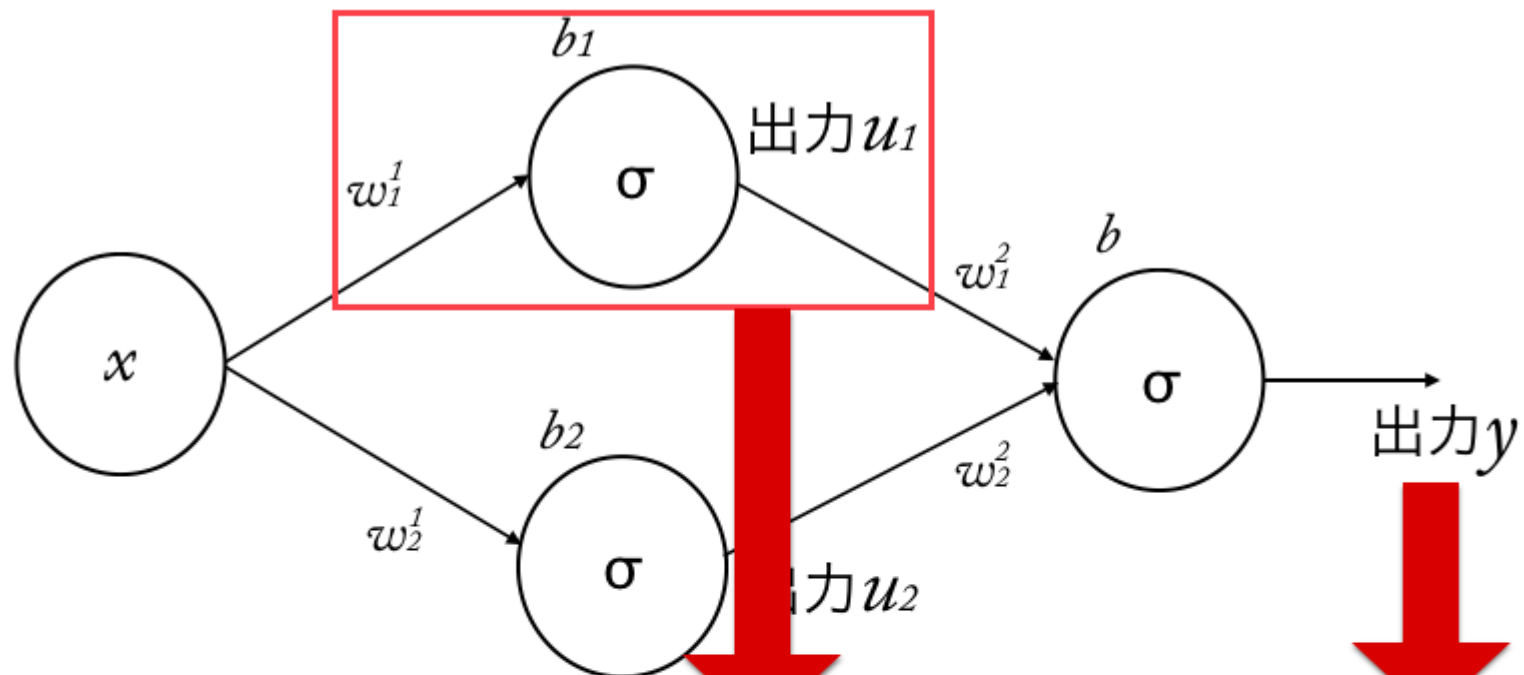
$$\text{出力: } y = \sigma(w_1^2 u_1 + w_2^2 u_2 + b)$$

$$\frac{\partial y}{\partial b} = (1-y) \cdot y$$

(ここは先程のシグモイド関数の微分を参照)

$$\frac{\partial E}{\partial b} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial b} = (y-t) \cdot (1-y) \cdot y$$

パラメータ w_k^1 の更新



連鎖律を用いる.

出力 y に u_1 があり, u_1 に w_k^1 があるというつながり.

平たく言えば, 偏微分を次々と掛けてあげる.

誤差

$$E = \frac{1}{2}(y-t)^2$$
$$\frac{\partial E}{\partial y} = y-t$$

1層目

$$\text{出力: } u_1 = \sigma(w_1^1 x + b_1)$$
$$\text{一般: } u_k = \sigma(w_k^1 x + b_k)$$
$$\frac{\partial u_k}{\partial w_k^1} = (1-u_k) \cdot u_k \cdot x$$

最終出力

$$y = \sigma(w_1^2 u_1 + w_2^2 u_2 + b)$$
$$\frac{\partial y}{\partial u_k} = (1-y) \cdot y \cdot w_k^2$$

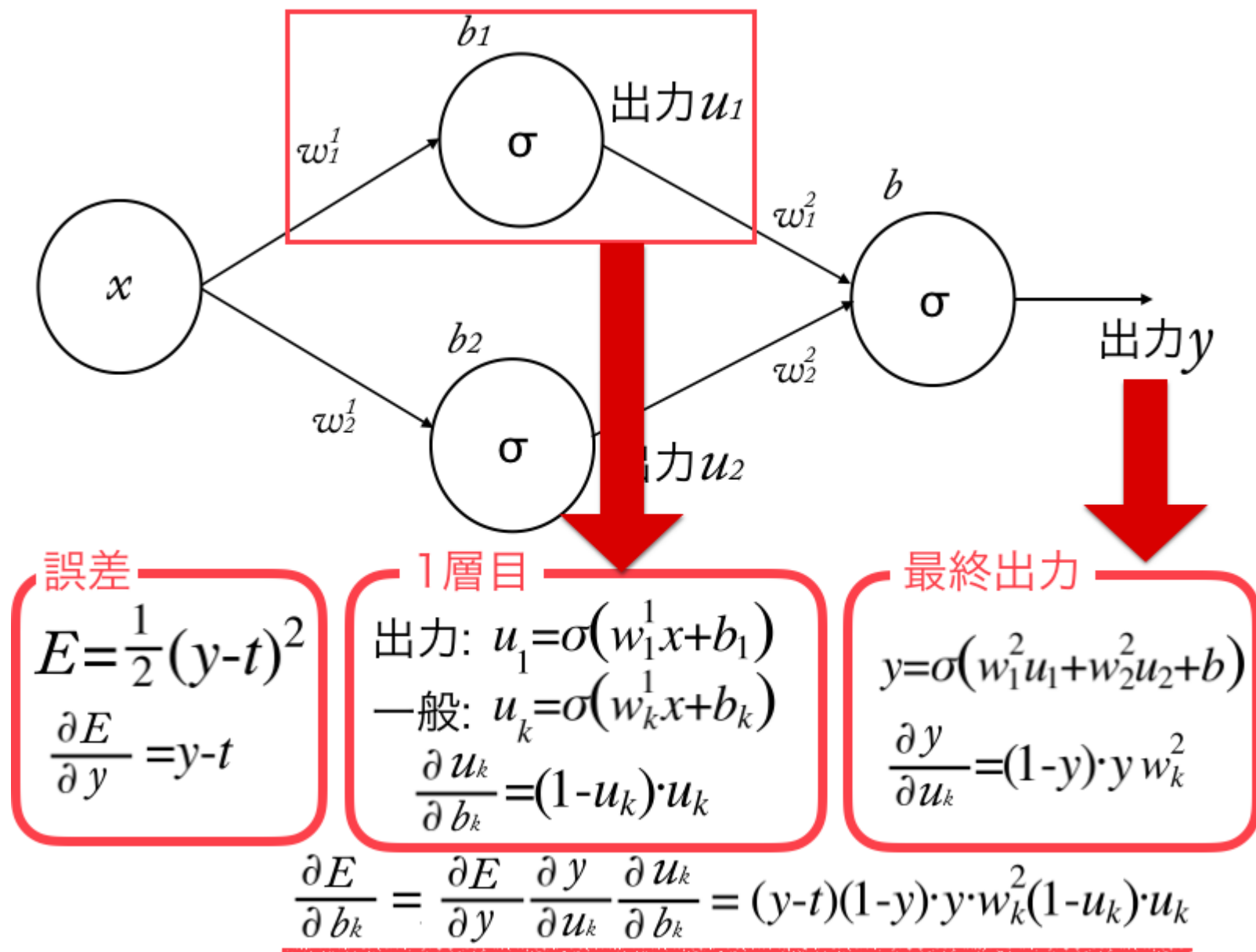
$$\frac{\partial E}{\partial w_k^1} = \frac{\partial E}{\partial y} \frac{\partial y}{\partial u_k} \frac{\partial u_k}{\partial w_k^1} = (y-t)(1-y) \cdot y \cdot w_k^2 (1-u_k) \cdot u_k \cdot x$$

パラメータ b_k の更新

b に対しても、同様に連鎖律を用いる.

狙いは、他のパラメータを固定して b だけを動かして E を最小にする.

少しずつ、学習率に従って更新する.



パラメータ更新のまとめ

➡ 以下の値で、重みやバイアスを更新していく

w_k^1 を $w_k^1 - \alpha \frac{\partial E}{\partial w_k^1}$ に更新

$$\frac{\partial E}{\partial w_k^1} = (y-t)(1-y) \cdot y \cdot w_k^2 (1-u_k) \cdot u_k x$$

w_k^2 を $w_k^2 - \alpha \frac{\partial E}{\partial w_k^2}$ に更新

$$\frac{\partial E}{\partial w_k^2} = (y-t) \cdot (1-y) \cdot y \cdot u_k$$

b_k を $b_k - \alpha \frac{\partial E}{\partial b_k}$ に更新

$$\frac{\partial E}{\partial b_k} = (y-t)(1-y) \cdot y \cdot w_k^2 (1-u_k) \cdot u_k$$

b を $b - \alpha \frac{\partial E}{\partial b}$ に更新

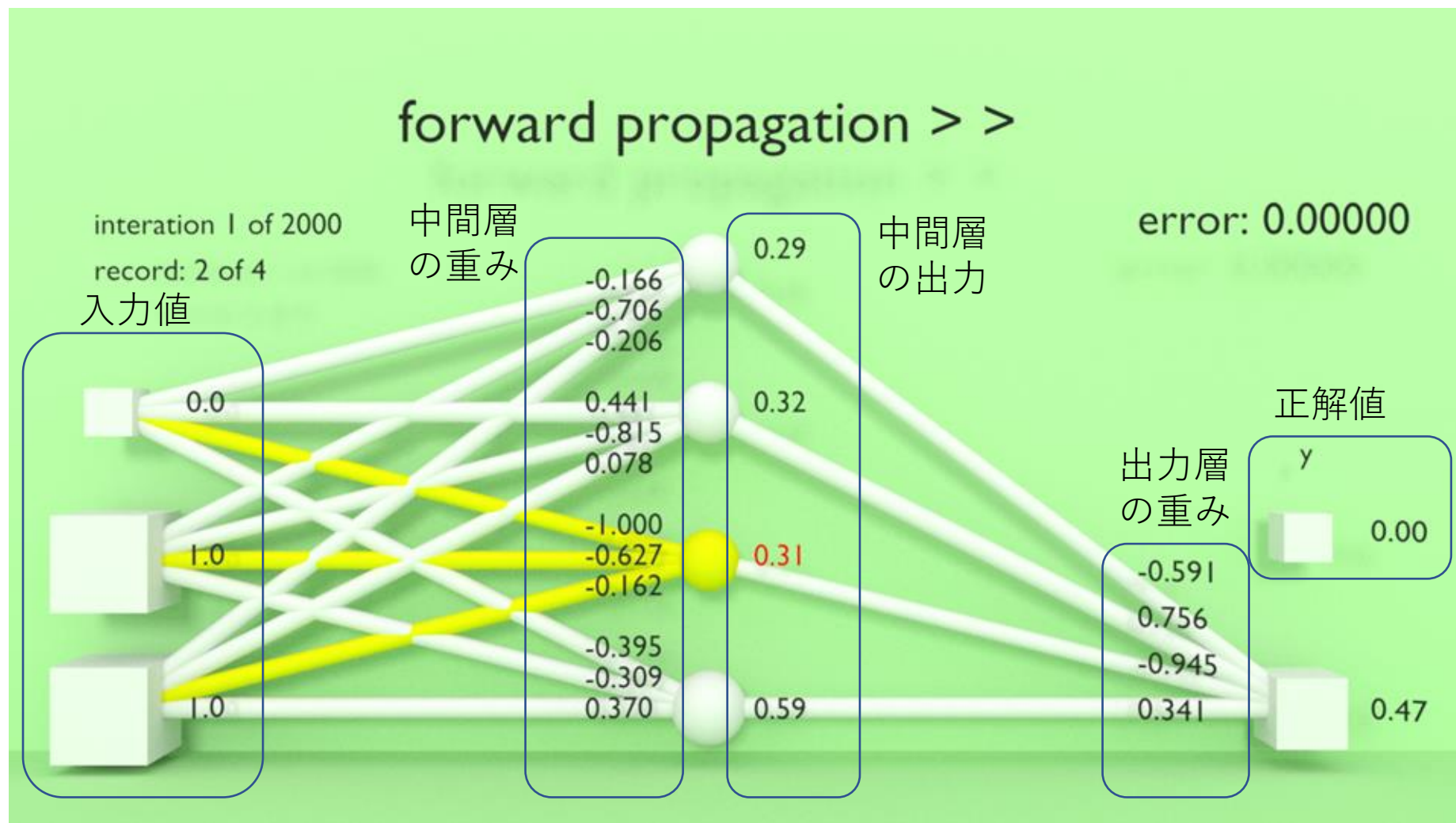
$$\frac{\partial E}{\partial b} = (y-t) \cdot (1-y) \cdot y$$

コンピュータは繰り返し計算が得意なので、あとは数字を入れるだけ

誤差逆伝播法の可視化

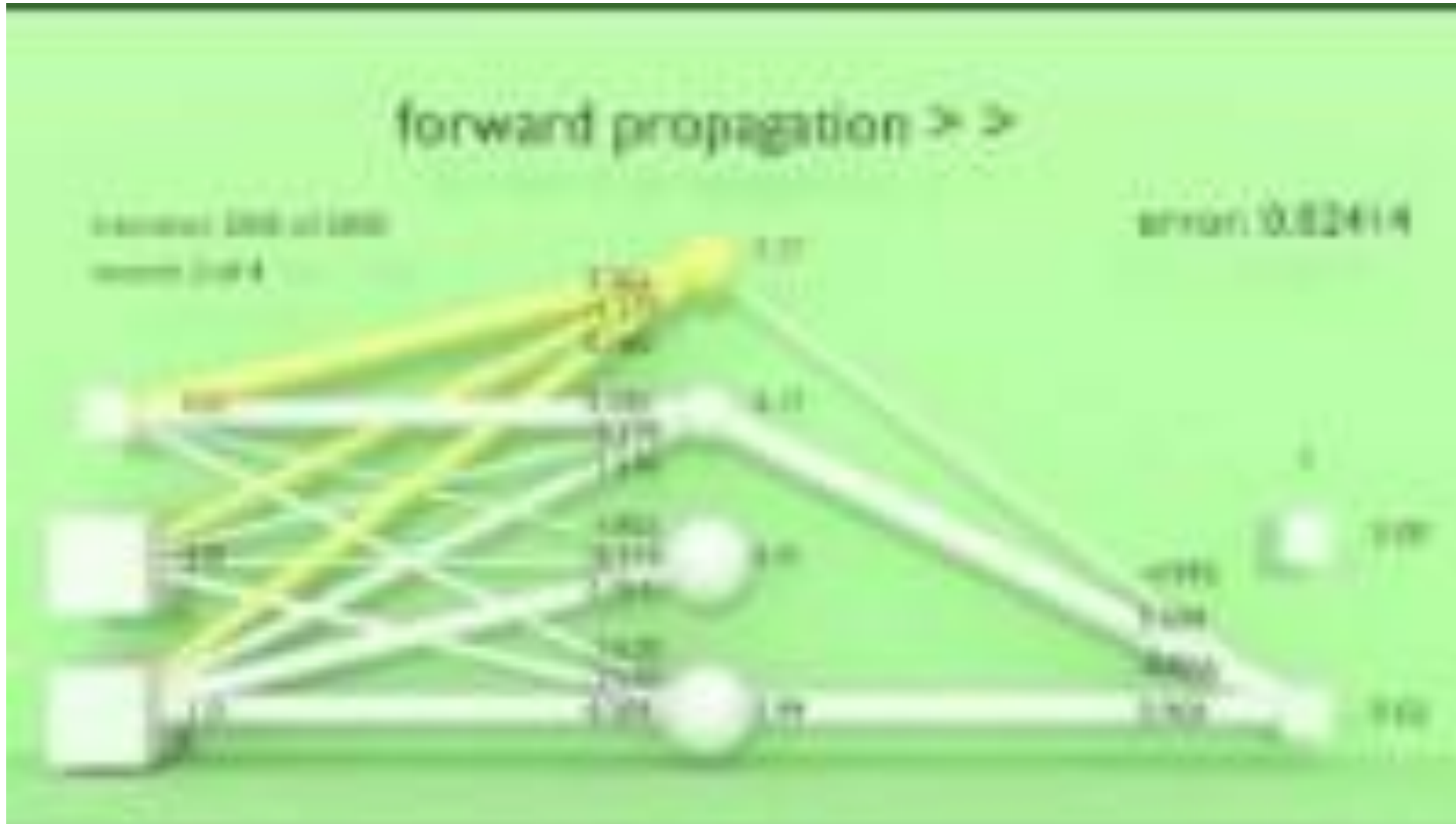
以下のデータを学習.
出力の誤差を用いて
back propagationで
ウェイトを調整する.
初回はランダム値.

Inputs				Output
0	0	1	0	
1	1	1	1	
1	0	1	1	
0	1	1	0	



誤差逆伝播法の可視化

同じデータを2000回与え、少しずつ重みを調整



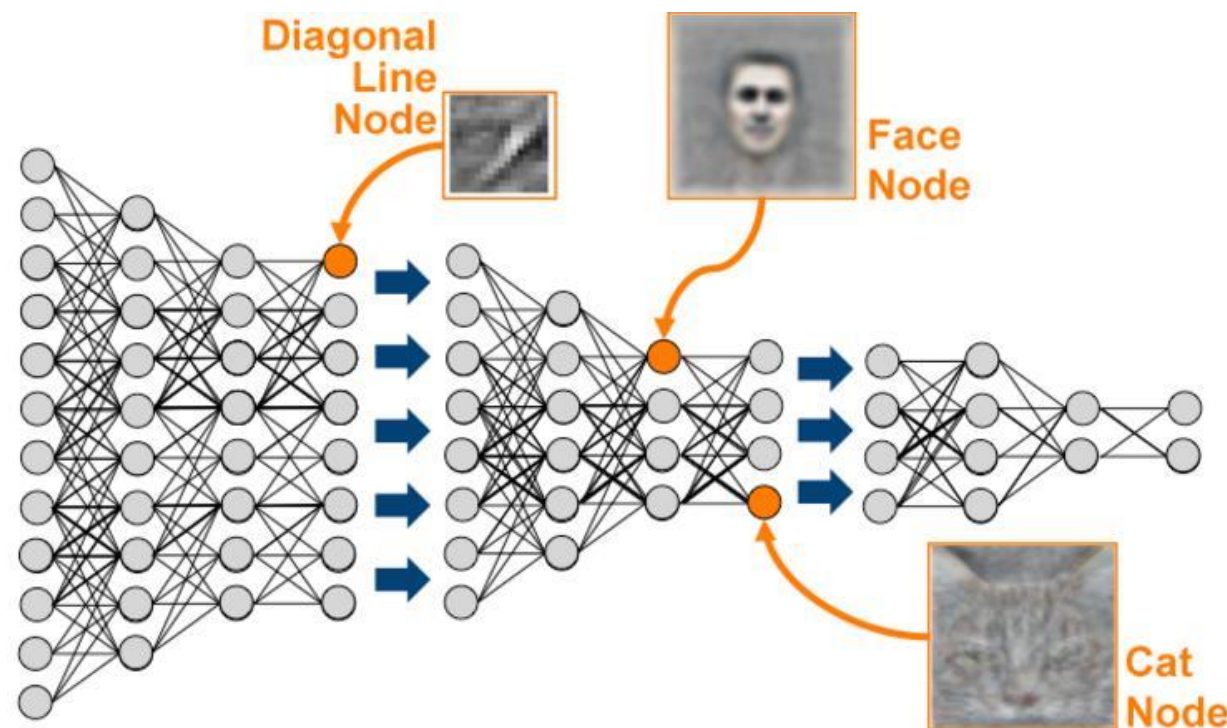
簡単な使いみちの例



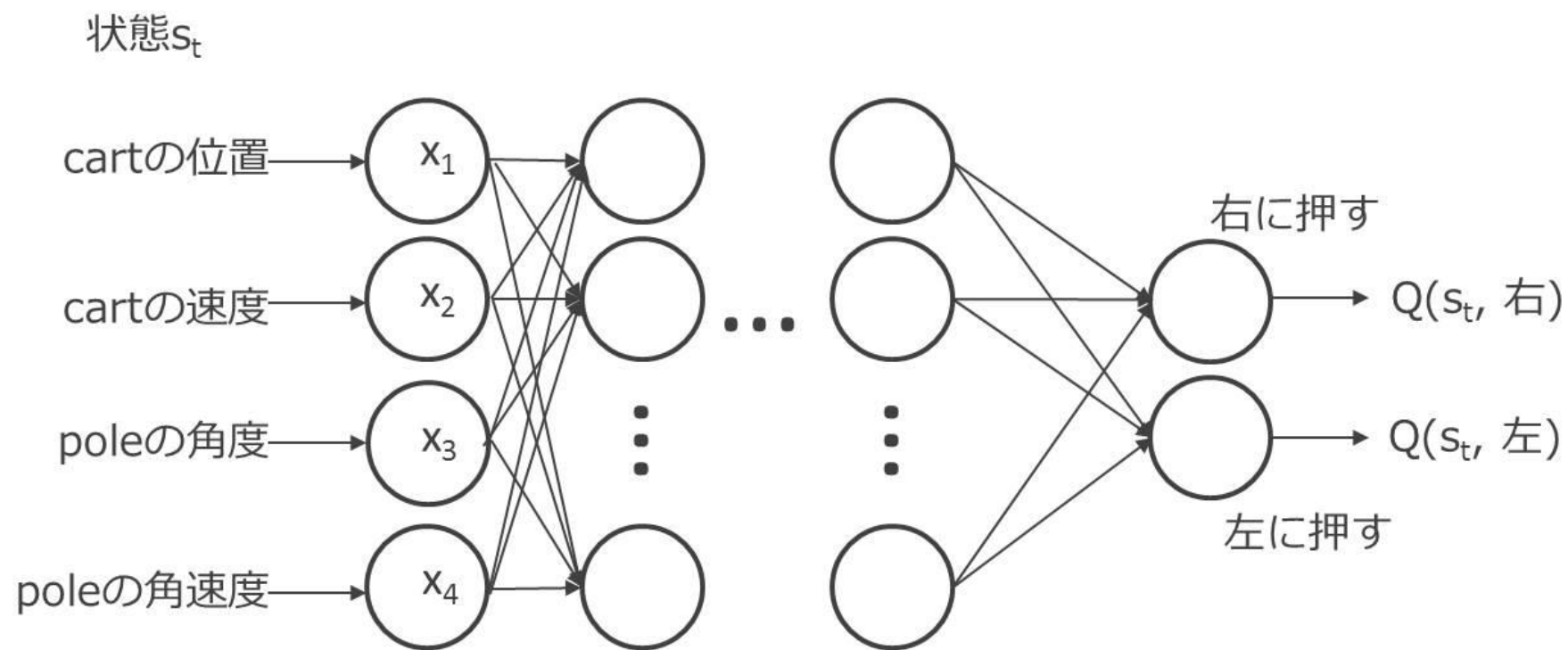
休憩

ディープラーニングの流行

- 2012年、画像識別の正確さを競う大会（ILSVRC）で、頭打ちに近かった識別精度を大幅に超えて、トロント大学がAlexNetというニューラルネットで優勝.
- 2012年、Googleからコンピュータが猫の概念を獲得したと発表.
- 正確には、猫画像に反応して発火するニューラルネットを作成した.



DQN

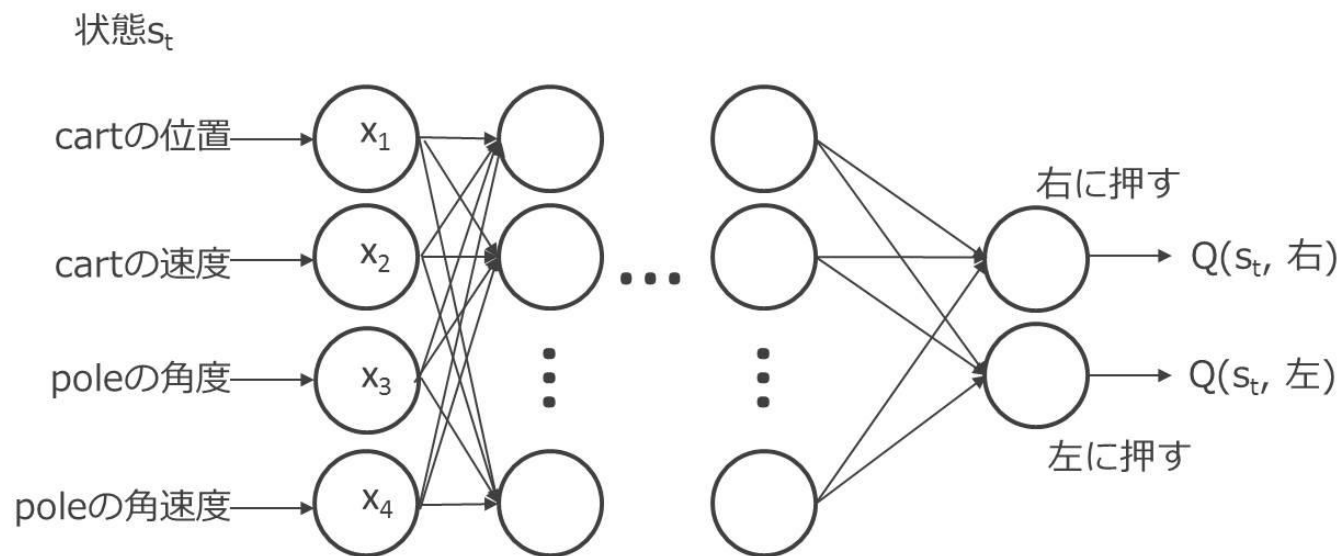


バックプロパゲーション

$$E(R_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t))$$

Fixed target Q-Network

- Q値はネットワークを利用して出すため、訓練データのQ値も学習ごとに更新される。
- 訓練データが持つ目的変数の値が、一定ではなく毎回更新されてしまう。
 - Fixed target Q-Networkとして、一定期間別のニューラルネットからQ値を得る。
 - target networkは、たまに同期してあげる。



バックプロパゲーション

$$E(R_{t+1} + \gamma \max_a Q(s_{t+1}, a_t) - Q(s_t, a_t))$$

畳み込みニューラルネットワーク

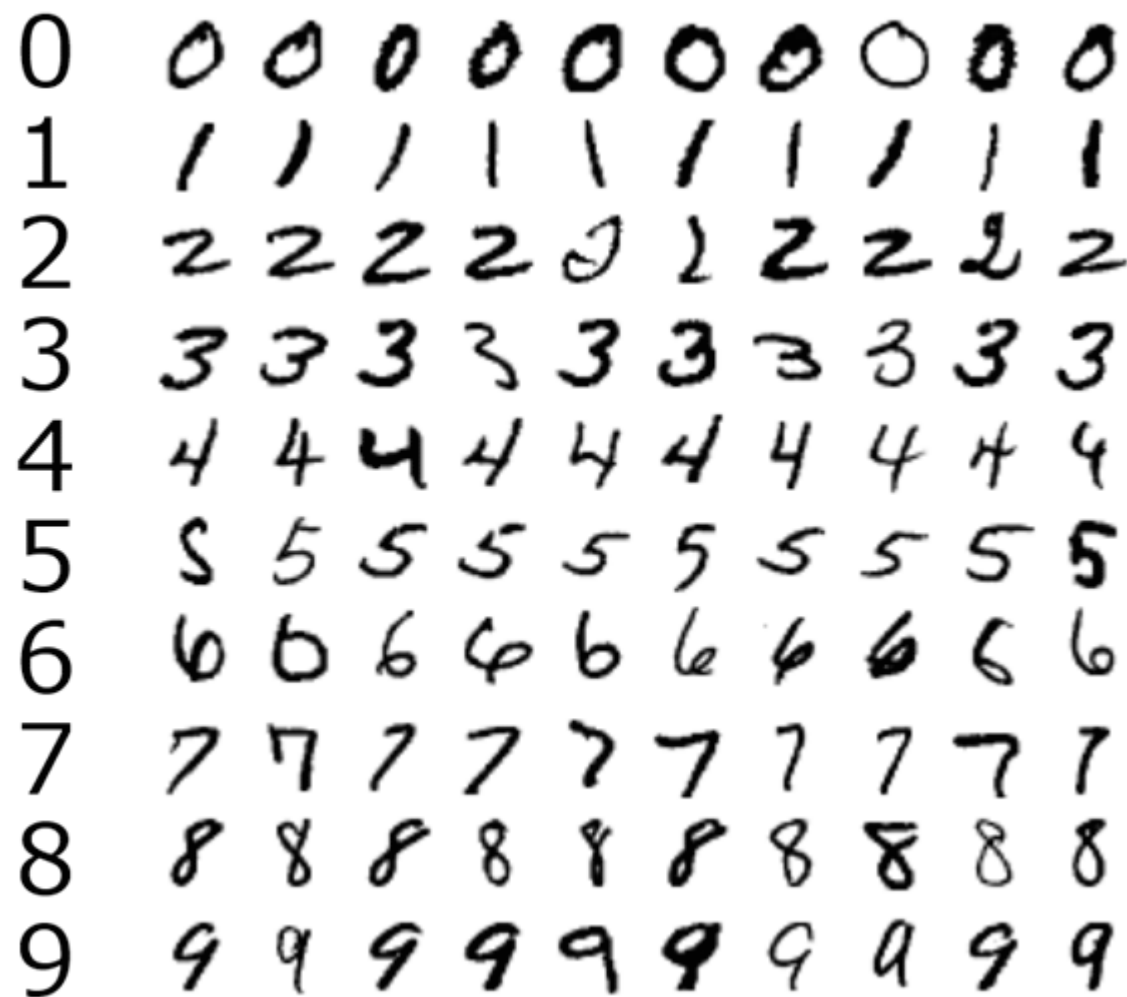
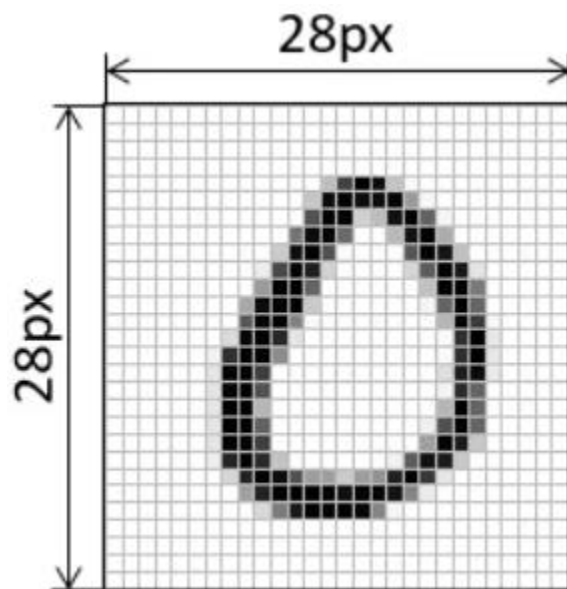
- Convolutional Neural Network (CNN) は，画像処理でよく使われる手法.

画像識別
(セマンティック セグメンテーション)



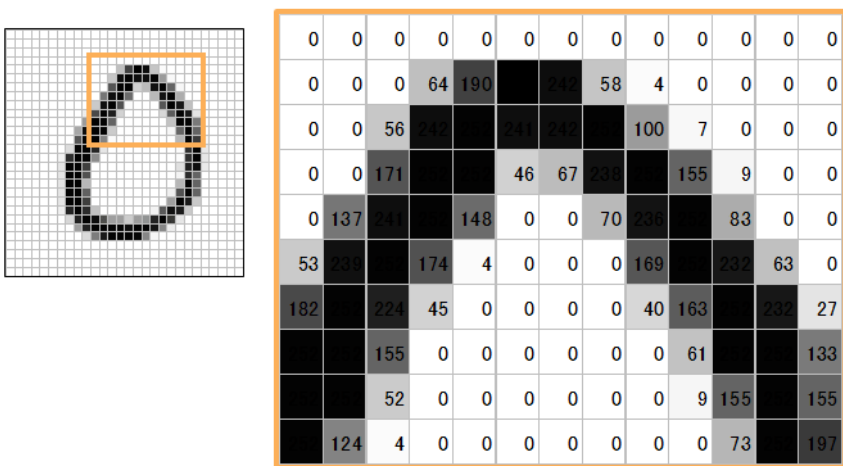
タスク例

- MNISTというデータセットを用いて手書きの数字を識別する



データセットの詳細

- 手書き画像には，0～9までのラベルが付いており，各ピクセルは色の濃さに応じて0～255の値がある

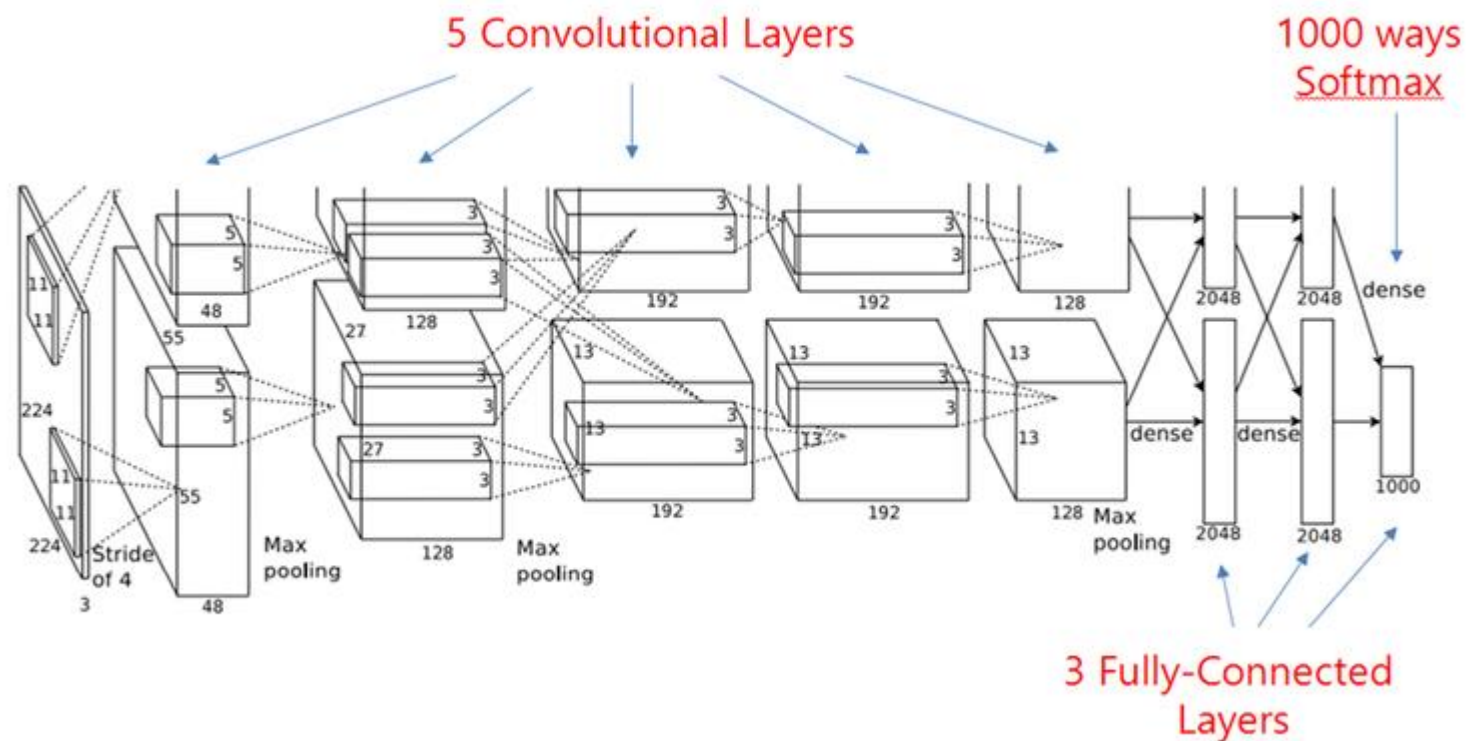


一辺は28ピクセルなので，784個の数値によって構成されている。



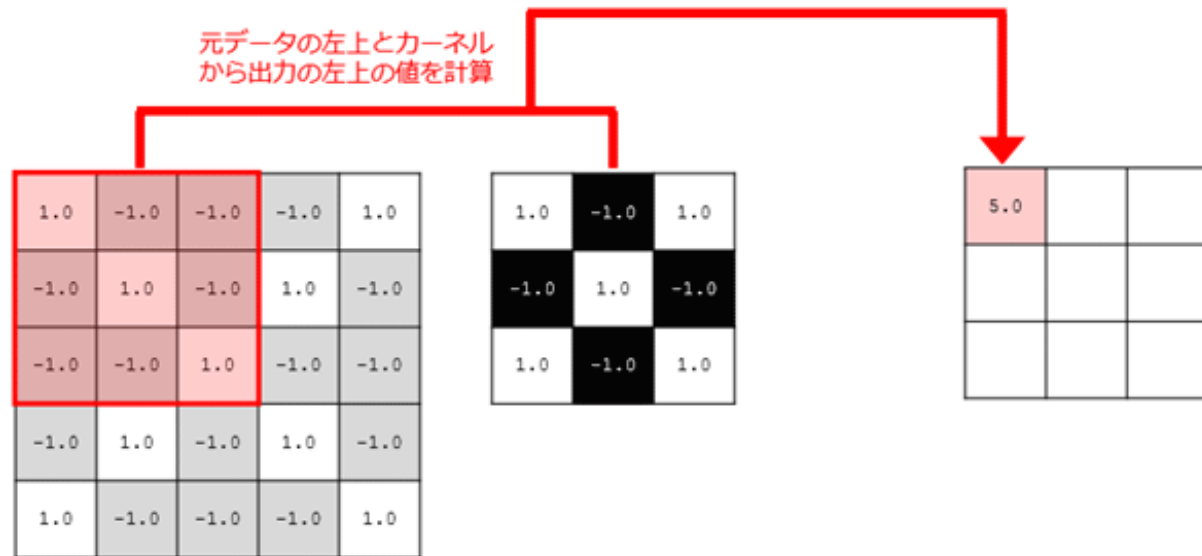
AlexNetで使されたモデル

- 下図に出てくる用語を解説する。
- また、AlexNetはRGBに対してだが、解説では白黒の1次元とする。



畳み込み (Convolutional)

- 画像に対して、あるブロックだけ虫眼鏡のように注目し、対応するピクセルとカーネル（フィルター）をかけ合わせる。



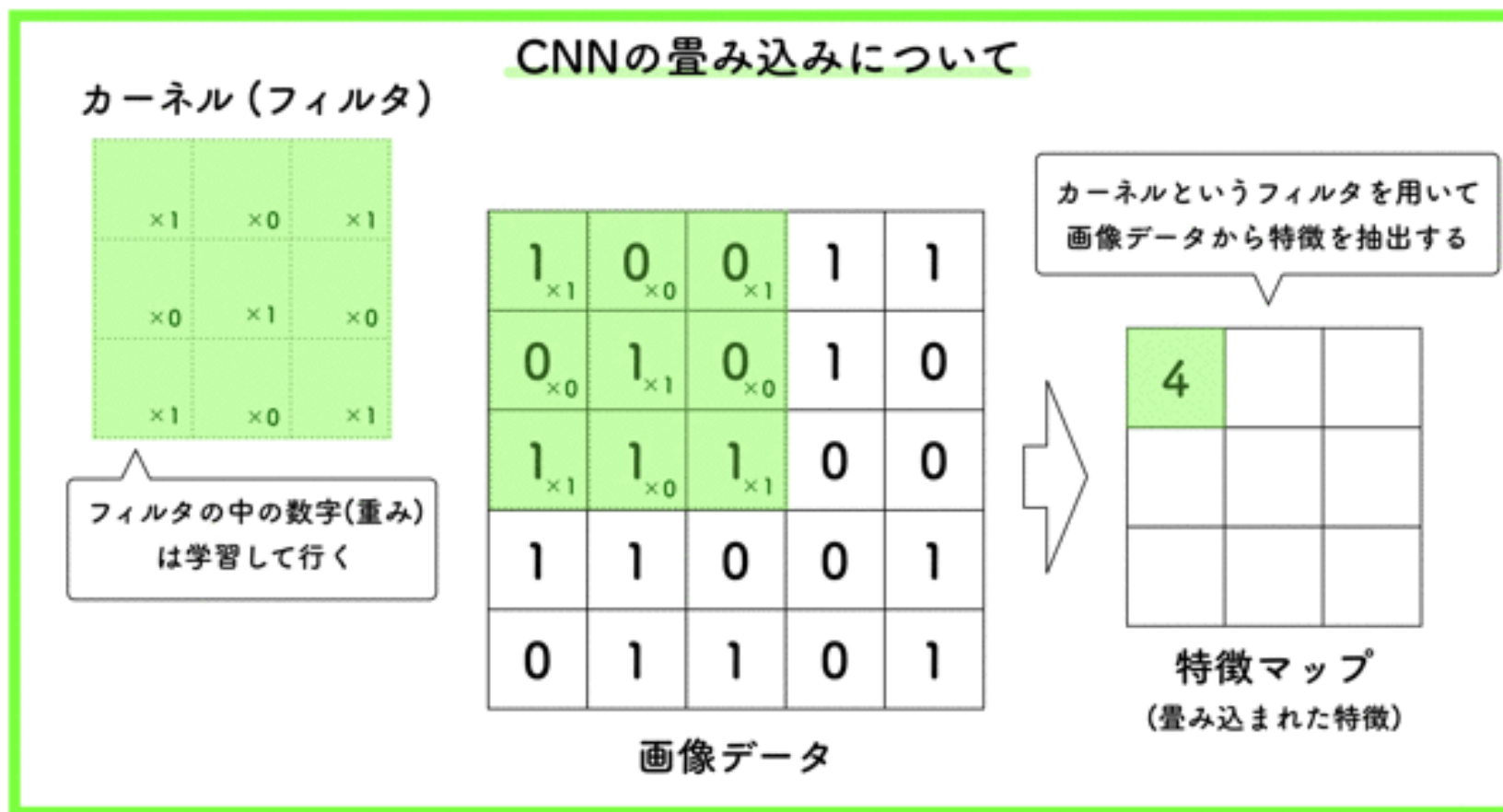
この例では、 3×3 のカーネルを適用し、一つずつ左にずらすことによって 3×3 の特徴マップを作成していく。



画像の特徴を、特徴マップという元画像より小さなデータとして変換（畳み込み）している。

動画によるイメージ

本来は、白黒の濃さに応じて0~255の値が入っていたり、RGBとして3つの数字が入っている。



ゼロパディング

- 端のピクセルは，中央に比べるとカーネルが当たりにくいいため特徴が考慮されにくい.
- そのため，周りを0で埋めたとしてカーネルを当てることで端の特徴も考慮できる.

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	0	1	0	0
0	0	0	1	0	0	0
0	0	0	0	0	0	0

ストライド

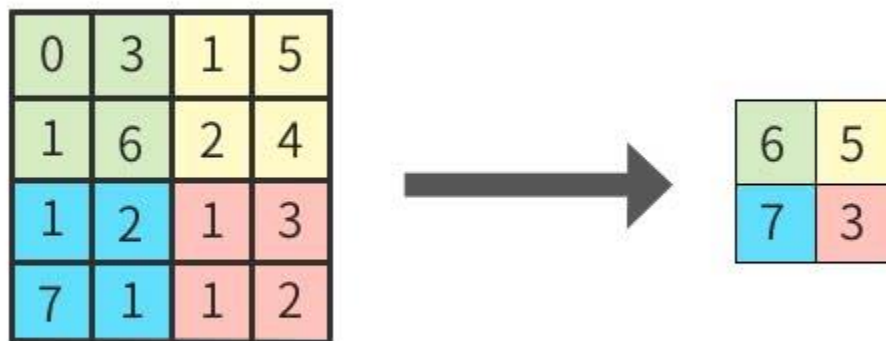
- カーネルを動かす間隔を調整. 以下の場合には2マスずらしている

0 _{×1}	1 _{×0}	1	0
1 _{×0}	0 _{×1}	1	0
1	0	1	0
0	1	0	0

0	

Max Pooling

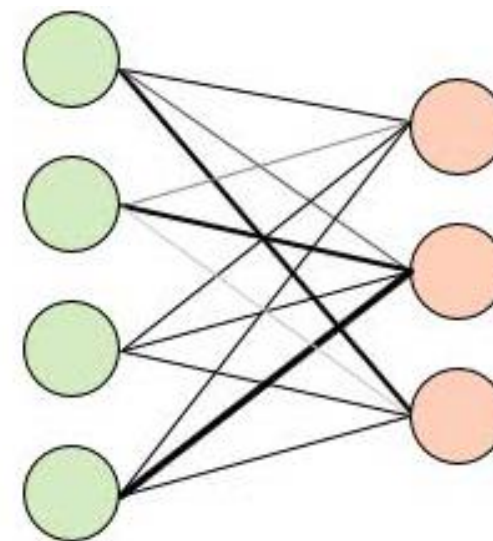
- 何らかの操作により，情報を変換して扱いやすくする
- Max Poolingでは，小領域に対して最大値だけを抽出してダウンサイジングする．



Max Pooling

全結合層

- 出力層の近くで使われることが多い.
- 最終的に, 出力したいクラスのみだけユニットを用意する
- 出力層での出力値が, そのカテゴリと予測される確率となる



Fully Connected層

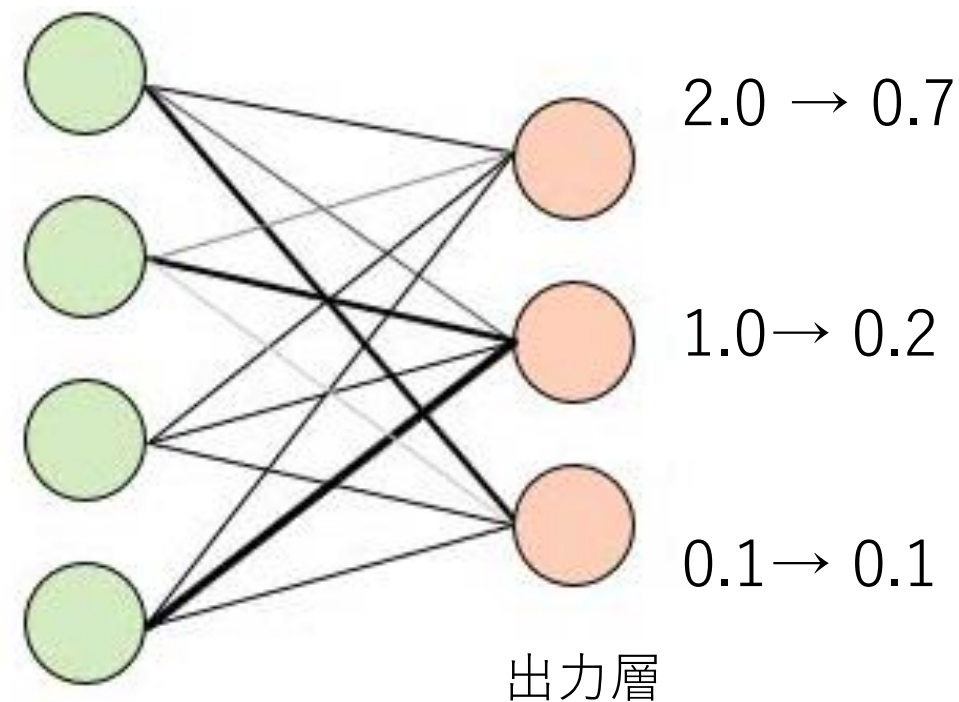
SoftMax

- 最終的な出力の数値の合計が 1 になるように調整してくれる仕組み.
- 確率として, 人間が理解しやすい数字となる

ソフトマックス関数S

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

yは出力された数値を示す.



実演

- CNNによる手書き文字の識別

<https://www.dropbox.com/s/w99pwdkge7kdgrp/MNIST.ipynb?dl=0>

[illegible]

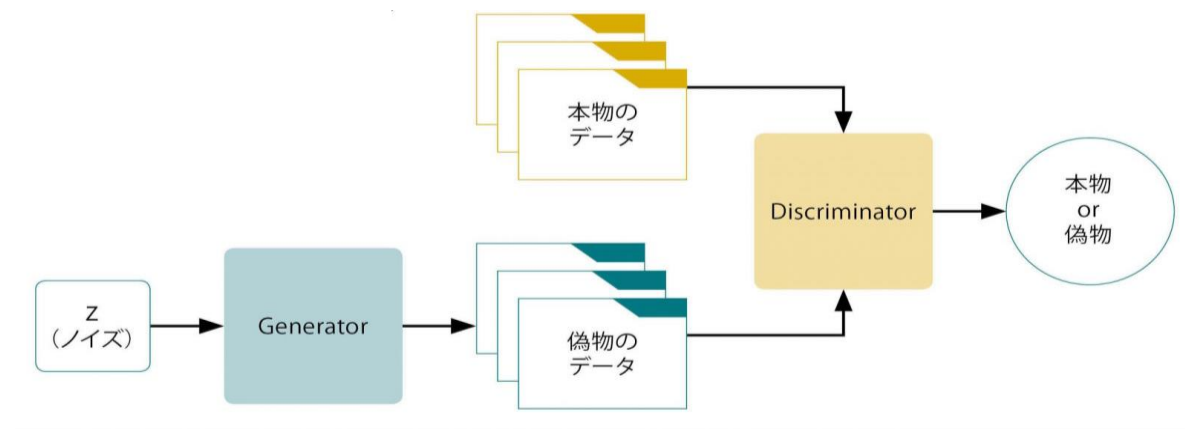
画像識別の体験

- <https://teachablemachine.withgoogle.com/train/image>
- (画像以外) <https://teachablemachine.withgoogle.com/train>

自分で画像を学習させ、識別させるモデルを作成できるWebサイト。

Generative Adversarial Network (GAN)

- 敵対的生成ネットワーク：2つのニューラルネットワークを互いに競い合わせて学習を深める
- Discriminator（分別器）は本物か人工的な画像化を見破る訓練をする。
- 分別器が見破れなかった人工的な画像が作成される。



GANの応用

着色



Cycle GAN



シマウマ → ウマ



特徴量の変化



GANの使われ方 2

- This person does not exist

<https://thispersondoesnotexist.com/>

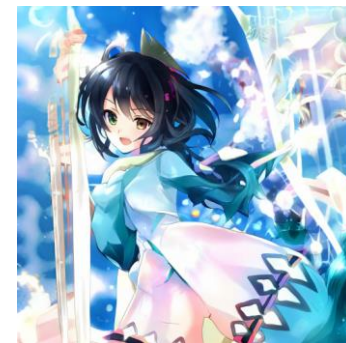
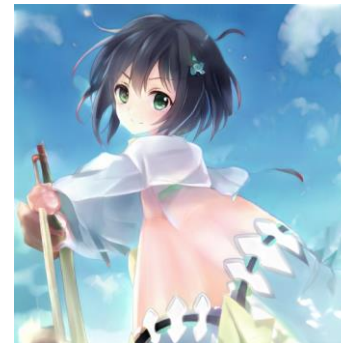


GANで作られた画像を見ることができる
ダミー企業の役員写真として使われた例もある.

Creativity=0.4

Creativity=1.0

GANの使われ方 3



This anime does not exist

- https://thisanimedoesnotexist.ai/index_jp.html
- どれだけ創造的にするか（手を加えるか）をcreativityで指定

This ~~~ does not existシリーズのまとめサイト

- <https://thisxdoesnotexist.com/>

物体識別とコンピュータ

コンピュータはいかに物体を即座に認識できるようになったのか

- https://www.ted.com/talks/joseph_redmon_how_computers_learn_to_recognize_objects_instantly?language=ja

レビューシートの提出

- 今日の授業に関するレビューシートを，manabaから提出すること．

後日不明点があれば，多胡まで．

7号館5階 第9実験室内 第9研究室

tago@net.it-chiba.ac.jp