

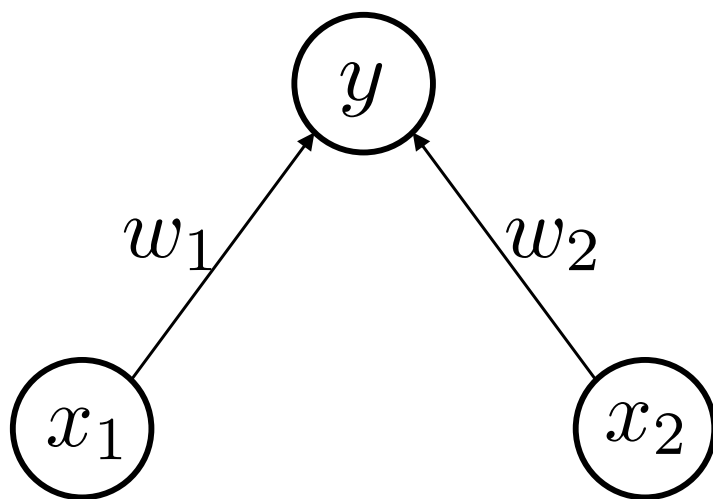
2022年度電気電子プログラミング及演習 総合課題

# ディープラーニングによる 手書き文字認識

黒橋 禎夫

京都大学工学部電気電子工学科

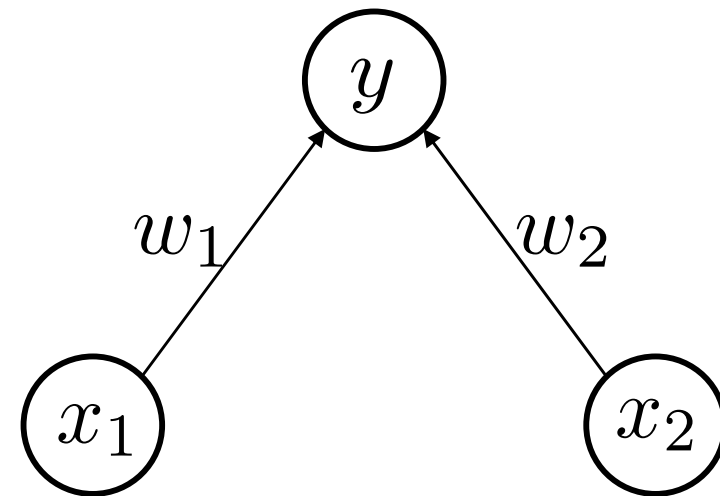
# 最も基本的なニューラルネットワーク



$$y = \begin{cases} 0 & (w_1x_1 + w_2x_2 \leq \theta) \\ 1 & (w_1x_1 + w_2x_2 > \theta) \end{cases}$$

# ANDゲート

$x_1$	$x_2$	$y$
0	0	0
1	0	0
0	1	0
1	1	1

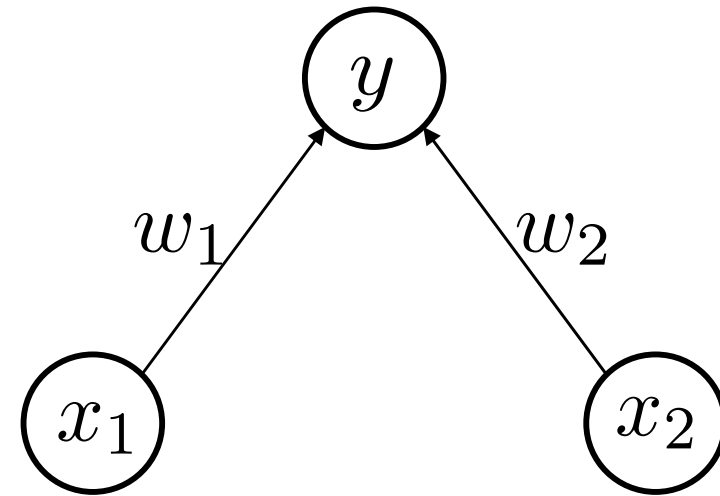


$$y = \begin{cases} 0 & (w_1x_1 + w_2x_2 \leq \theta) \\ 1 & (w_1x_1 + w_2x_2 > \theta) \end{cases}$$

$$y = \begin{cases} 0 & (0.5x_1 + 0.5x_2 \leq 0.7) \\ 1 & (0.5x_1 + 0.5x_2 > 0.7) \end{cases}$$

# ORゲート

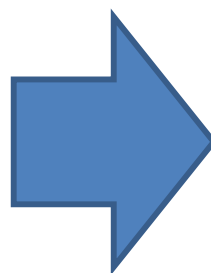
$x_1$	$x_2$	$y$
0	0	0
1	0	1
0	1	1
1	1	1



$$y = \begin{cases} 0 & (w_1x_1 + w_2x_2 \leq \theta) \\ 1 & (w_1x_1 + w_2x_2 > \theta) \end{cases}$$

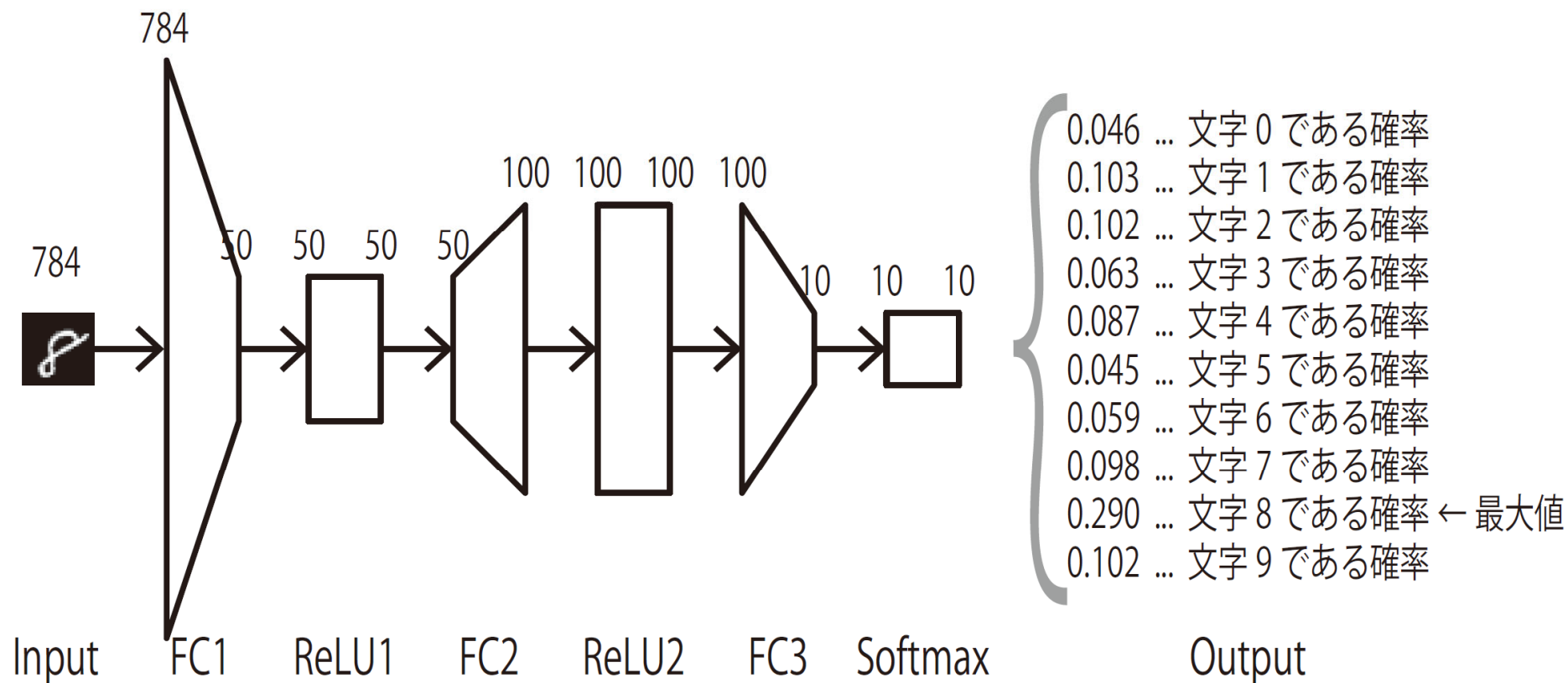
$$y = \begin{cases} 0 & (0.7x_1 + 0.7x_2 \leq 0.5) \\ 1 & (0.7x_1 + 0.7x_2 > 0.5) \end{cases}$$

# 手書き文字認識



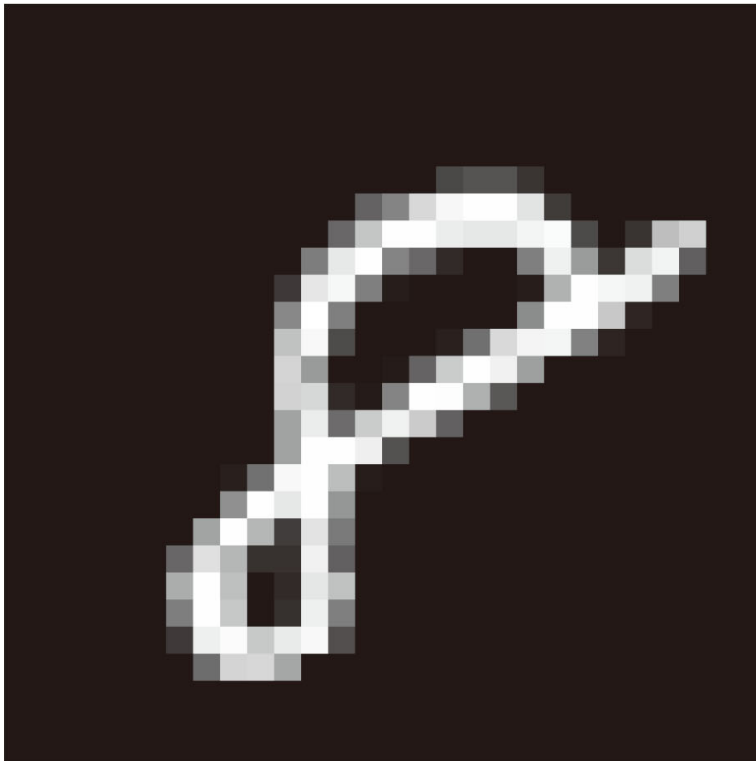
8

# NNによる手書き文字認識

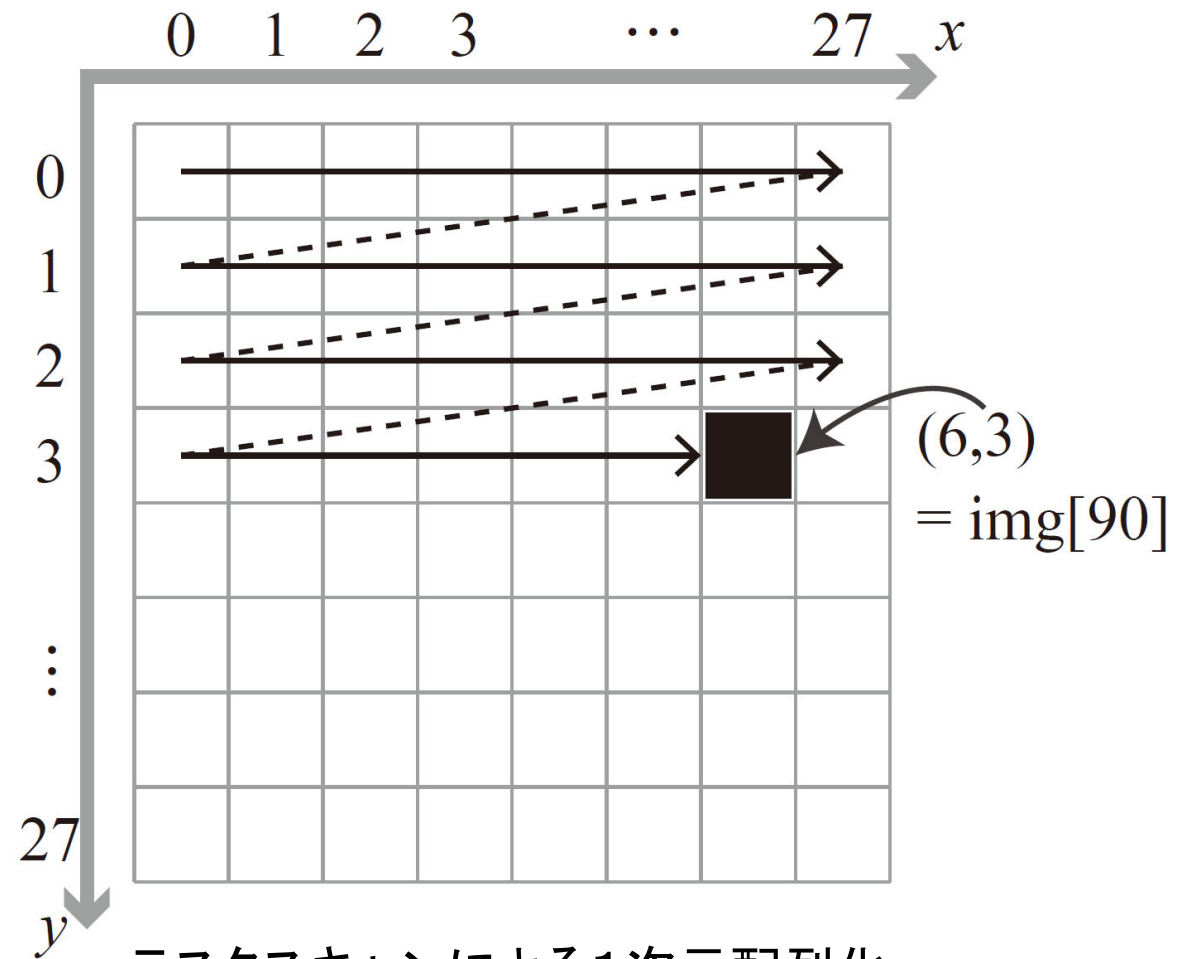


パラメータは**教師データ**(画像・数字ペアのセット)から**学習**！！

# 画像とは



画像＝輝度の2次元配列  
(例:縦28×横28画素)



ラスタスキャンによる1次元配列化  
(例:  $28 \times 28 = 784$ 要素の配列)

画像 = 多次元ベクトル(縦28×横28なら784次元)  
文字認識 = 多次元ベクトルから整数[0:9]への変換

# 手書き文字認識とは

入力画像ベクトル $x$ (784次元)を入力すると,

- 文字「0」である確率
- 文字「1」である確率
- ...
- 文字「9」である確率

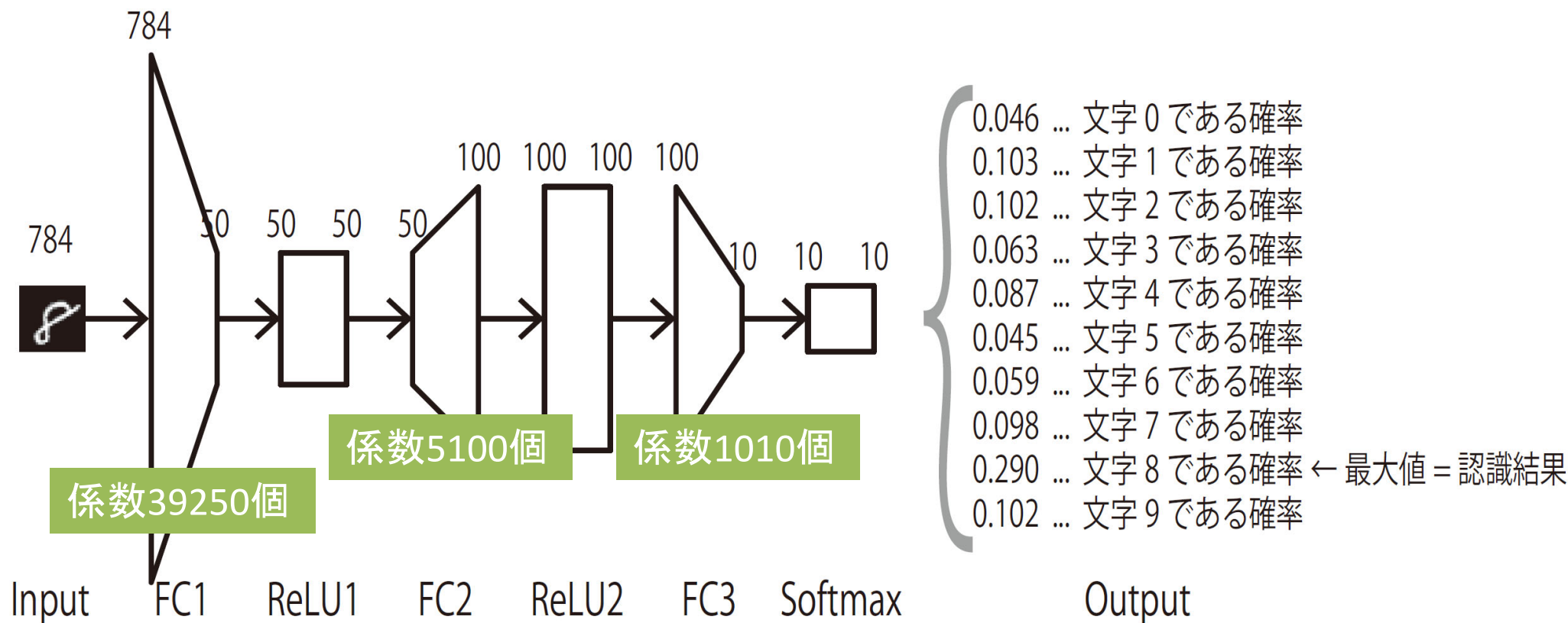
を10次元ベクトル $y$ として出力する関数

$$y = f(x)$$

をつくること.



# NNによる手書き文字認識



$$y = f(x; \theta)$$

$\theta$  : 係数ベクトル (46460次元)

FC  $y = Ax + b$

ReLU  $y = x > 0 ? x : 0$

Softmax  $y = \frac{e^x}{\sum e^x}$

パラメータは**教師データ**(画像・数字ペアのセット)から**学習**！！

# 教師あり学習

- 仮定：予め正解（教師データ）が付与された「手書き文字画像」が一定数与えられているとする。
  - 教師データ，訓練データ，と呼ばれる。
- 問題：訓練データに対して，常に正解を返すような関数 $f$ の係数 $\theta$ を決めることができるか？

⇒（確率的）勾配法

# 勾配法による学習

出力 $y$ は確率分布



理想は正解で確率1, ほかは確率0が出力されること  
つまり $(0, \dots, 0, 1, 0, \dots, 0)$ のようなベクトルとして確率分布を出力してほしい



確率分布間の距離＝交差エントロピーを最小化したい

$$E = - \sum_k t_k \log y_k$$

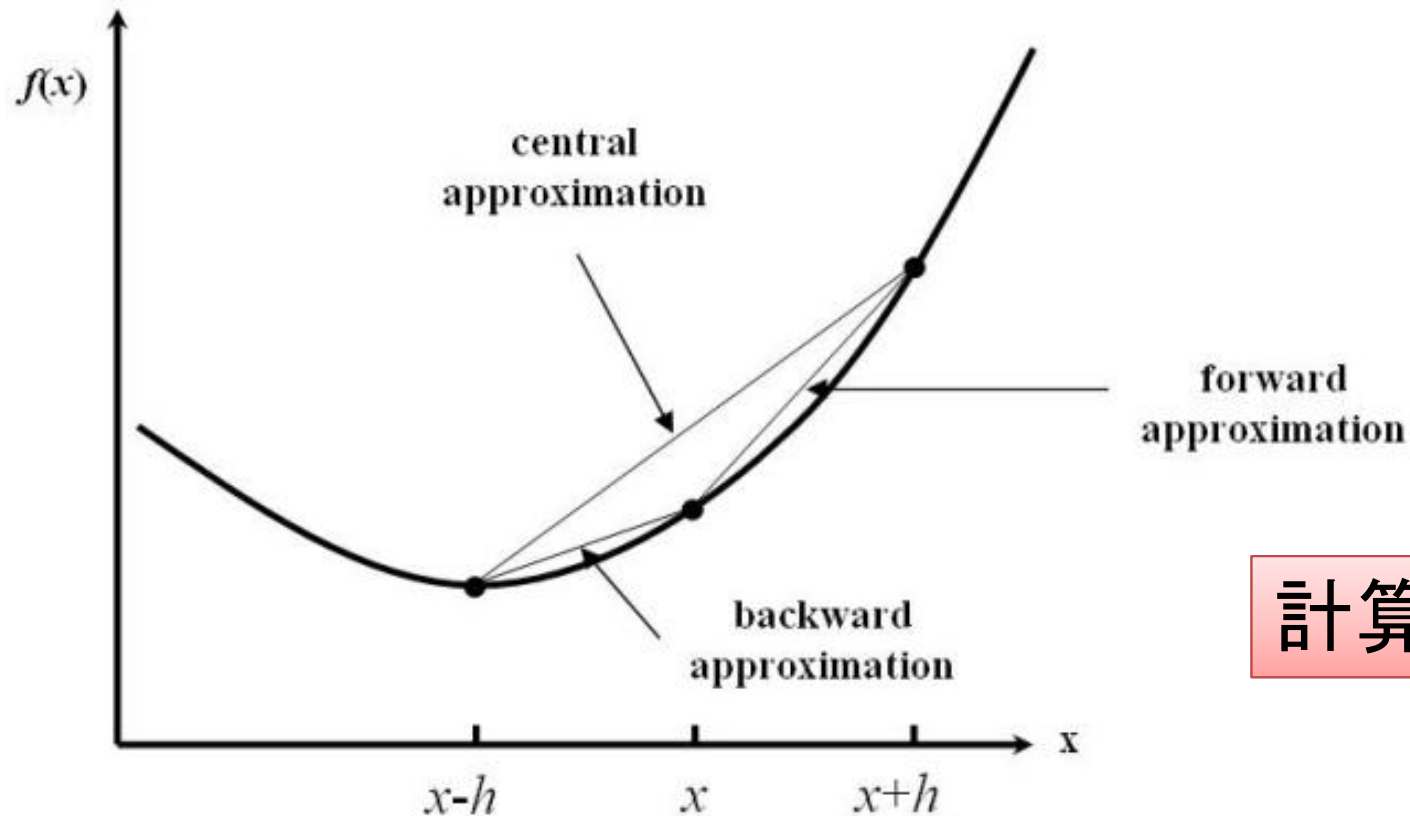


交差エントロピーを $\theta$ で微分して勾配を求め, その逆方向に $\theta$ を更新

元の $y = f(x; \theta)$ を $\theta$ の関数とみなして、勾配  $\frac{\partial E(\theta)}{\partial \theta}$  を計算する

# 計算機による微分：数値微分

$$\frac{\partial E}{\partial \theta_k} = \frac{E(\theta + \epsilon_k) - E(\theta - \epsilon_k)}{2\epsilon} \quad \text{により計算}$$



計算量が多すぎる

# 計算機による微分：解析微分

$\frac{\partial E}{\partial \theta_k}$ を導出して、そのままCで実装



NNを変更するごとに、導出しなおすのが面倒



微分の連鎖律を使えば、導出しなおす必要がない

$$E(\boldsymbol{\theta}) = E(f(\boldsymbol{\theta}), \mathbf{t}) = E\left(f_{\text{SM}}\left(f_{\text{FC3}}\left(f_{\text{ReLU2}}\left(f_{\text{FC2}}\left(f_{\text{ReLU1}}\left(f_{\text{FC1}}(\boldsymbol{\theta})\right)\right)\right)\right)\right)\right)$$

$$\frac{\partial E}{\partial \theta_k} = \frac{\partial E f_{\text{SM}}}{\partial f_{\text{FC3}}} \frac{\partial f_{\text{FC3}}}{\partial f_{\text{ReLU2}}} \frac{\partial f_{\text{ReLU2}}}{\partial f_{\text{FC2}}} \frac{\partial f_{\text{FC2}}}{\partial f_{\text{ReLU1}}} \frac{\partial f_{\text{ReLU1}}}{\partial f_{\text{FC1}}} \frac{\partial f_{\text{FC1}}}{\partial \theta_k}$$

誤差逆伝搬法

# 確率的勾配法

ベクトル勾配 =  $\frac{\partial E}{\partial \theta}$  を計算する際、訓練データ  $x$  をパラメータとして使用する。そのときの  $x$  は訓練データ1個？ 100個？ 10000個？



全データを使うと計算量が多い。  
しかし極少数のデータから学習すると学習が偏る。



全  $N$  個から  $n$  ( $n \ll N$ ) 個ランダムに選んで勾配を計算

# 総合課題：書き文字認識

## 2段階プロセス

- 学習：訓練データからNNの係数を決める（逆伝搬）
- 推論：新たに与えられたデータに対して認識（順伝搬）

## 必要な要素

- 訓練データ      ⇒ PandAで配布
- 補助関数        ⇒ PandAで配布
- NNの構成        ⇒ 配布資料で指定

## 最終課題の必須要素

- 学習：訓練データを読み込んで、NNの係数を決める
- 推論：画像を読み込んで、認識結果を表示