

# 機械学習 第9回 深層学習

立命館大学 情報理工学部

福森 隆寛

Beyond Borders

# 講義スケジュール

## □ 担当教員 1 : 福森 (第1回～第15回)

1	機械学習とは、機械学習の分類
2	機械学習の基本的な手順
3	識別 ( 1 )
4	識別 ( 2 )
5	識別 ( 3 )
6	回帰
7	サポートベクトルマシン
8	ニューラルネットワーク

9	深層学習
10	アンサンブル学習
11	モデル推定
12	パターンマイニング
13	系列データの識別
14	半教師あり学習
15	強化学習

## □ 担当教員 2 : 叶昕辰先生 (第16回の講義を担当)

# 今回の講義内容

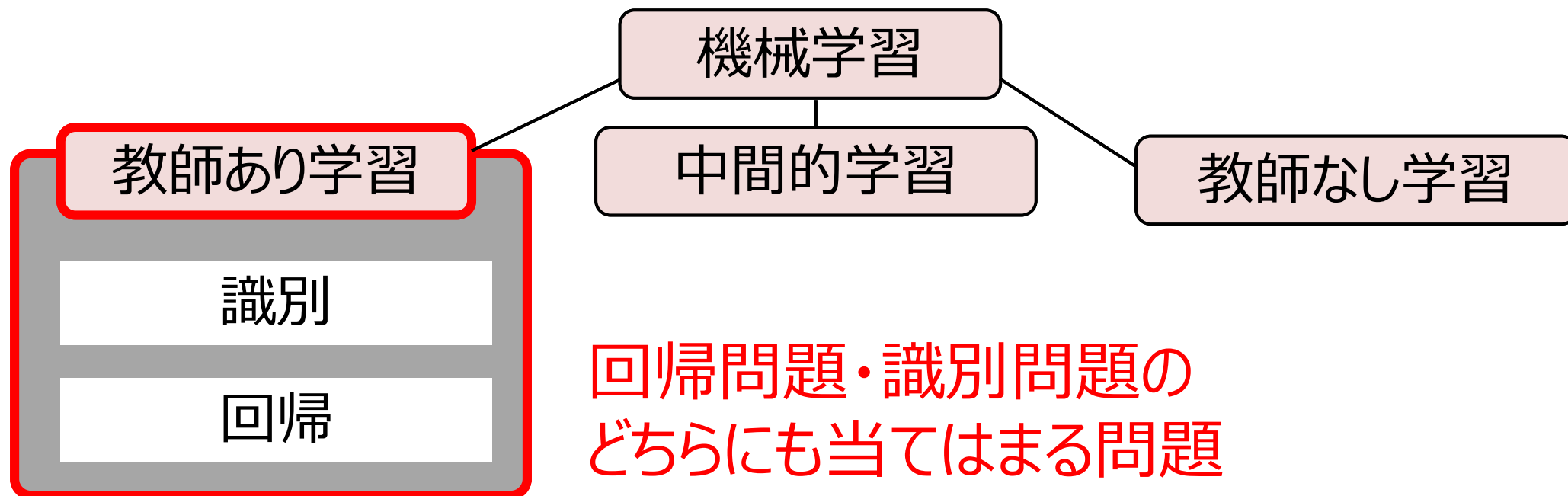
- 取り扱う問題の定義
- 深層学習
- DNNのモデル
- 多階層ニューラルネットワーク
- 畳み込み<sup>たたこ</sup>ニューラルネットワーク
  - 学習
  - オートエンコーダ Auto-encoders
  - 学習の工夫（ドロップアウト） Errors in learning (dropouts)
- リカレントニューラルネットワーク Recurrent Neural Networks
- 演習問題

# 取り扱う問題の定義：教師あり問題

- 特徴ベクトルを入力して、それをクラス分けする識別器、または、それに対応する数値を出力する関数を作る

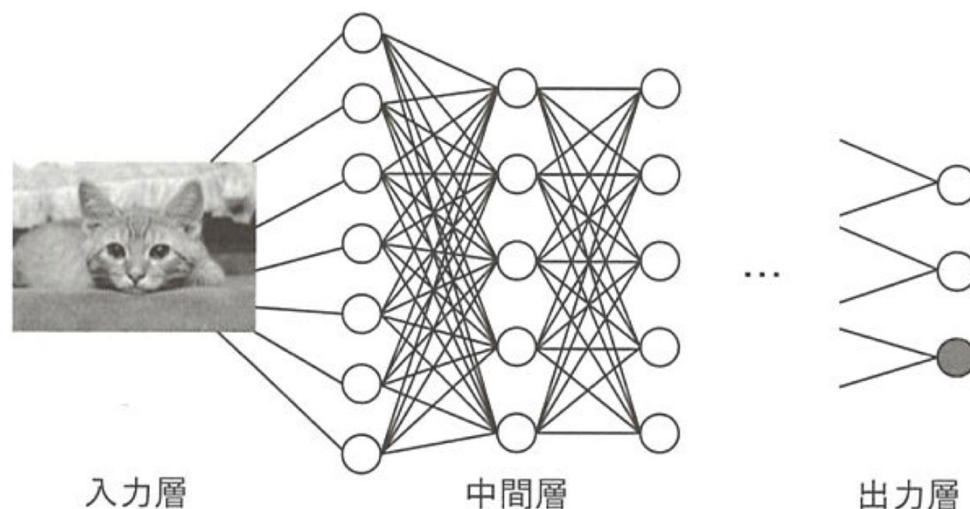
※ 教師あり学習の問題での学習データは、以下のペアで構成される

入力データの特徴ベクトル  $\leftarrow \{\underline{x_i}, \underline{y_i}\}, \quad i = 1, 2, \dots, \underline{N} \longrightarrow$  学習データの総数  
正解情報



# 深層学習

- 深層学習を単純に定義すると
  - 特徴抽出前の信号を入力とする多階層ニューラルネットワークの学習
- Deep Neural Network (DNN)
  - 深層学習に用いるニューラルネットワーク



深層学習を行う多階層ニューラルネットワーク (DNN)

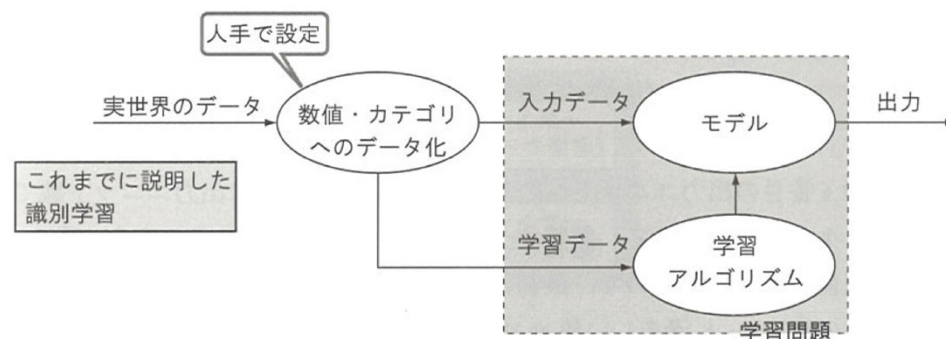
# 深層学習

## □ 表現学習 (representation learning)

- 特徴抽出前の<sup>なま</sup>生データに近い信号から、その内容を表現する特徴を学習する
- 深層学習は、表現学習と呼ばれることもある

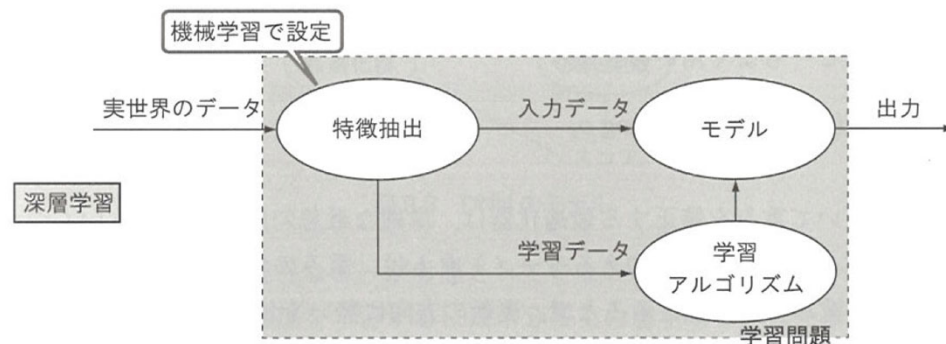
これまでの識別問題

識別に役立つ特徴が、ある程度わかっていて  
それらの特徴からなるベクトルを用いて学習



深層学習

どのような特徴を抽出するのかも、データから学習する



これまでの学習と深層学習の違い

# DNNのモデル

## □ ニューラルネットワークにおける識別問題の学習を抽象化して考える

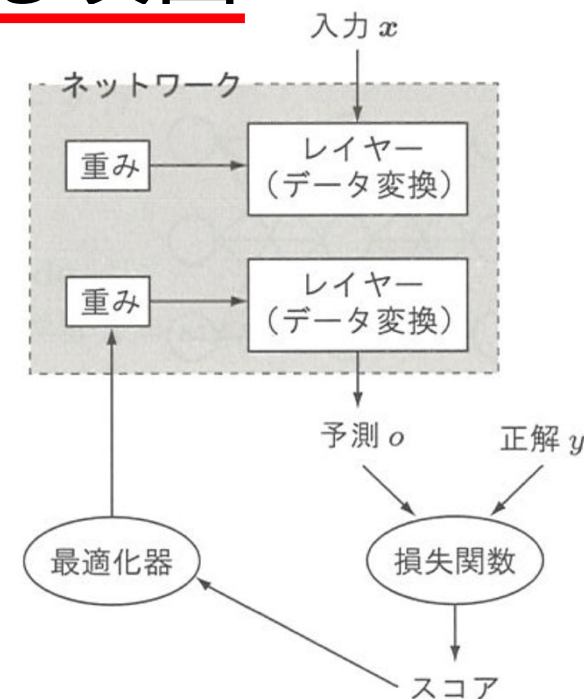
ちゅうしょうか

## □ DNNのモデルを構築する上で設定する項目

こうもく

1. 多階層ニューラルネットワークの構成
2. 損失関数
3. 重みを修正する最適化器

そんしつ



DNNのモデル

# DNNのモデル

## □ 多階層ニューラルネットワークの構成

### ■ 層数

### ■ 各層のユニット数

### ■ 活性化関数

- シグモイド関数、ReLU、softmax関数など

- softmax関数

– ネットワークの出力を確率値に変換する目的で出力層に使用される

$$f(h_i) = \frac{\exp(h_i)}{\sum_{j=1}^c \exp(h_j)}$$

$h_i$  :  $i$ 番目の出力ユニットへの入力重み付き和、 $c$  : 出力ユニット数



# DNNのモデル

## □ 損失関数

- 出力 $o_i$ と望ましい値 $y_i$ の差を評価する関数
- 「誤差の二乗和」や「クロスエントロピー」がよく用いられる
  - 誤差の二乗和

$$E(\mathbf{w}) \equiv \frac{1}{2} \sum_{x_i \in D} (y_i - o_i)^2$$

- クロスエントロピー

$$E(\mathbf{w}) \equiv - \sum_{x_i \in D} y_i \log o_i$$

# DNNのモデル

## □ 重みを修正する最適化器（様々な種類がある）

### ■ 最急勾配法うんどうりょうに加えるモーメンタム（運動量）

1. 1時刻前の勾配ベクトルいきおを重み付きよくせいで加える方法
  - 更新の方向に勢いをつけることで収束を早め、振動を抑制する効果
2. 準ニュートン法（2次微分を更新式に加えて修正を早める）

### ■ adam (adaptive moment estimation)

- モーメンタムに指数平滑移動平均しすうへいかつを用いる方法
- 分散に関するモーメントも用いると、まれに観測される特徴軸に対して大きく更新する効果がある

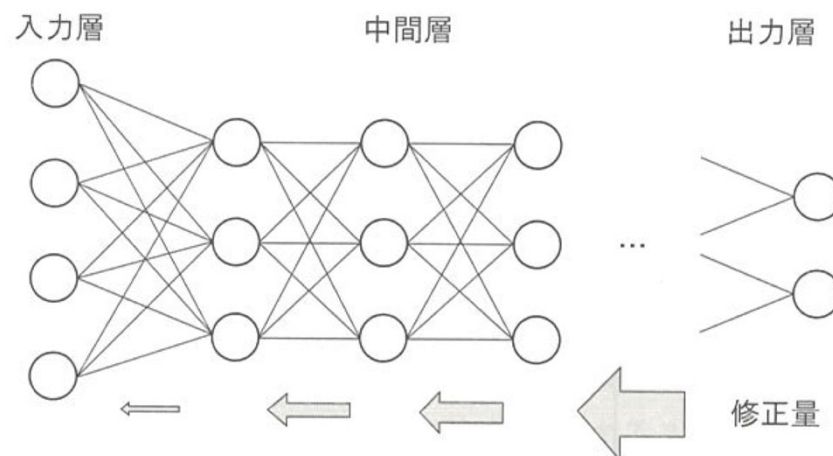
### ■ 階層が深い場合

- AdaGrad : 勾配の二乗を使って学習係数を自動調整
- RMSProp : 勾配の二乗の指数移動平均を用いる方法

# 多階層ニューラルネットワーク

## □ 多階層ニューラルネットワークの学習

- ニューラルネットワークの学習手法である誤差逆伝播法は多階層構造にも適用可能
- 誤差逆伝播法の問題点として、入力層に向かうにつれて、修正量が少なくなり、多階層では入力側の重みがほとんど動かない
  - この問題を解決するのが事前学習法



多階層の誤差逆伝播法（修正量は入力に戻るほど小さくなる）

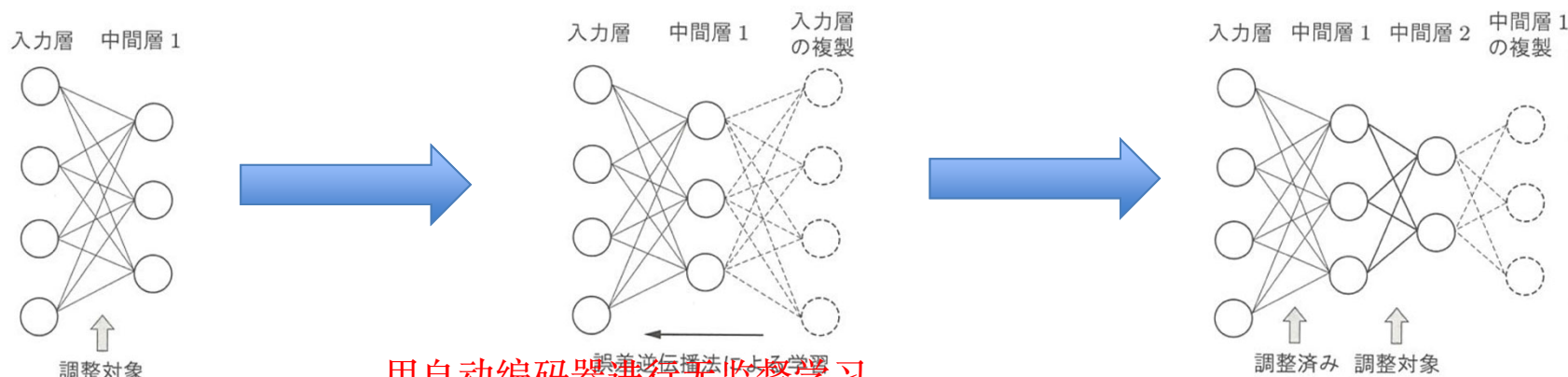
# 多階層ニューラルネットワーク

## □ 事前学習法 (pre-training)

- 何らかの方法で重みの初期パラメータを適切なものに事前調整する

- 入力 $x$ の情報をなるべく失わないように うしな 无监督学习是从输入层开始逐层进行的，这样输入的信息就会尽可能不丢失。  
入力層側から1層ずつ順に教師なし学習を行う

- 入力層から上位に上がるにつれ、ノードの数が減るので、  
特徴となる情報を保持しつつ、階層を重ねながら、抽象度の  
高い情報表現を獲得することが深層学習のアイデア



事前調整対象の重み

用自动编码器进行无监督学习  
オートエンコーダによる教師なし学習

1階層上の事前調整

# 多階層ニューラルネットワーク

## □ 事前学習のアルゴリズム

入力：正解付学習データ  $D$

出力： $L$ 層DNN

$F \leftarrow D$

**for**  $l = 1$  to  $L - 1$  **do**

*/\*  $h$ ：誤差関数、 $\Lambda$ ：ニューラルネットワーク、 $W$ ：ネットワークの重み \*/*

単層特徴抽出器の学習： $\hat{\Lambda} = \operatorname{argmin} h(\Lambda; F)$

単層特徴抽出器からDNNのパラメータ抽出： $W_l \leftarrow \widehat{W}(\hat{\Lambda})$

単層特徴抽出器の適応： $F \leftarrow \Phi(F)$

**end for**

softmax法などで最上位層を含めた誤差逆伝播法による学習

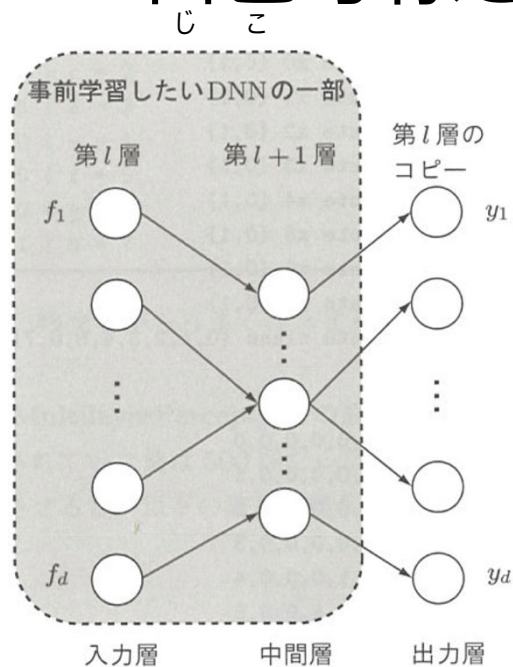
$h(\Lambda; F)$ ：ニューラルネットワーク $\Lambda$ のデータ $F$ における誤差

$\Phi(F)$ ：データ $F$ に対して単層特徴抽出器を適用した結果

# 多階層ニューラルネットワーク

## □ オートエンコーダ

- 事前学習の一つで、入力 $x$ の情報をなるべく失わず、より少ないノードへの写像を学習する手段
- 3階層のフィードフォワード型のニューラルネットワークで自己写像を学習



$d$ 次元の入力 $f$ と、同じく $d$ 次元の出力 $y$ の距離の全学習データに対する総和が最小になるようにニューラルネットワークの重みを調整

$$\min \sum_{i=1}^N \text{Dist}(f_i, y_i)$$

通常、距離Distは、ユークリッド距離が使われる

出力層の値

- ・入力が2値の場合 → シグモイド関数を活性化関数とした出力
- ・入力が連続値の場合 → 中間層の出力の重み付き和を出力

オートエンコーダの概念

# 演習問題9-1（10分間）

- 以下の0から7までの数を自己写像する  
ニューラルネットワーク（オートエンコーダ）を考える
  - 入力層・出力層：8次元ベクトル
    - 表現する数に対応する特徴のみが1、その他は0
      - 例：「3」という数に対する入力ベクトルは「0,0,1,0,0,0,0,0」
  - 中間層のノード数：3
- このネットワークを学習した結果、  
中間層では、入力がどのように表現されているか  
（入力からどのような情報が獲得されたか） 考えよ

# 演習問題9-1（10分間） 解答例

□ このネットワークを学習した結果、中間層では、入力がどのように表現されているか考えよ

## □ 解答例

■ 「2進<sup>しん</sup>エンコーディングの概念」が獲得されている（かも）

■ 0から7の数は、3桁<sup>けた</sup>以内の2進数で表現できる

- 8ノードの入力情報が、より少ない3ノードの中間情報で表現
- 「3」を例に考えると

– 入力層：「0,0,1,0,0,0,0,0」、中間層（2進数）：「0,1,1」

オートエンコーダは、このようにして得られた中間層の値を入力として、1階層上にずらして、同様の表現学習を行う。この手順を積み重ねると、入力に近い側では単純な特徴が、階層が上がっていくにつれて複雑な特徴が学習される



# 多階層ニューラルネットワーク

## □ 多階層ニューラルネットワークの過学習

- ニューラルネットワークの階層を深くするとパラメータも増えるので、過学習の問題が深刻となる
- ドロップアウトで過学習を起こりにくくさせる

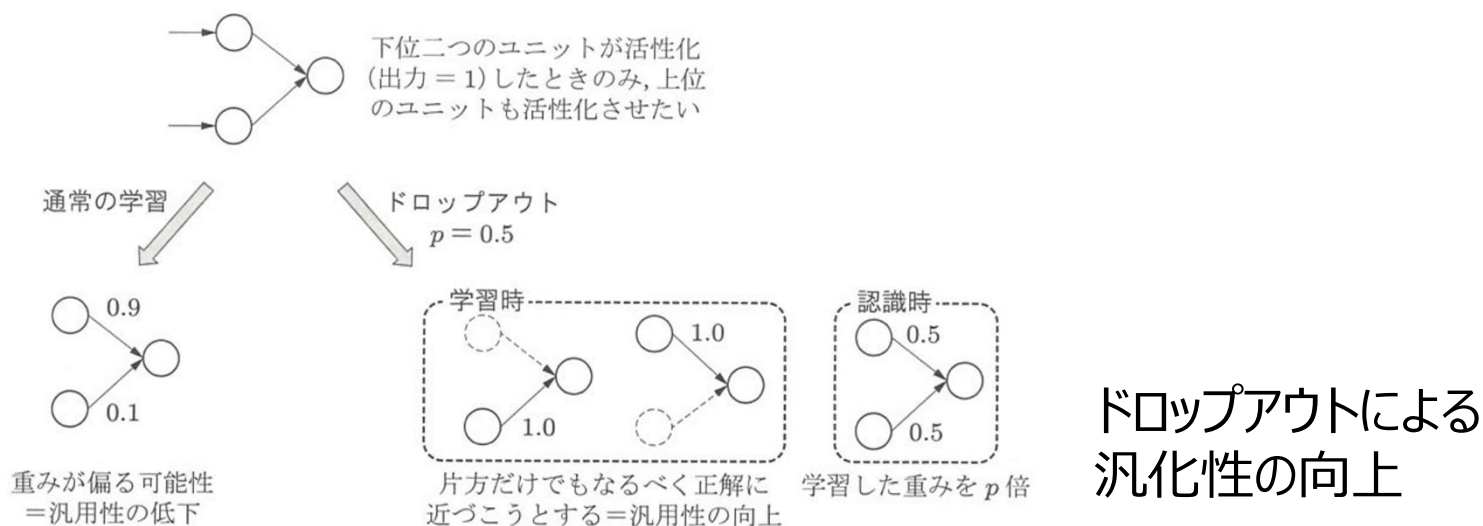
## □ ドロップアウト 通过在每层中以固定的速度随机禁用单元进行学习。

- 各層のユニットを一定の割合でランダムに無効化して学習
  - 対象とするミニバッチのデータが変わるごとに無効化するユニットを選びなおして学習を繰り返す
- 学習時の自由度を意図的に下げることで、正解を出力するための結合重みを特定の値に限定されやすくなる

# 多階層ニューラルネットワーク

## □ ドロップアウトの手順

1. 各層のユニットを割合 $p$ でランダムに無効化
2. 1. のネットワークに対して、ミニバッチのデータを用いて誤差逆伝播法による学習を行う
  - 対象とするミニバッチのデータが変わるごとに無効化するユニットを再選択して学習を繰り返す
3. 識別するときは、重みを $p$ 倍して計算



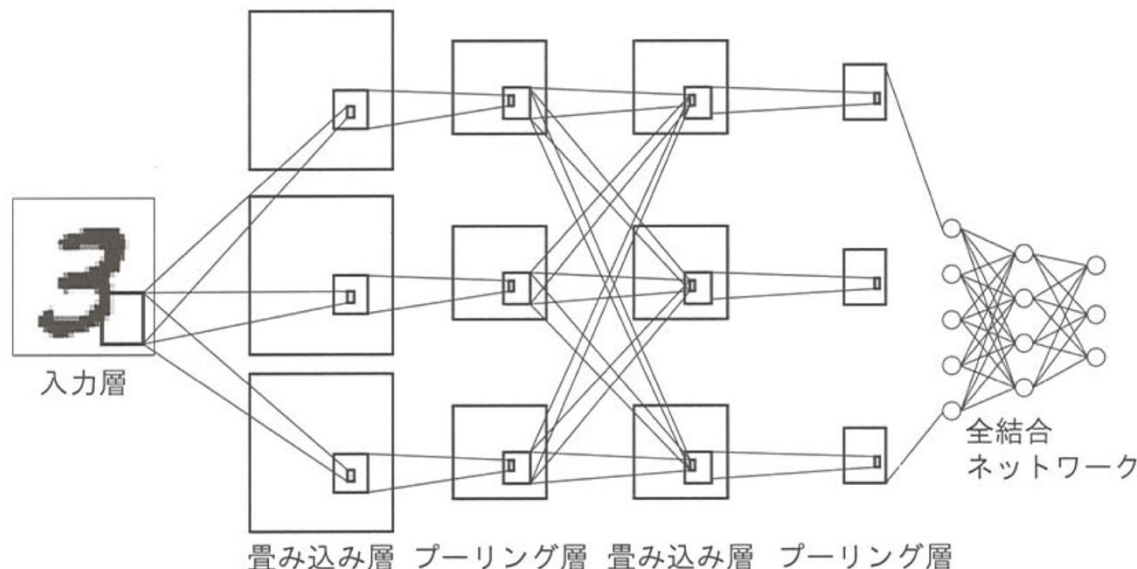
# 畳み込みニューラルネットワーク

## □ 畳み込みニューラルネットワーク

■ convolutional neural network; CNN

■ 畳み込み層と、プーリング層を交互に配置し、最後のプーリング層の出力を受ける全結合のニューラルネットワークを最終出力側に配置

卷积层和池化层交替排列，接收最后一个池化层输出的所有连接的神经网络被安排在最终输出端。



※ 学習するパラメータ

- 畳み込み層のフィルタ係数
- 全結合層の結合重み

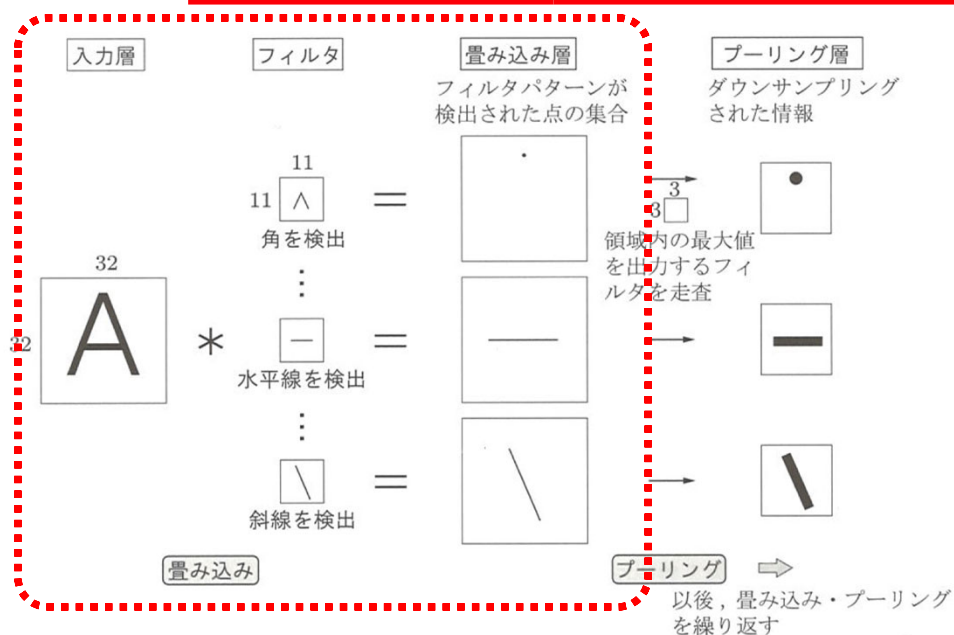
需要学习的参数

- 卷积层的过滤系数
- 所有组合层的组合权重

# 畳み込みニューラルネットワーク

## □ 畳み込み層 卷积层

- 画像のフィルタリングに相当する処理を行う層 一个执行相当于图像过滤的处理的层。
- フィルタリング（畳み込み）の結果で得られた値を シグモイド関数やReLUなどの活性化関数に代入
- 学習データから各フィルタのフィルタ係数を学習する

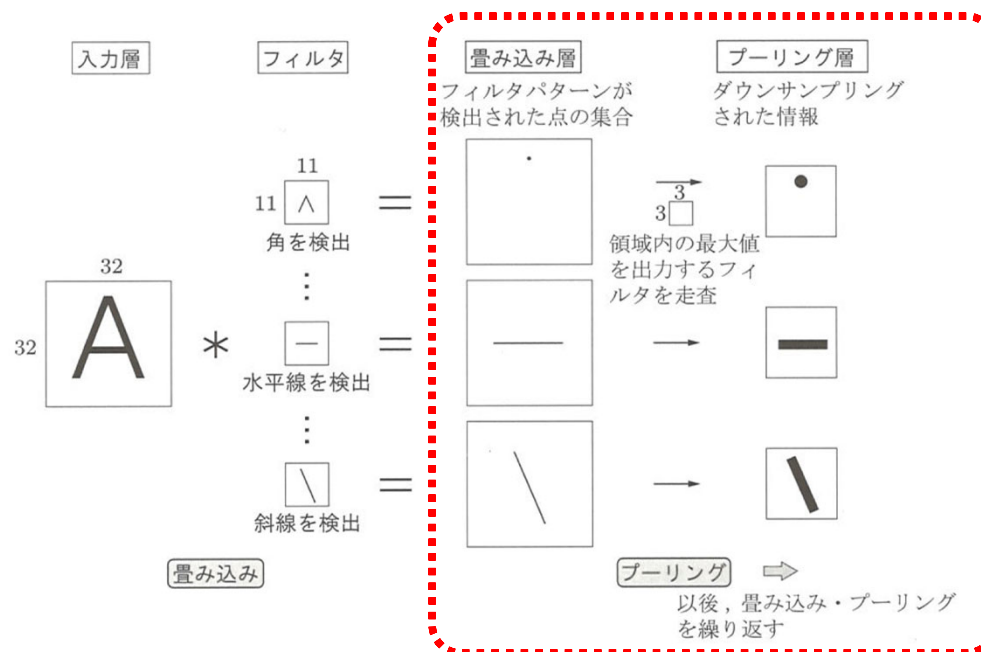


畳み込みニューラルネットワークの演算

# 畳み込みニューラルネットワーク

## □ プーリング層

- 注目領域内の平均、あるいは最大値を出力する層
- 畳み込み層よりも少ないユニットで構成
- 領域内のパターンの位置変化<sup>きゅうしゅう</sup>を吸収



畳み込みニューラルネットワークの演算

# リカレントニューラルネットワーク

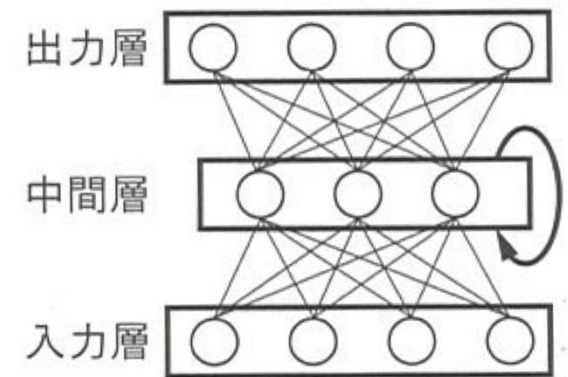
## □ リカレントニューラルネットワーク

■ recurrent neural network; RNN

■ 中間層の出力が時間遅れで自分自身に戻ってくる構造をもつネットワーク

■ 時系列信号や自然言語などの系列パターンに対して有効

- 単純に各時点の入力からだけでは出力を決めることが難しく、それまでの入力系列が役立つ場合に有効
- 例えば
  - 動画像を入力して異常を検知する
  - ベクトル化された単語系列を入力して品詞列を出力する

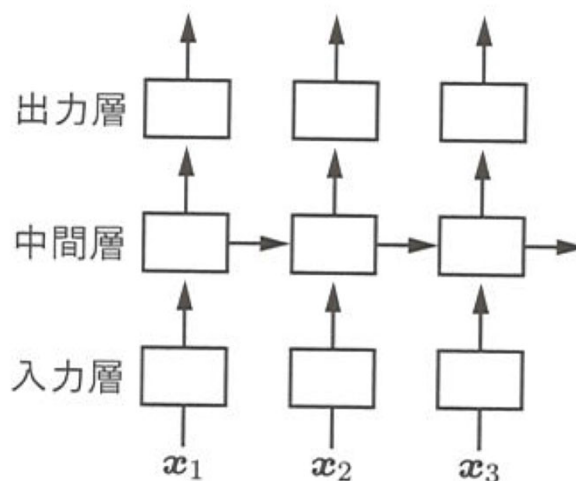


リカレントニューラルネットワーク

# リカレントニューラルネットワーク

## □ リカレントニューラルネットワーク（つづき）

- RNNの入力：特徴ベクトルの系列  $x_1, x_2, \dots, x_T$
- 中間層は、入力層からの情報に加えて  
一つ前の中間層の活性化状態を入力とする  
じこく
- 時刻 $t$ における出力は、時刻 $t - 1$ 以前の  
すべての入力をもとに計算されるので、深い構造をもつ
- 結合重みの学習にて、単純な誤差逆伝播では勾配消失  
問題が発生



きかんろ 帰還路を時間方向に展開した  
てんかい リカレントニューラルネットワーク

# リカレントニューラルネットワーク

## □ 長・短期記憶 (Long Short-Term Memory; LSTM)

### ■ 中間層のユニットを記憶構造にもつ 特殊なメモリユニット (LSTMセル) に置き換える方法

- 従来のRNNの機能に加えて、情報の流れを制御する  
3つのゲート（入力ゲート、出力ゲート、忘却ゲート）を持つ
- ゲートの開閉は、入力情報をもとに判定される
  - 入力ゲート：現在の入力が自分に関係あるものか？
  - 出力ゲート：自分は出力に影響を与えるべきか？
  - 忘却ゲート：これまでの情報を忘れて良いのか？

### ■ 学習時には、誤差もゲートで制御されるので、 必要な誤差のみが伝播することで、勾配消失問題を回避



# 演習問題9-2（10分間）

- 畳み込みニューラルネットワークを用いたモデルを構築することを考える。このネットワークを構築するために、事前に決めておくべきパラメータは何か考えよ
  - 1つ目は「活性化関数を何にするか？」
  - それ以外に事前に決めておかないといけないものは何？