1

# データ解析演習(2)

## 分類モデルの構築

高間　康史
下川原　英理，　何 宜欣
システムデザイン学部・情報通信システムコース

2

## 講義内容

- 分類モデルの構築
  - 前回のおさらい
  - 手法の分類
- 最近傍法
  - 1-NN
  - K-NN
- ナイーブベイズ
  - ベイズの定理
- サポートベクターマシン（SVM）
  - 線形分離可能な場合
  - カーネルトリック
- ニューラルネットワーク
  - 基本アルゴリズム
  - 深層学習

# 分類モデル構築とは

3

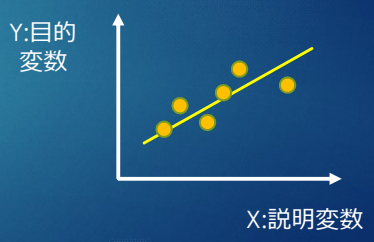
- 未知のデータを，既知のクラスに分類
  - グループ分けという点では，クラスタリングと同様
- クラスタリングとの違い
  - **教師あり学習**：訓練データから学習
    - グループ（クラス）の個数・意味が既知
    - 訓練データ（所属クラスが既知）から**分類モデル構築**
  - **教師なし学習**：訓練データなし　←　クラスタリング
    - グループ（クラスター）の個数・意味が未知
    - グループの意味は後から考える

# データの話

4

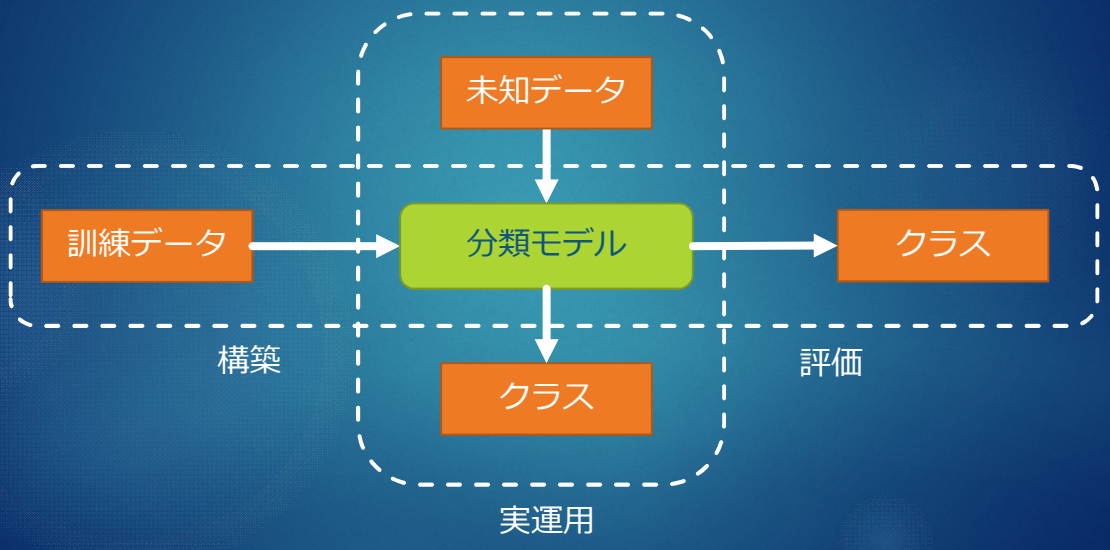
- 目的変数・従属変数　　※変数→属性と呼ぶ場合もあり
  - 予測・分類したい対象：クラス
  - （通常）分類器毎に１つ
- 説明変数，独立変数
  - 予測・分類に利用する情報
  - （通常）複数
- 訓練データ＝説明変数**＋目的変数**
- 未知データ＝説明変数

例）回帰分析
Y=aX+b



# 分類学習の流れ






# テストデータ



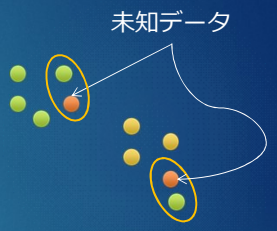
- （通常）訓練データと別に用意する必要
  - 未知のデータ（＝訓練とは異なるデータ）を正しく分類できる事が重要
- 既知データを訓練データ，テストデータに分けて利用
  - 既知データ：クラスがわかっているデータ
  - テストサンプル法：既知データをランダムに分割
    - 訓練データを多く　⇒　分類精度向上（十分な学習）
    - テストデータを多く　⇒　信頼性向上（十分な評価）

    } トレードオフ
  - 交差検証法（CV: cross-validation）
    - 効率的な既知データの活用
    - 通常利用すべき

# 最近傍法
# Nearest Neighbor



- メモリベース：既知データをそのまま使う
- 一番近い既知データと同じクラスに分類
- 問題：ノイズの影響をうけやすい

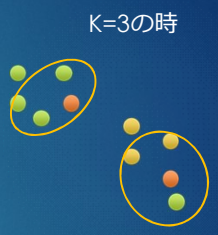



- 解決策: 多数決の導入（ノイズに対するロバスト性向上）
  - K-NN
- WekaではIBk

# K-NN

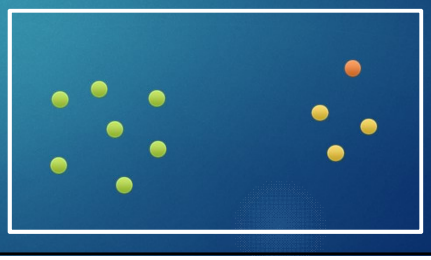


- 近傍のデータK個で多数決
- ノイズに対するロバスト性




- 注意点
  - Kの設定に注意：むやみに大きくしても×
  - データ数，クラスの分布を確認すべき

K=10
⇒全て●に分類



# ナイーブベイズ
## Naive Bayes



- 各クラスへの所属確率を推定
- 確率モデルを訓練データから学習
- ベイズの定理
  - 事前確率：$P(X)$
  - 事後確率：$P(X|Y)$
  - 尤度：$P(Y|X)$
  - 証拠：$P(Y)$

$$P(X|Y) = \frac{P(X)P(Y|X)}{P(Y)}$$

# 例題



- ある町で車によるひき逃げ事故があった.
- その町で走っている車の15%が青色である.
- 事故の目撃者は，ひき逃げをした車は青色だと証言した.
- その時間帯・場所において，目撃者は80%正しく識別できることがわかっている.
- 事故を起こした車が青色であった確率はいくらか

# 解答

11

- その町で走っている車の15%が青色である．
  - X: 車が青色，1-X: 他の色　⇒　P(X) = 0.15, P(1-X) = 0.85
- 事故の目撃者は，ひき逃げをした車は青色だと証言した．
  - Y: 青色と証言
- その時間帯・場所において，目撃者は80%正しく識別できることがわかっている．
  - P(Y | X) = 0.8 … 実際に青で，正しく証言
  - P(Y | 1-X) = 0.2 … 実際は青ではないのに見間違えて証言
  - P(Y) = P(Y | X)P(X) + P(Y | 1-X)P(1-X)
- 事故を起こした車が青色であった確率はいくらか
  - P(X | Y) = P(X)P(Y | X)/P(Y) = 0.15 x 0.8 / (0.8 x 0.15 + 0.2 x 0.85) = 0.41
  - 証言なし（事前確率）：0.15 ⇒ 証言あり（事後確率）：0.41 に上昇

# 分類モデル構築 例題

12

- X: クラス（目的変数）．1変数
- Y: 属性・特徴（説明変数）．通常複数（ベクトル）
  - Y=(曇，高，休日)，X=○
- 確率（事前確率・尤度・証拠）：訓練データから計算
- Y=(晴，高，休日)の時の売上は？
- 問題：同じYが訓練データに存在しない
- 解決策：独立性の仮定

$$P(Y|X) = \prod_i P(Y_i|X)$$

| 天気 | 気温 | 休/平日 | 売上 |
|---|---|---|---|
| 曇 | 高 | 休日 | ○ |
| 晴 | 低 | 休日 | ○ |
| 雨 | 低 | 休日 | ○ |
| 曇 | 高 | 平日 | ○ |
| 晴 | 高 | 平日 | ○ |
| 曇 | 低 | 休日 | ○ |
| 雨 | 高 | 平日 | × |
| 曇 | 低 | 平日 | × |
| 晴 | 低 | 平日 | × |
| 曇 | 低 | 平日 | × |

# 解答

13

- 天気
  - 売上○: 9 … **晴: 3**, 曇: 4, 雨: 2
  - 売上×: 7 … **晴: 2**, 曇: 3, 雨: 2
- 気温
  - 売上○: 8 … **高: 4**, 低: 4
  - 売上×: 6… **高: 2**, 低: 4
- 休／平日
  - 売上○: 8 … **休日: 5**, 平日: 3
  - 売上×: 6 … **休日: 1**, 平日: 5
- ラプラス推定：各頻度に1追加
  ←ゼロ頻度問題への対処

$P(\text{晴}, \text{高}, \text{休日} \mid \circ)$

$= P(\text{晴} \mid \circ)P(\text{高} \mid \circ)P(\text{休} \mid \circ) = \frac{3}{9}\frac{4}{8}\frac{5}{8} = \frac{5}{48}$

$P(\text{晴}, \text{高}, \text{休日} \mid \times) = \frac{2}{7}\frac{2}{6}\frac{1}{6} = \frac{1}{63}$

$P(\circ \mid \text{晴}, \text{高}, \text{休日}) \propto P(\circ)P(\text{晴}, \text{高}, \text{休日} \mid \circ) = \frac{5}{48}\frac{7}{12} = 0.061$

$\vee$

$P(\times \mid \text{晴}, \text{高}, \text{休日}) \propto P(\times)P(\text{晴}, \text{高}, \text{休日} \mid \times) = \frac{1}{63}\frac{5}{12} = 0.0066$

# 離散データと連続データ

14

- 説明変数が離散（Nominal），連続（Numerical）どちらでも利用可能

Class
Attribute　yes (0.63)　no (0.38)　$P(x)$

==========

temperature

| | yes | no |
|---|---|---|
| hot | 3.0 | 3.0 |
| mild | 5.0 | 3.0 |
| cool | 4.0 | 2.0 |
| [total] | 12.0 | 8.0 |

$P(y_i|x)$

離散型
確率分布

humidity

| | yes | no |
|---|---|---|
| high | 4.0 | 5.0 |
| normal | 7.0 | 2.0 |
| [total] | 11.0 | 7.0 |

※ラプラス推定

Class
Attribute　yes (0.63)　no (0.38)

==========

temperature

| | yes | no |
|---|---|---|
| mean | 72.9697 | 74.8364 |
| std. dev. | 5.2304 | 7.384 |
| weight sum | 9 | 5 |
| precision | 1.9091 | 1.9091 |

連続型
確率分布
（正規分布）

humidity

| | yes | no |
|---|---|---|
| mean | 78.8395 | 86.1111 |
| std. dev. | 9.8023 | 9.2424 |
| weight sum | 9 | 5 |
| precision | 3.4444 | 3.4444 |

# サポートベクターマシン
# SVM: Support Vector Machine

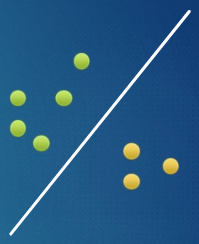
15

- 近年主流．多くの場合で高性能
- クラスを分離する境界（平面）を求める
- 訓練データの偏りに頑健
  境界付近のデータのみをモデル構築に利用
- カーネルトリック：線形分離不能な場合にも有効
- WekaではSMO, libSVM

# 線形分離とは

16

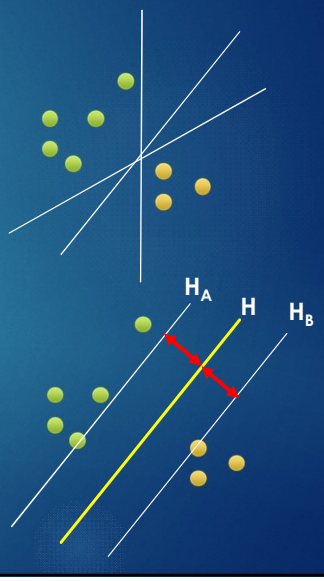
- 各クラスのデータを分離する**平面**の存在
  - 2次元の場合：直線
  - 4次元以上の場合：超平面
  - $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + w_0 = 0$
- 線形判別関数
  - 平面のどちら側にあるかでクラスを判別
- 線形分離不可能な場合もあり
- とりあえずは線形分離可能な場合について説明



# 分離超平面の学習

17

- 分離超平面をどこに引くか？
  - 可能性は無数に存在
- SVMの考え方
  - マージン最大化
  - マージン：両クラスの間の空間
  - マージンの真ん中に分離超平面を引く




- $H_A$, $H_B$：クラスA, Bで境界に一番近いデータ（サポートベクター）を通る平面
- H // $H_A$ // $H_B$
- D(H, $H_A$) = D(H, $H_B$)
- $H_A$, $H_B$間の距離が最大となるH

# 学習結果の例

18

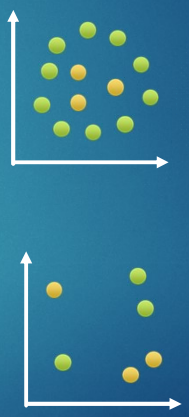
- WekaのSMO利用
  - 属性の正規化なし
- 入力：weather.numeric.arff
  - outlook: sunny / overcast / rainy
  - Temperature, humidity: numeric
  - Windy: true / false
- (sunny, 80, 90, true)を入力
  - 0.8342+0.065*80+0.1003*90-0.3138-13.7308 =1.0166
- (overcast, 83,86,false)を入力
  - -0.9772+0.065*83+0.1003*86-13.7308 =-0.6872
- 別クラスに分類

```
SMO

Kernel used:
  Linear Kernel: K(x,y) = <x,y>

Classifier for classes: yes, no

BinarySMO

Machine linear: showing attribute weights, ...

        0.8342 * outlook=sunny
+      -0.9772 * outlook=overcast
+       0.143  * outlook=rainy
+       0.065  * temperature
+       0.1003 * humidity
+      -0.3138 * windy
-      13.7308
```

# 線形分離不可能な場合

19

- 分離超平面が存在しない
- 非線形な境界
- 通常のSVMでは対応不能
- 解決策
  - ソフトマージンSVM
    - 識別誤りを許容
  - カーネルトリック
    - 高次元空間に写像



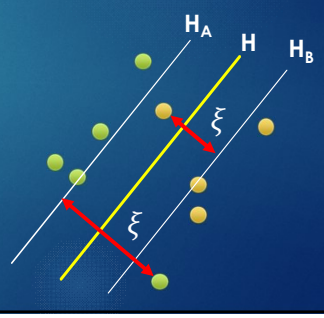
XOR

| x1 | x2 | class |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# ソフトマージンSVM

20

- 通常のSVM：制約付き最小化問題
  - Minimize $\|\boldsymbol{w}\|$ … 目的関数：Hの法線ベクトルの大きさを最小化
    - マージン最大化に対応
  - Subject to $\forall(\boldsymbol{x}^i, y^i), y^i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}^i + b) \geq +1$
    … 制約：全訓練データを正しく分類
- ソフトマージンSVM
  - Minimize $\|\boldsymbol{w}\|^2 + \boldsymbol{c}\sum_{x^i}\boldsymbol{\xi^i}$ … $\xi^i$：$x^i$の制約違反量
  - Subject to $\forall(\boldsymbol{x}^i, y^i), y^i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}^i + b) \geq +1 - \boldsymbol{\xi^i}$
    … 制約違反を許容→目的関数のペナルティ



# カーネルトリック

21

- 元の空間：線形分離不可能　→　高次元空間：線形分類可能
- 高次元空間で線形識別関数を求める
- カーネル関数：$K(x, x') = \phi(x)^{\mathrm{T}}\phi(x')$
  - $\phi(x)$: 高次元空間への非線形写像
  - 識別関数$g(\phi(\boldsymbol{x})) = \boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}) + w_0 = \sum_{\boldsymbol{x}^i} \alpha_i y^i \, \phi(\boldsymbol{x})^{\mathrm{T}}\phi(\boldsymbol{x}^i) + w_0$
    $= \sum_{\boldsymbol{x}^i} \alpha_i y^i \, K(\boldsymbol{x}, \boldsymbol{x}^i) + w_0$ … 個々のデータの非線形変換は不要！
- カーネル関数の例
  - 多項式カーネル：$K(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}' + l)^p$
  - RBFカーネル：$K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\sigma\|\boldsymbol{x} - \boldsymbol{x}'\|^2)$
    （ガウシアンカーネル）

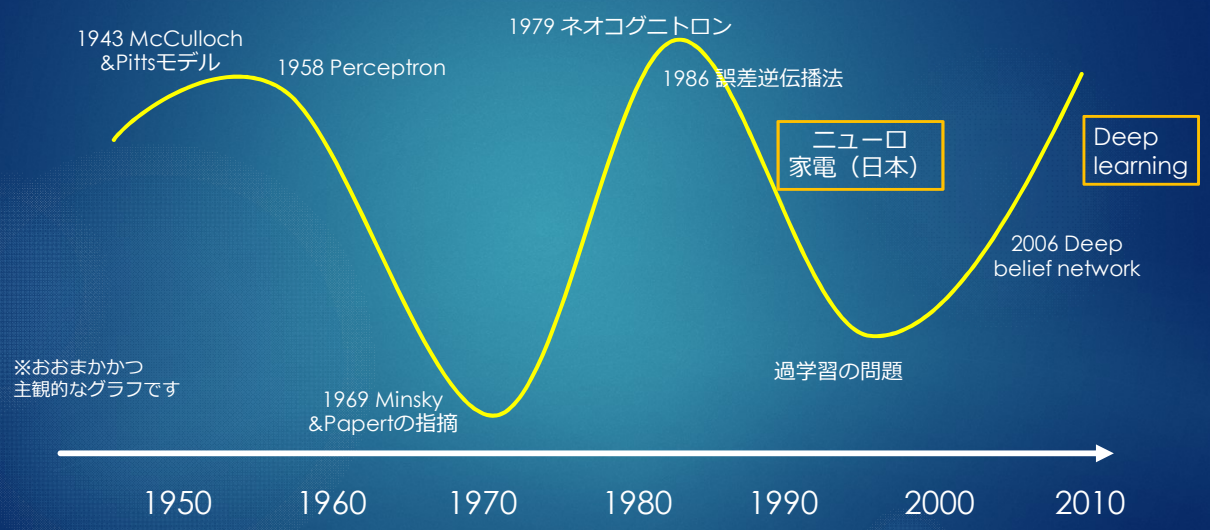
# ニューラルネットワーク
## Neural Network

22

- 脳神経系をモデルにした情報処理システム
- 線形分離不能な場合でも機能
- 学習結果がブラックボックス　⇔　決定木
- Weka: MultilayerPerceptionで利用可能

# ニューラルネットワークの歴史

23




# ニューロンのモデル

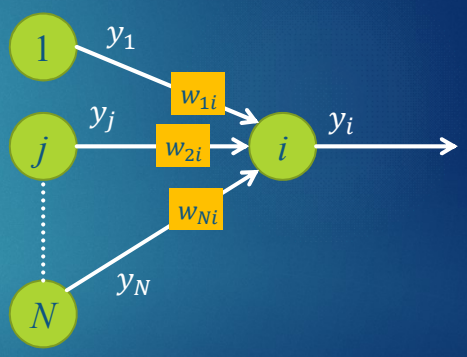
24

- McCulloch&Pitts のモデル (1943)
- 線形閾値関数
  - 出力1: 興奮
  - $w_{ij}$：シナプス結合
    - 正：興奮性結合
    - 負：抑制性結合

$$y_i = \begin{cases} 1, x_i \geq \beta_i \\ 0, x_i < \beta_i \end{cases}$$

$$x_i = \sum_j w_{ji} y_j$$

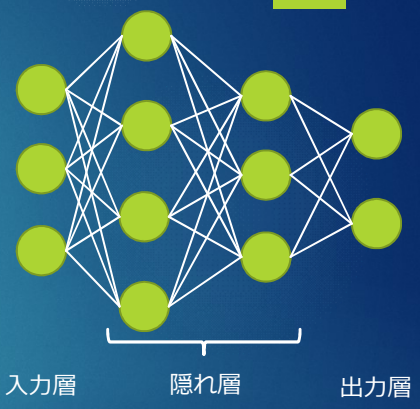
- パーセプトロン（Perceptron）(Rosenblatt, 1958)
  - シナプス結合の学習アルゴリズム
  - 線形分離可能な場合に学習可能（Minsky&Papert, 1969)



# 階層型ニューラルネットワーク

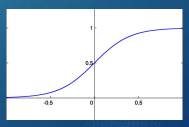
25

- 線形分離以外も学習可能
- 第1層：入力層．入力信号（説明変数）
- 最終層：出力層．目的変数（該当クラスが出力）
- 隠れ層（hidden layer）：中間ユニット，隠れユニット
- 同一層内のユニット間で情報伝達なし
  - 下位層から上位層へ
- ユニット出力：シグモイド関数
  - (0,1) の連続値．微分可能
  - $b_i$: $i$ 層のユニット数




$$y_j^i = \frac{1}{1 + exp\{-x_j^i - \beta_j^i\}}$$

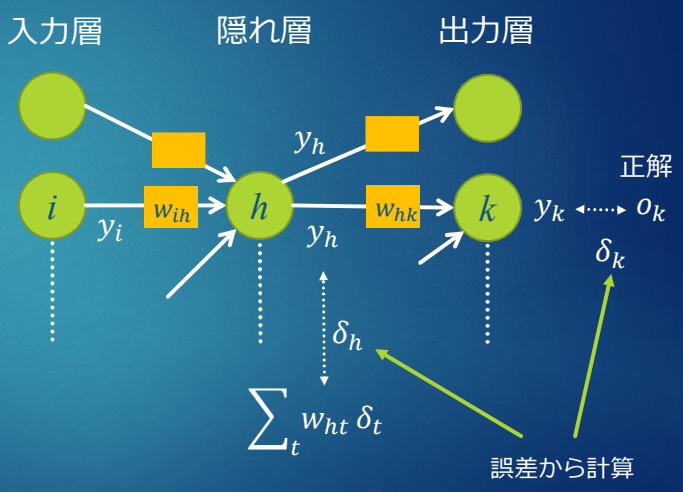
$$x_j^i = \sum_{k=1}^{b_{i-1}} w_{kj}^i y_k^{i-1}$$



# 誤差逆伝播法 (Backpropagation)

26

- 重み（シナプス結合）の更新
  - 出力層　$w_{hk} \leftarrow w_{hk} + \eta \delta_k y_h$
  - 隠れ層　$w_{ih} \leftarrow w_{ih} + \eta \delta_h y_i$
  - $\eta$：学習係数
  - 訓練データにより更新
- 二乗誤差の最小化
  - 出力層：正解（教師信号）と出力の誤差
  - 隠れ層：？？とユニット出力の誤差

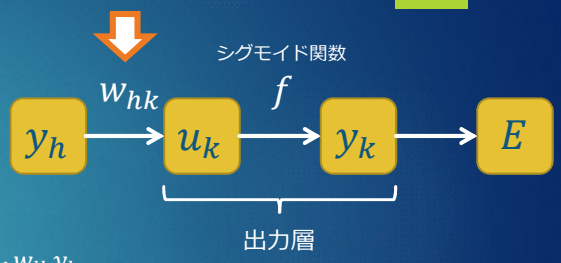


# 出力層の学習

27

- 訓練データ：$(x_1, x_2, \cdots, x_N, o_1, \cdots, o_M)$
- 誤差：
$$E = \frac{1}{2}\sum_{t}^{M}(o_t - y_t)^2$$
- 出力層ユニット $k$ の出力：$y_k = f(u_k), u_k = \sum_l w_{lk} y_l$
- 出力層ユニット $k$ と隠れ層 $h$ の重みの修正量：

$$\Delta w_{hk} = -\eta \frac{\partial E}{\partial w_{hk}} = -\eta \frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial u_k}\frac{\partial u_k}{\partial w_{hk}}$$
$$= \eta \underbrace{(o_k - y_k) y_k (1 - y_k)}_{\delta_k} y_h$$



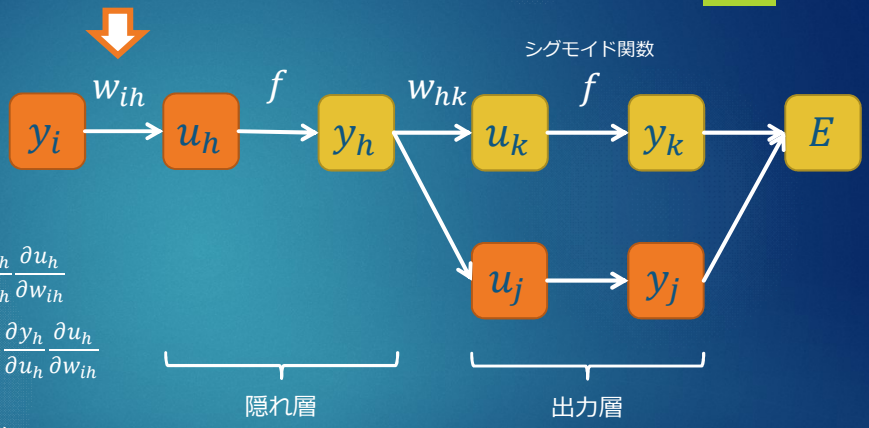

$$\frac{\partial E}{\partial y_k} = \frac{\partial}{\partial y_k}\left\{\frac{1}{2}(o_k - y_k)^2\right\} = -(o_k - y_k)$$

$$\frac{\partial y_k}{\partial u_k} = \frac{\partial}{\partial u_k}\left\{\frac{1}{1 + exp\{-u_k - \beta_k\}}\right\}$$
$$= \frac{exp\{-u_k - \beta_k\}}{(1 + exp\{-u_k - \beta_k\})^2} = y_k(1 - y_k)$$

$$\frac{\partial u_k}{\partial w_{hk}} = y_h$$

# 中間層の学習

28




$$\Delta w_{ih} = -\eta \frac{\partial E}{\partial w_{ih}} = -\eta \frac{\partial E}{\partial y_h}\frac{\partial y_h}{\partial u_h}\frac{\partial u_h}{\partial w_{ih}}$$
$$= -\eta \left\{\sum_{t=1}^{M}\frac{\partial E}{\partial y_t}\frac{\partial y_t}{\partial u_t}\frac{\partial u_t}{\partial y_h}\right\}\frac{\partial y_h}{\partial u_h}\frac{\partial u_h}{\partial w_{ih}}$$
$$= \eta \underbrace{\sum_{t=1}^{M}\delta_t w_{ht}\, y_h(1 - y_h)}_{\delta_h} y_i$$

$$\sum_{t=1}^{M}\frac{\partial E}{\partial y_t}\frac{\partial y_t}{\partial u_t}\frac{\partial u_t}{\partial y_h} = -\sum_{t=1}^{M}(o_t - y_t)y_t(1 - y_t)w_{ht} = -\sum_{t=1}^{M}\delta_t w_{ht}$$

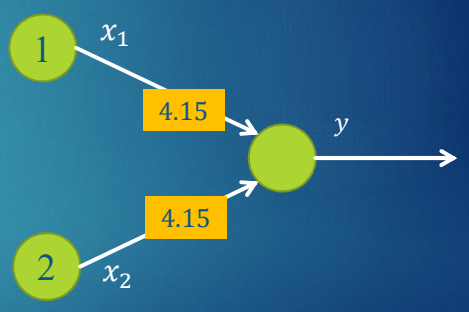
# 単層ニューラルネットワークの学習例

29

- 訓練データ：x1, x2, o (class)
  - 0, 0, 0
  - 0, 1, 1
  - 1, 0, 1
  - 1, 1, 1

OR

- 出力例: x1, x2, y
  - 0, 0, 0.142
  - 0, 1, 0.913
  - 1, 0, 0,913
  - 1, 1, 0.998

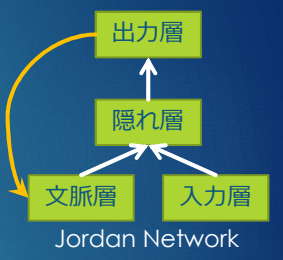



※WekaのMultilayerPerceptionでhiddenLayers=0にして実行

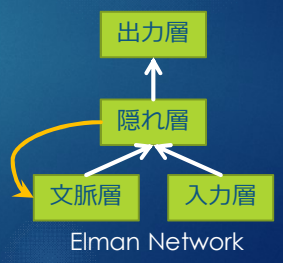
# リカレントニューラルネットワーク
## Recurrent Neural Network

30

- 基本は階層型
- 上位層から一部の情報が下位層に戻る
- 時系列データの処理
  - 音声認識，文法解析
  - 入力データ間に時間的関係
- Jordan Network: 出力層出力をフィードバック
- Elman Network: 中間層出力をフィードバック




Jordan Network

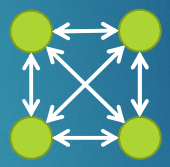


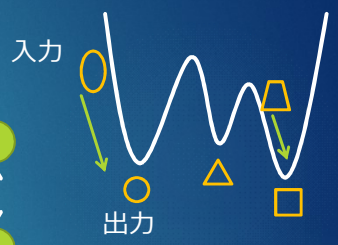

Elman Network

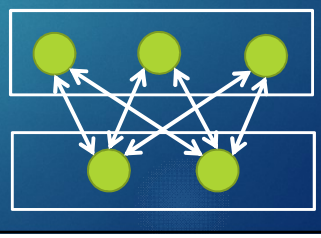
# ホップフィールドネットワーク
# Hopfield Network

31

- 相互結合型ネットワーク
  - 対称的な結合（$w_{ij} = w_{ji}$）
  - 状態更新によりエネルギー極小状態へ
  - 連想記憶が可能
- ボルツマンマシン
  - 確率的な状態更新：焼きなまし法
    - エネルギー最小値（global minimum）
  - 隠れ層を持つことが可能：入力に対応しないユニット
- 制限ボルツマンマシン
  - 同一層内ユニット間に結合なし
  - 深層学習の学習方法






32

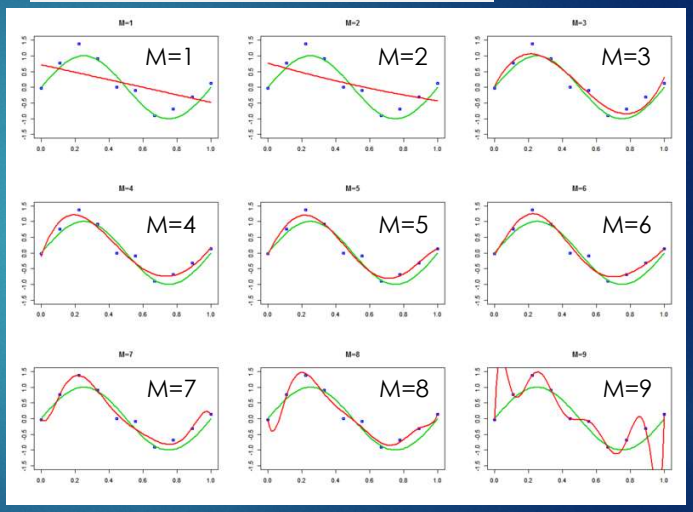
# ニューラルネットワークの関連話題

# 過学習 Overfitting

33

- 過学習とは
  - 訓練データへの過剰な適応
  - 未知データでの性能低下
  - E.g. 高次関数の採用
- オッカムの剃刀
  - 同程度の説明力なら単純なモデルを選ぶべき
- 決定木の場合：枝刈り
  - 末端の枝を省略

Sin(2πx) +ランダムノイズ　vs. 近似曲線




http://eizoo.hatenablog.com/entry/2013/05/02/000000

# SVMとの関係

34

- どちらも分離超平面の学習
- ニューラルネットワーク
  - 訓練データの誤差が小さくなるように決定
  - 訓練データ依存：過学習，局所最適解
- SVM
  - マージン最大化
  - 正例・負例の境界の真ん中に設定

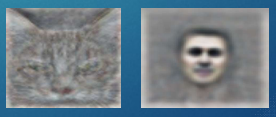
# 深層学習（Deep Learning）

35

- 認識（分類モデル構築）：画像，音声
  - 2012: 画像認識，音声認識，化合物の活性予測で最高精度達成（圧倒的）．ドメインの非専門家が達成
  - Microsoft: 音声認識を深層学習ベースに
- 大規模ニューラルネットワークからの**教師なし**学習
  - Google: Youtubeから200x200ピクセルの画像（動画から1枚ずつ）1000万枚をランダムに選択して学習
  - 大規模：9階層，10億パラメータ，1000台x 3日間
  - 人の顔，体，猫の顔に反応するニューロン

Q. V. Le et al., Building High-level Features Using Large Scale Unsupervised Learning, ICML12, 2012.

ニューロンが一番反応したテスト画像



ニューロンが一番反応する画像（数値計算）

# 深層学習の利点

36

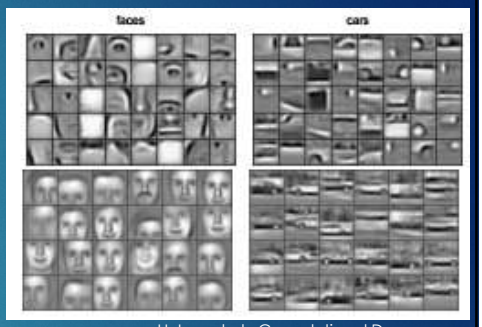
- 特徴抽出の問題を解決：職人芸から自動化
- 表現学習
  - 「観測データから本質的な情報を抽出し表現」する方法を学習
  - Cf. 主成分分析，データ圧縮，データ要約，クラスタリング
- ラベルなしデータから基本特徴量の有効な組合せを学習可能
  - クラスタリング：局所的な表現学習
  - 深層学習：大域的な表現学習
    - 階層的：単純な表現　→　総合的・抽象的な特徴に変換

# 深層学習の学習方法

37

- 多階層のネットワーク：どうやって学習？
- プレトレーニング：層毎に学習
  - 目的：入力層に近いほど学習が進まない問題の解決
  - 入力データの集合を再現できるように教師なし学習
    - オートエンコーダ　or　**制限ボルツマンマシン**
  - その後ファインチューニング：全体を教師あり学習（誤差逆伝播法）
- 畳みこみニューラルネットワーク
  - 歴史あり：初期は手書き文字認識
  - 脳の視覚野がモデル：受容野→単純型細胞→複雑型細胞（抽象的特徴）
  - 平行移動に対する不変性
  - プレトレーニング不要

（上）第2層→（下）第3層




H. Lee et al., Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations, ICML09, 2009.

# まとめ

38

- 今日学んだこと
  - 最近傍法
  - ナイーブベイズ
  - サポートベクターマシン
  - ニューラルネットワーク
- 来週
  - グループワーク・発表
- 今日の演習
  - データ作成と分類モデル構築