機械学習の基礎
-機械学習モデルの基礎-

スタートアップゼミ#5

2024/5/7

小川大智

目次

1. 機械学習モデルとは
2. 代表的なアルゴリズム
3. 深層ニューラルネットワーク
    1. 多層パーセプトロン
    2. CNN
    3. RNN
    4. Attention
4. モデルの学習・検証
5. 演習

1. 機械学習モデルと従来モデル

3回の人工知能ブーム

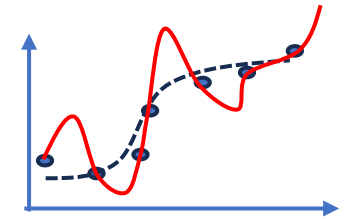
- 1次（探索と推論）→性能の限界
  - 1950年代後半-1960年代
  - 探索木による探索と記号論理による推論
- 2次（知識表現）→知識入力の限界
  - 1980年代
  - エキスパートシステム：専門家の推論プロセスをハードコーディング
- 3次（機械学習）
  - 2000年代-現在
  - データの学習による知識の獲得，深層学習

  - 従来のモデル：人によって意味付けされた記号論的なモデル
  - 機械学習：データの中の**パターン（帰納バイアス）**を自動で獲得

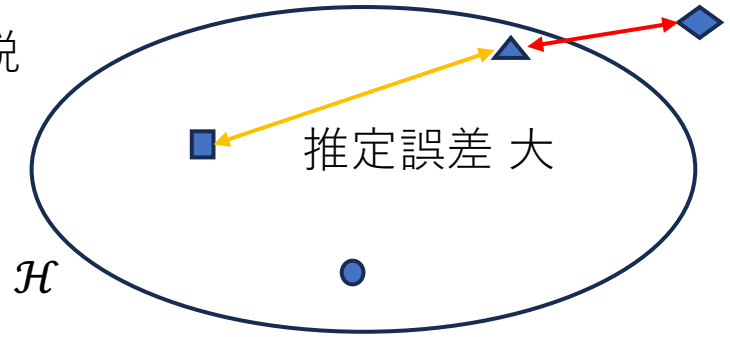
1. 統計的機械学習の基礎

データと機械学習

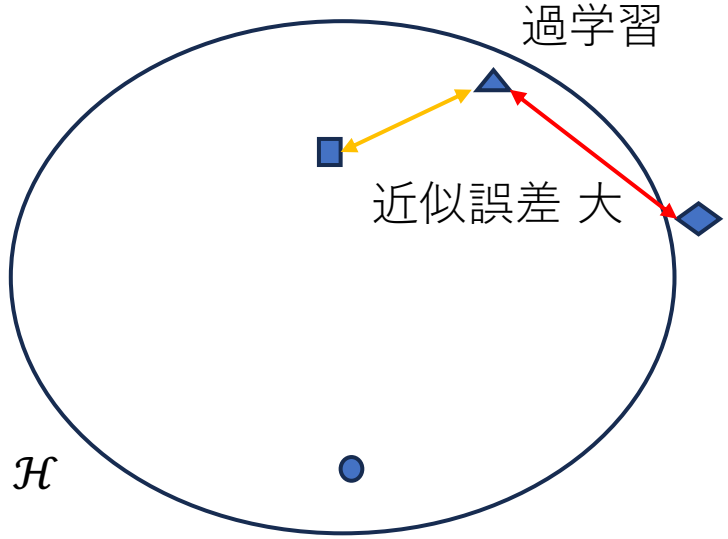
- データ：入力$\boldsymbol{x}$と出力yの組$D_n = \{(\boldsymbol{x}, y)\}_{i=1}^{n} \sim P(X, Y)$
- 入力空間$\mathcal{X}$，出力空間$\mathcal{Y}$
- 仮説集合$\mathcal{H} = \{h: \mathcal{X} \to \mathcal{Y}\}$
- 損失関数$\mathcal{L}(x, y; h) = l(h(x), y)$
- データD_nを精度よく近似するように，仮説hを訓練する．
  - 経験リスク$\hat{R}(h) = \frac{1}{n}\sum_i \mathcal{L}(x_i, y_i; h) = \frac{1}{n}\sum_i l(h(x_i), y_i)$



過学習



データが少ない場合



仮説集合が大きい場合

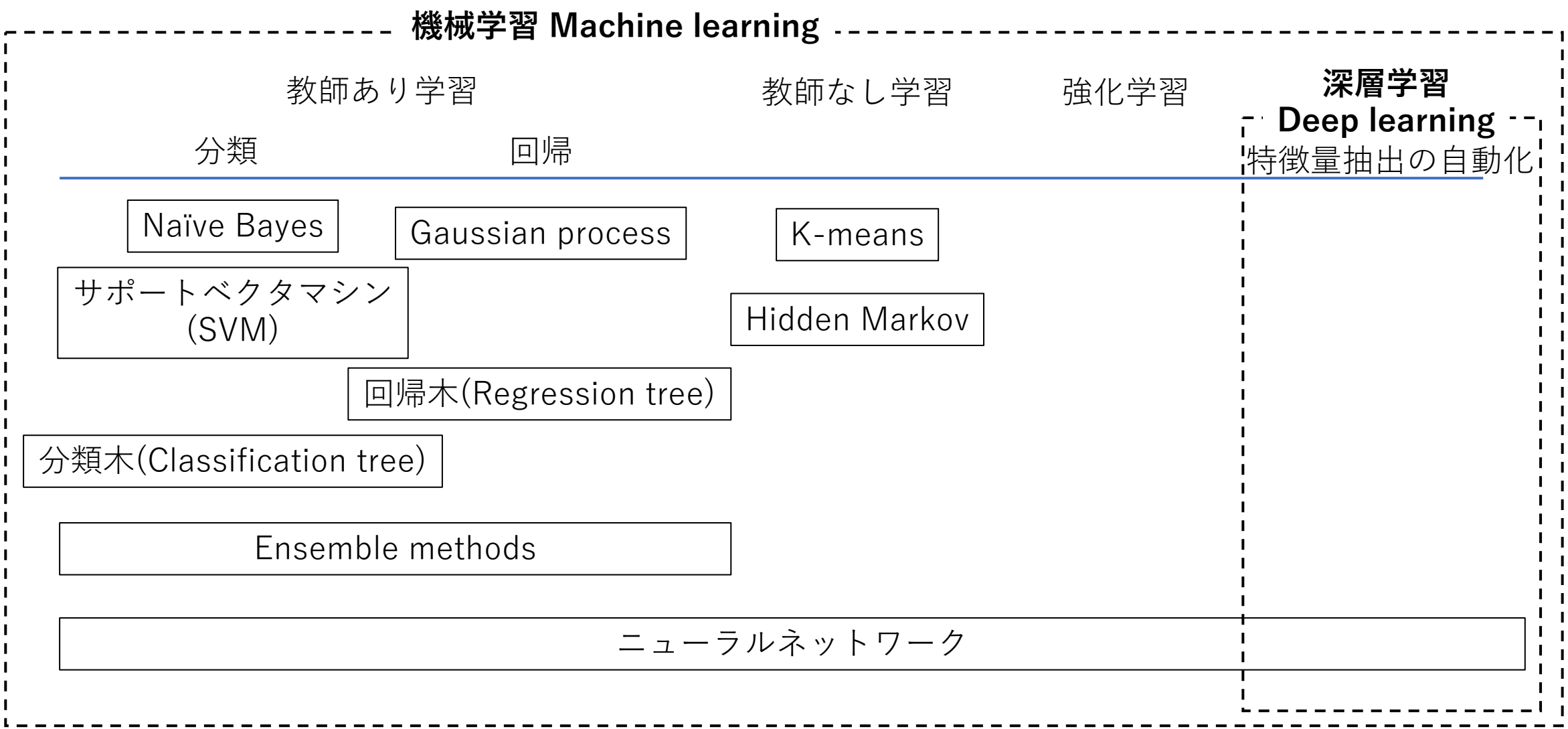
1. 統計的機械学習の基礎

具体例

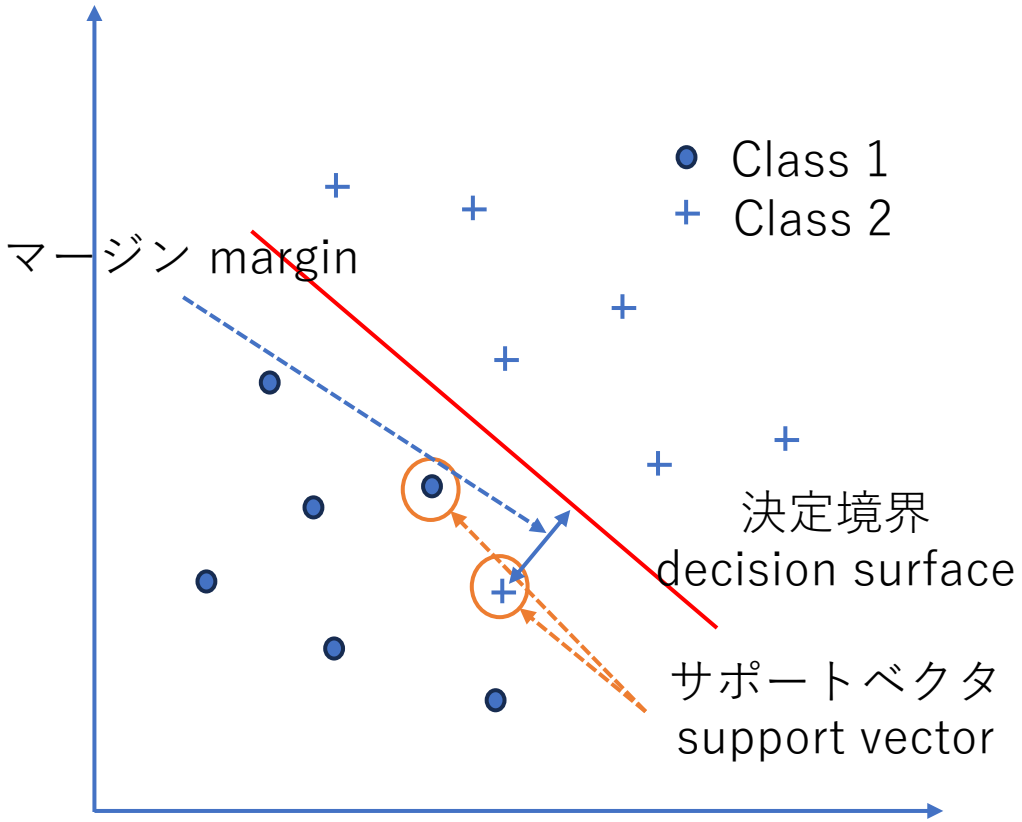
- 教師あり学習
  - 2値分類問題　$y = \{0, 1\}$，ロジスティック損失$l = \log(1 + e^{-yh(x)})$
  - 分類問題　$y = \{0, 1, \cdots, N-1\}$，交差エントロピー損失$l = -\sum_i \delta_{i,y} \log h_i(x)$
  - 回帰問題　$y \in \mathbb{R}$，二乗損失$l = (h(x) - y)^2$
- 教師なし学習
  - オートエンコーダ　再構成損失$l = |h(x) - x|^2$

    ただし，$h(x) = x$とすれば$l = 0$となってしまうので，通常は次元を絞るなどして$\mathcal{H}$を限定する．
- 強化学習
  - 行動主体(agent)が環境(environment)の中で行動する．
  - 観測される環境の状態$s(t) \in \mathcal{S}$
  - 方策(policy)πに従う行動(action)$a(t) \in \mathcal{A}$
  - 環境が行動主体に与える報酬(reward)$\mathrm{r}(t+1) = r\big(a(t), s(t), s(t+1)\big) \in \mathbb{R}$
  - 期待累積報酬$R(t|\pi) = E[\sum_t \gamma^t r(t+k+1)\,|\pi]$を最大化する$\pi$を学習

2. 機械学習の代表的手法



2. サポートベクタマシン(SVM)

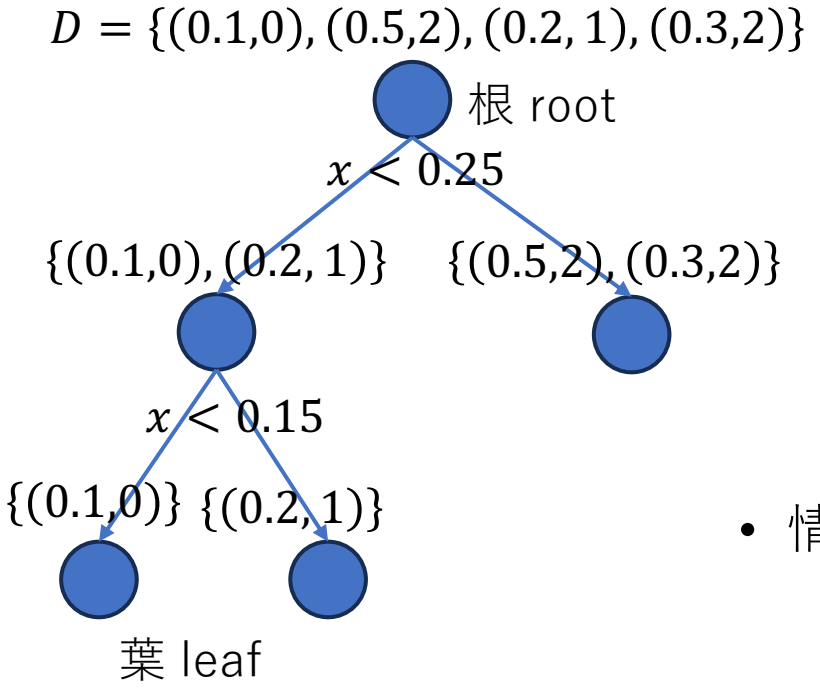


SVM
- 主に分類問題に用いられる.
- 説明変数空間での距離を扱うので，正則化が必要.

- 決定境界：分類に用いる境界線 $w^T x + w_0 = 0$
- サポートベクタ：各クラスの決定境界に最も近い点
- マージン：サポートベクタと決定境界の距離
  - $m = \min_i \frac{|w^T x_i + w_0|}{|w|}$
  - クラスの決定境界に対する方向をy_iとすると，$\frac{y_i(w^T x_i + w_0)}{|w|} \geq m$
- マージンを最大化するような決定境界を学習
  - マージンの定義より，$\frac{y_i(w^T x_i + w_0)}{m|w|} - 1 \geq 0$
  - $|w| = 1/m$となるように標準化すると，
  - $\min |w|^2 \quad \text{s.t. } g_i(w, w_0) = y_i(w^T x_i + w_0) - 1 \geq 0$

  →ラグランジュの未定乗数法により求解.
  - カーネルトリックを使うと非線形な決定境界も考えられる.

2. 決定木(Decision tree)

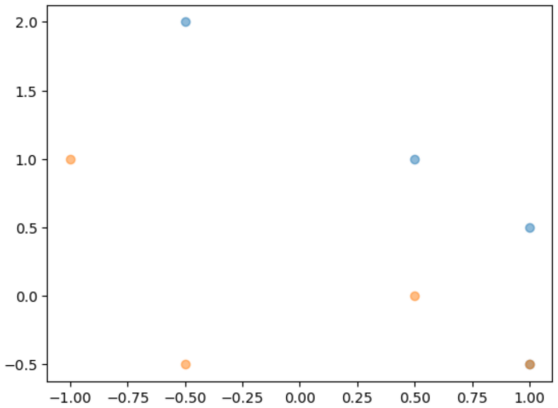


決定木（分類木と回帰木の総称）

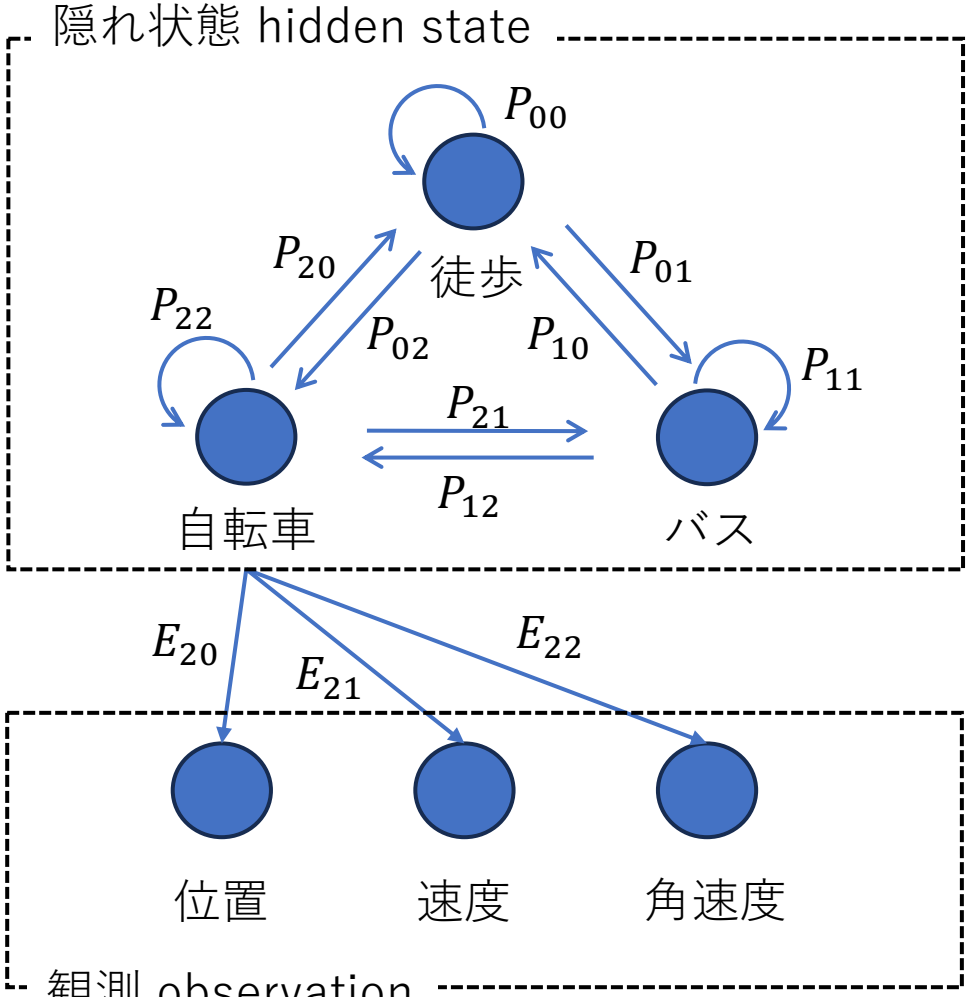
- 根ノードから条件分岐を繰り返し，至った葉ノードの値を返す.
- 不純度・エントロピー
  - **ジニ不純度 Gini impurity**
    - ノードt内のクラスiのトレーニングサンプル割合$p(i|t) = n_{i,t}/n_t$
    - ジニ不純度$I_G(t) = 1 - \sum_i p(i|t)^2$
    - 全て同じクラスなら$I_G(t) = 0$
    - 全サンプルが異なるクラスなら$I_G(t) = 1 - 1/n_t$
  - **エントロピー Entropy**
    - $I_H(t) = -\sum_i p(i|t) \log p(i|t)$
    - 全て同じクラスなら$I_H(t) = 0$
    - 全サンプルが異なるクラスなら$I_H(t) = \log n_t$
- 情報利得 Information gain
  - 条件分岐による不純度・エントロピーの減少分を最大化
    - $\Delta I = I(t) - I(t_{left}) - I(t_{right})$
  - CART法：ジニ不純度の最小化
  - ID3, C4.5：エントロピーの最小化

2. 練習問題（決定木）

- データ
  - クラス0：{(-0.5, 2.0), (0.5, 1.0), (1.0, 0.5), (1.0, -0.5)}
  - クラス1：{(-1.0, 1.0), (-0.5,-0.5), (0.5, 0.0), (1.0, -0.5)}
- データをジニ不純度を最小にするように一回分割してください.



2. 隠れマルコフモデル(Hidden Markov model)

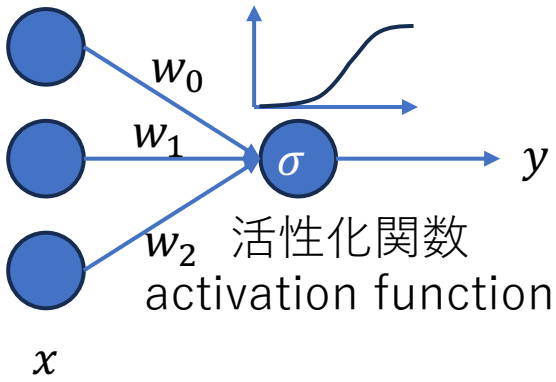


隠れマルコフモデル HMM

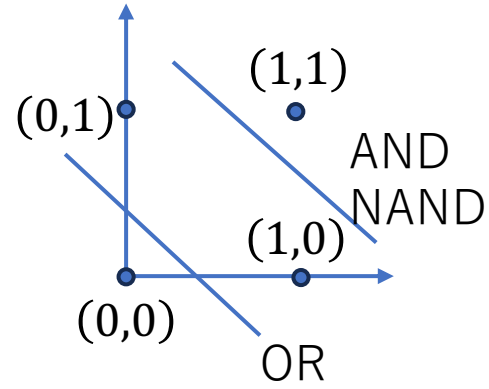
- マルコフ性 Markov property：次状態の確率分布が現在状態のみに依存する.
- マルコフ連鎖 Markov chain：離散的な状態，離散時間でのマルコフ過程
  - $P(X_{t+1} = x | X_t = x_t, X_{t-1} = x_{t-1}, \cdots, X_0 = x_0) = P(X_{t+1} = x | X_t = x_t)$
  - 初期状態確率ベクトル initial probability vector π_0
  - 遷移行列 transition matrix T
  - 出力行列 emission matrix E
- Baum-Welchアルゴリズム(EMアルゴリズム)により推定
  - EMアルゴリズム：欠損値を含むデータでの推定
    →隠れ状態が欠損値

3. 多層パーセプトロン

パーセプトロン



- 脳神経のニューロンの発火を模擬
- 線形分離可能な分類問題を求解可能
- 論理回路としての表現力
  - AND $(0,0),(0,1),(1,0) \rightarrow 0,(1,1) \rightarrow 1$
  - OR $(0,0) \rightarrow 0,(0,1),(1,0),(1,1) \rightarrow 1$
  - NAND$(1,1) \rightarrow 0,(0,0),(0,1),(1,0) \rightarrow 1$
  - XORは表現できない.
    $(0,0),(1,1) \rightarrow 0,(0,1),(1,0) \rightarrow 1$

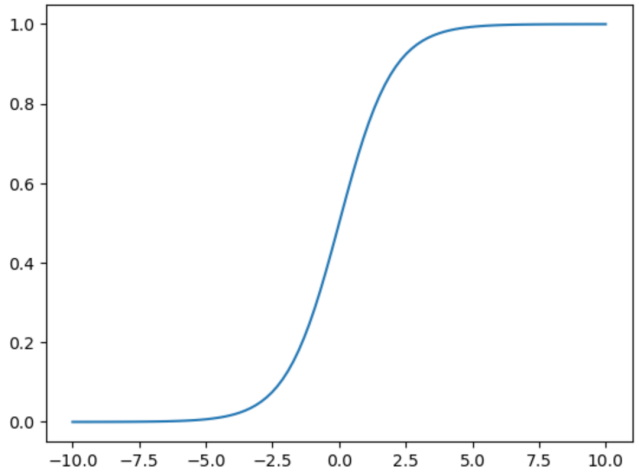


万能近似定理

- 2層以上のニューラルネットワークは任意の連続関数を任意の精度で近似できる.
- 理論上は浅いニューラルネットワークで十分だが，深いネットワークを用いることで階層的な表現が可能となる.

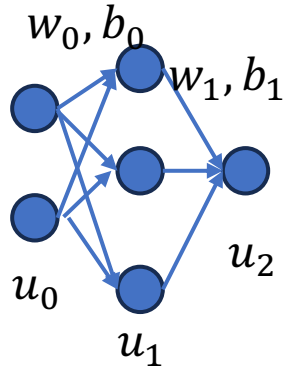
3. 練習問題（多層パーセプトロン）

- AND, OR, NANDを表すパーセプトロン$y = sigmoid(w^T x + b)$を一つずつあげてください.
- ここでは，$x \in \{0,1\}^2$とし，$sigmoid(-10) \approx 0, sigmoid(10) \approx 1$とできるとします.



3. 後方誤差伝搬

- 微分の連鎖率 chain rule
  - $\frac{\partial f \circ g}{\partial x} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial x}$
- n層のニューラルネット
  - $u_{i+1} = f\left(w_i^T u_i + b_i\right) \coloneqq \frac{1}{1+e^{-\left(w_i^T u_i + b_i\right)}},\ \therefore f' = f(1-f)$
  - 損失関数　$L = \mathcal{L}(y, u_n) \coloneqq \frac{1}{2}(u_n - y)^2, \therefore \frac{\partial \mathcal{L}}{\partial u_n} = (u_n - y)^T = \delta_n$
  - 各層での誤差：$\delta_{n-1} \coloneqq \delta_n\{u_n \circ (1-u_n)\}, \delta_{i-1} \coloneqq w_i^T \delta_i\{u_i \circ (1-u_i)\}$
  - $\frac{\partial L}{\partial w_{n-1}} = \frac{\partial \mathcal{L}}{\partial u_n}\frac{\partial u_n}{\partial w_{n-1}} = \frac{\partial \mathcal{L}}{\partial u_n} f'(w_{n-1}^T u_{n-1} + b_{n-1}) u_{n-1}^T$
    $= (u_n - y)^T\{u_n \circ (1-u_n)\}u_{n-1}^T = \delta_{n-1}\, u_{n-1}^T$
  - $\frac{\partial L}{\partial w_{n-2}} = \frac{\partial \mathcal{L}}{\partial u_n}\frac{\partial u_n}{\partial u_{n-1}}\frac{\partial u_{n-1}}{\partial w_{n-2}} = \frac{\partial \mathcal{L}}{\partial u_n} f'(w_{n-1}^T u_{n-1} + b_{n-1}) w_{n-1}^T f'(w_{n-2}^T u_{n-2} + b_{n-2}) u_{n-2}^T$
    $= \delta_{n-1} w_{n-1}^T\{u_{n-1} \circ (1-u_{n-1})\}u_{n-2}^T = \delta_{n-2} u_{n-2}^T$

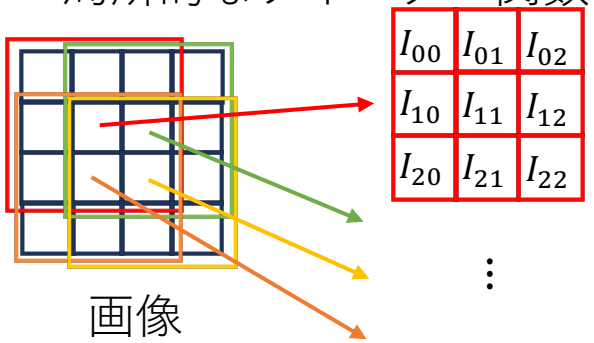


$$\rightarrow \frac{\partial L}{\partial w_{n-i}} = \delta_{n-i} u_{n-i}^T$$

3. Convolutional Neural Network (CNN)

CNN：[畳み込み層+活性化関数+プーリング層]を積み重ねる

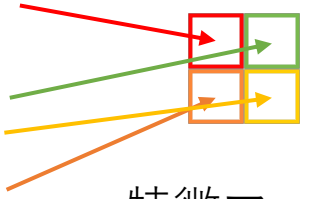
畳み込み層 convolutional layer：
局所的なフィルター関数を順に作用させる.



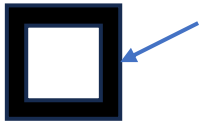
$$\begin{matrix} I_{00} & I_{01} & I_{02} \\ I_{10} & I_{11} & I_{12} \\ I_{20} & I_{21} & I_{22} \end{matrix} \vdots$$

フィルター

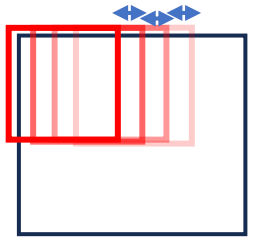
$$\circ \quad \begin{matrix} A_{00} & A_{01} & A_{02} \\ A_{10} & A_{11} & A_{12} \\ A_{20} & A_{21} & A_{22} \end{matrix}$$



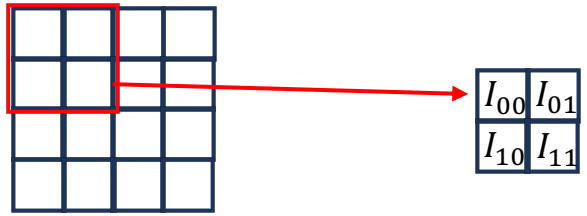
ハイパーパラメータ
・カーネルサイズ kernel size
・出力チャンネル数 output channel
・パッディング padding



・ストライド stride



プーリング層 pooling layer：
特徴量をまとめることにより大域的で抽象的な表現を得る.



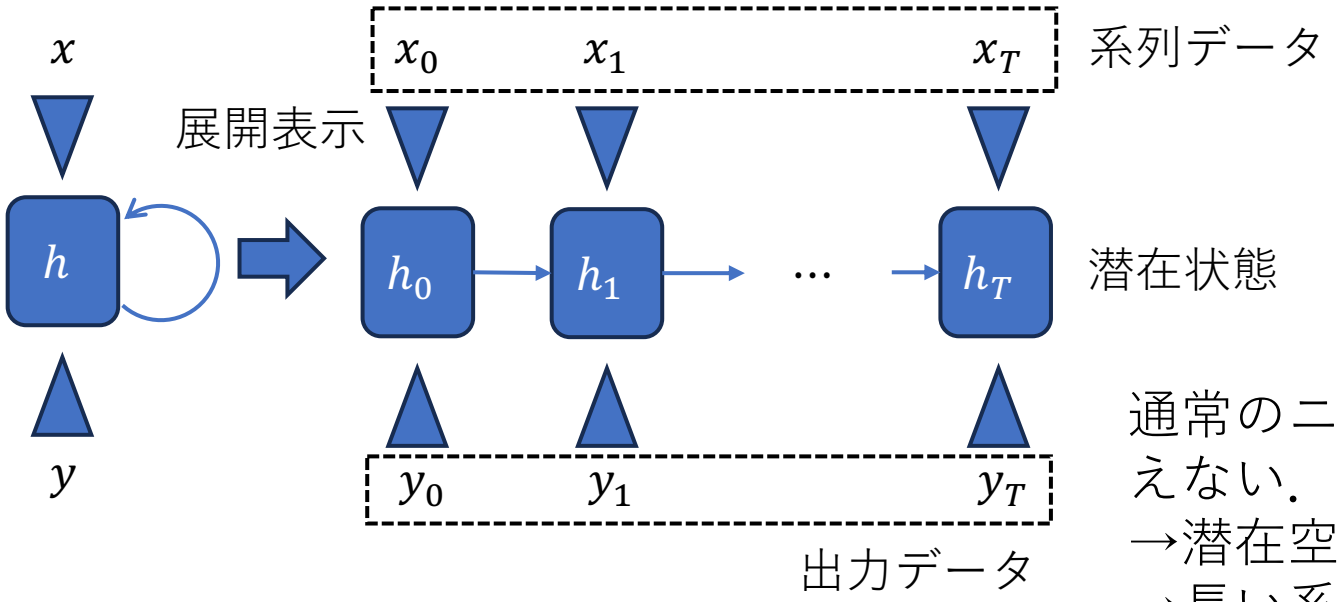
$$\begin{matrix} I_{00} & I_{01} \\ I_{10} & I_{11} \end{matrix}$$

$$\max(I_{00}, I_{01}, I_{10}, I_{11})$$

ハイパーパラメータ
・ウィンドウサイズ window size
・パッディング padding
・ストライド stride

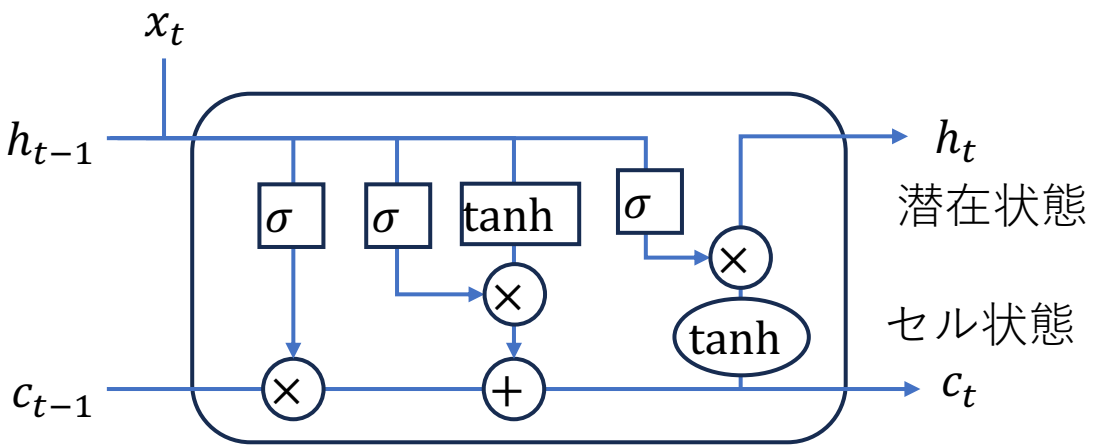
○パラメータ数を大幅に抑えられる．画像サイズが変わっても適用可能．並行移動に対して不変.
×回転や収縮を扱うのが困難．ノイズ添加により出力が大幅に変わる場合あり.

3. Recurrent Neural Network (RNN)



通常のニューラルネットワークは**可変長の入出力**を扱えない.
→潜在空間での再帰的な演算
→長い系列における記憶の保持(LSTM)

LSTM



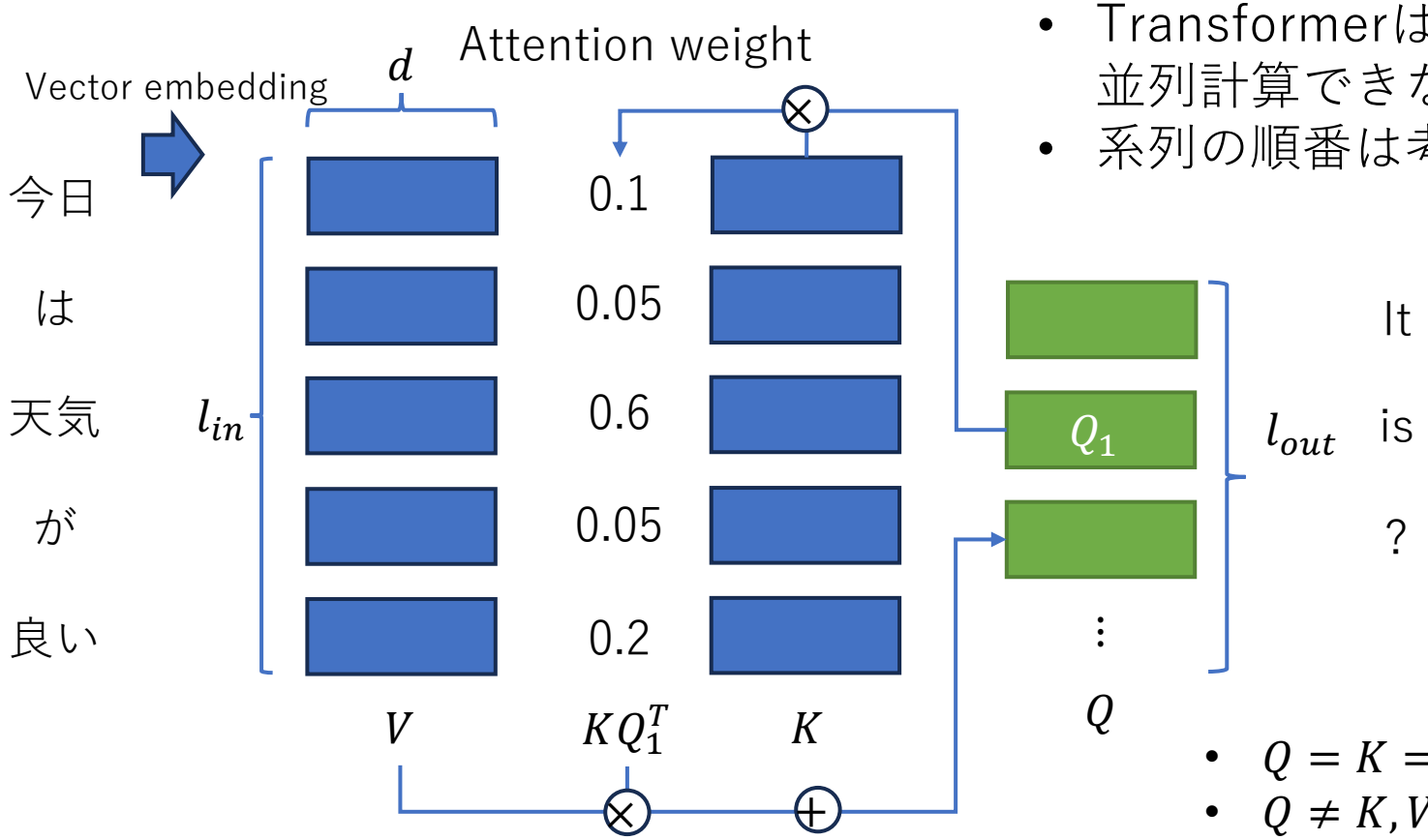
○可変長の入出力を扱える.
×前から順に計算するので並列計算ができない.

セル状態：系列の長期的な情報を保持

3. Attention

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$Q \in \mathbb{R}^{l_{out} \times d}, K \in \mathbb{R}^{l_{in} \times d}, V \in \mathbb{R}^{l_{in} \times d}$$

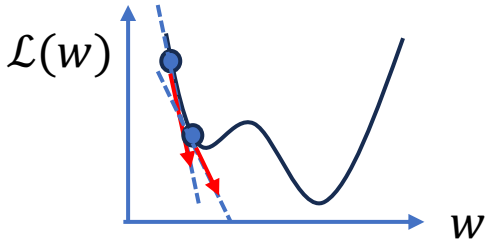


Attention機構：データのどの部分に注目するか

- CNNのフィルターとイメージは近い
- CNNによる特徴量抽出などとも組み合わせ可能
- TransformerはAttentionを内部に持つことでRNNの並列計算できない性質を克服
- 系列の順番は考慮されない→位置エンコーディング

- $Q = K = V$ self-attention
- $Q \neq K, V$ SourceTarget-attention

4. 機械学習モデルの学習アルゴリズム

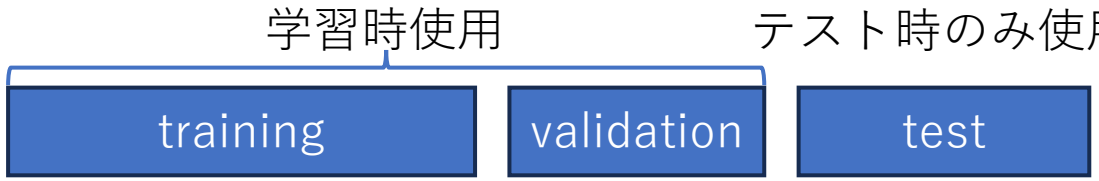


確率的勾配降下法 stochastic gradient descent(SGD)

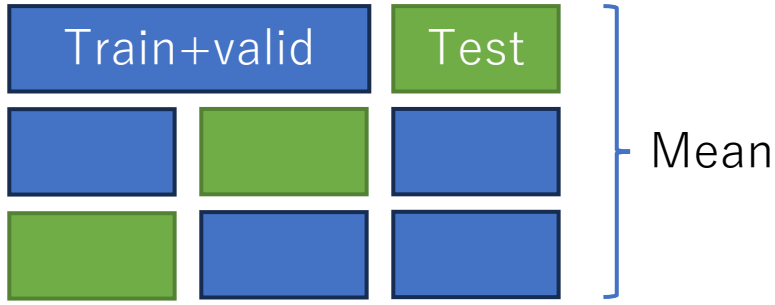
- $w \leftarrow w - \alpha \nabla_w \mathcal{L}(w)$
- ハイパーパラメータ：学習率α
- 「確率的」：一つのデータのみについての勾配を用いる→データの取り方による勾配の変動が生じる
  - 並列化ができない
  - ミニバッチSGD：複数のデータについての勾配を用いることで計算効率を向上
- 学習の進行とともに更新幅を調整するアルゴリズムが多数存在
  - Momentum SGD：勾配値を調整
    - $g_t \leftarrow \nabla_w \mathcal{L}(w), v_t \leftarrow \mu v_{t-1} + g_t, w_t \leftarrow w_{t-1} - \mu v_{t-1} - \gamma g_t$
  - RMSProp：学習率を調整
    - $g_t \leftarrow \nabla_w \mathcal{L}(w), s_t \leftarrow \beta s_{t-1} + (1-\beta) g_t^2, w_t \leftarrow w_{t-1} - \frac{\alpha}{\sqrt{s_t}+\varepsilon} g_t$
  - Adam：モーメンタムと学習率調整の組み合わせ
    - $g_t \leftarrow \nabla_w \mathcal{L}(w), v_t \leftarrow \beta_1 v_{t-1} + (1-\beta_1) g_t, s_t \leftarrow \beta_2 s_{t-1} + (1-\beta_2) g_t^2, w_t \leftarrow w_{t-1} - \frac{v_t}{\sqrt{s_t}+\varepsilon} g_t$

4. モデルの検証方法

- データの分割
  - **教師データ** training data：勾配計算・モデル更新に用いるデータ
  - **検証データ** validation data：モデル学習中にモデルが過学習していないことを検証するためのデータ
  - **テストデータ** test data：学習済みモデルの評価に用いるデータ
- ホールドアウト検証 hold-out validation
  - 全データを教師データ＋検証データとテストデータに分割し，学習とテストを行う．



- 交差検証法 k-hold cross validation
  - 全データをいくつかに分割し，教師データ＋検証データとテストデータを取り替えながら学習とテストを複数回行う．
  - データ数がある程度少ない場合でも適用可能



参考

- MATLABの機械学習チュートリアル https://jp.mathworks.com/solutions/machine-learning/tutorials-examples.html
- Courseraの深層学習コース https://www.coursera.org/specializations/deep-learning
- Tensorflowの機械学習チュートリアル　https://www.tensorflow.org/resources/learn-ml?hl=ja
- 松井孝太，熊谷亘．転移学習．機械学習プロフェッショナルシリーズ．2024.