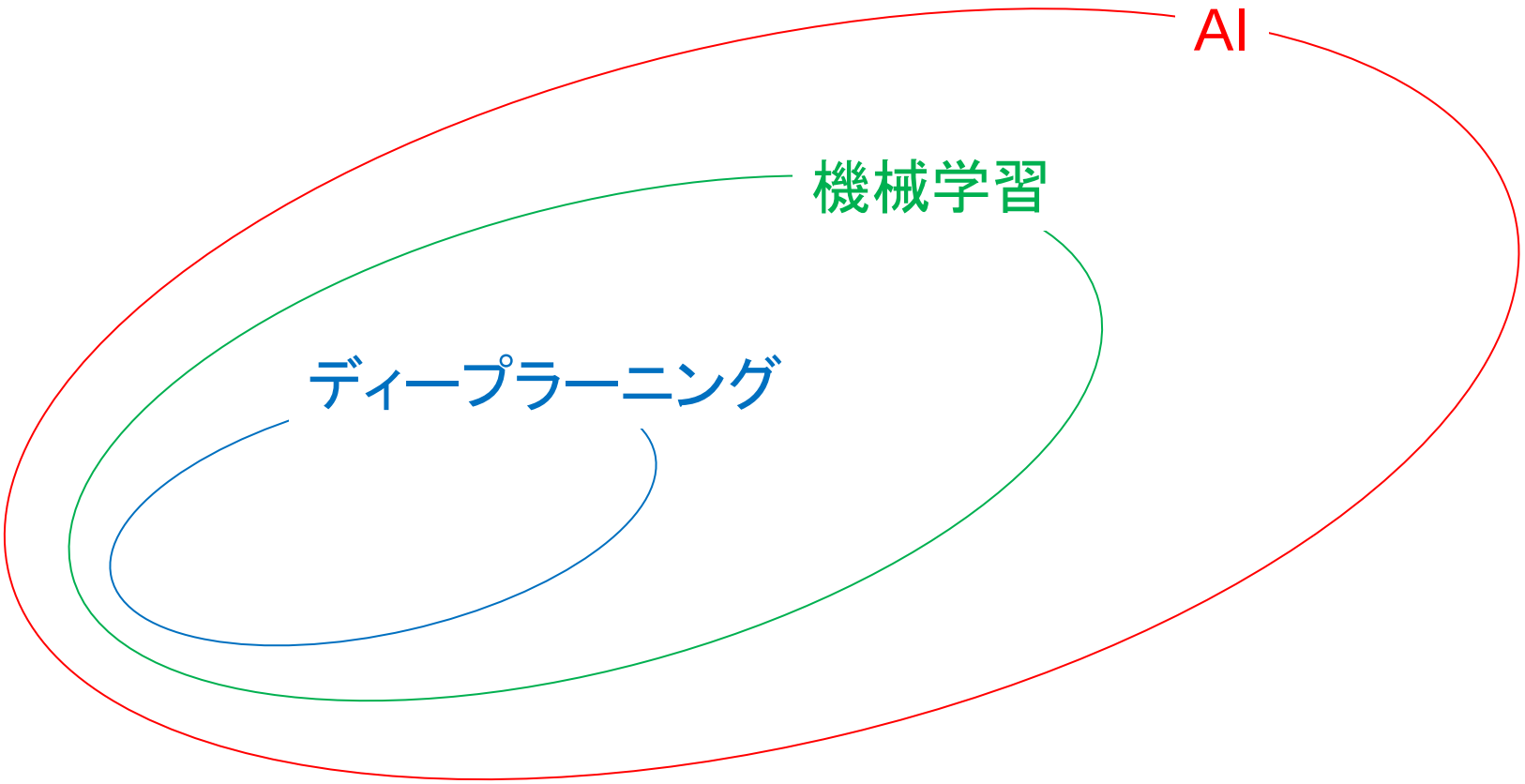
はじめての機械学習
(データ科学)

Outline

- 機械学習の利用で重要な知識
- 機械学習用の計算時間の考え方
- 機械学習で分かること
- 第一原理計算との連携
- データベース

機械学習の利用で重要な知識

機械学習は人工知能（AI）の一分野



人工知能（学会）における研究分野は多くある
例えば、機械学習、推論機構、知識表現、メディア理解など

ディープラーニングはAlphaGo（アルファ碁）などが有名

知能化技術の世代

第一世代	単純動作	プログラム通り動くだけ：論理、定理証明、自動販売機など
第二世代	探索・知識	手順に従って解を探す：知識工学、エキスパートシステムなど
第三世代	機械学習	処理や知識を自動獲得: 統計的機械学習、深層学習など
第四世代	自動定式化	問題の解法を自立生成、最適化、探索、進化計算など
第五世代	感性理解	人とロボットの共生：感性脳情報科学など
第六世代	人工能	脳と同等の機能の構築、高度知能、脳型計算など

世代の付け方は研究者によって異なる

第二世代のエキスパートシステムは上手くいかなかった

第三世代のディープラーニング（深層学習）がAlphaGo
ある特定の機能に特化した特化型のAIはいくつか成功している

現在、ある特定の機能に限定されない汎用型のAIの開発が進められている

これまで何度もAIは冬の時代を経験している
その理由は自由な発想でAIを開発できていないため
上の表は気にせず自由な発想で開発して欲しい

情報エントロピー

$$\sum_{i}^{N} p_i = 1$$

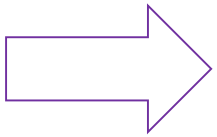
確率

情報エントロピー　$$H = -\sum_{i}^{N} p_i \log_2 p_i$$

対数（log）の底はどれでも良いが、2を選ぶことが多い

熱力学でのエントロピーは乱雑（ランダム）であると大きい値
同じように、情報エントロピーは不確定（ランダム）であるほど大きな値

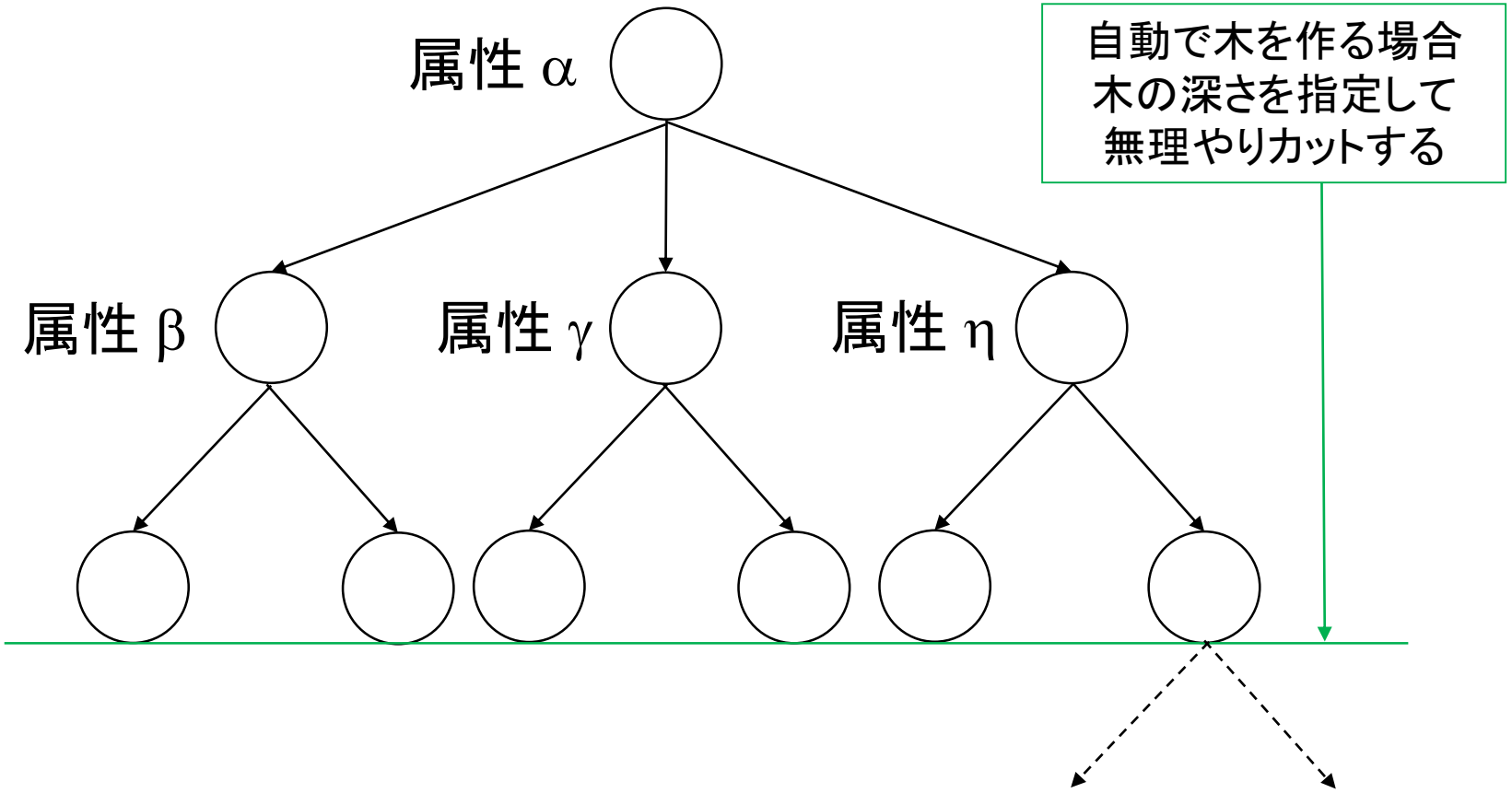
情報エントロピー：大
不確定要素が多い



情報エントロピー：小
確定的な要素が多い

一般に、情報エントロピーが大きく減少する属性値から順番に分類することで
簡潔な表現を行うことができる

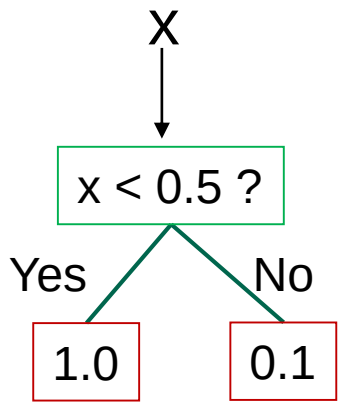
決定木 [2]

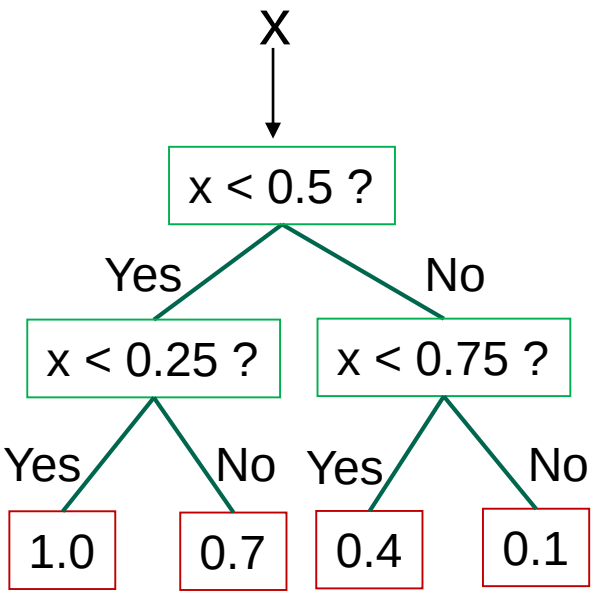


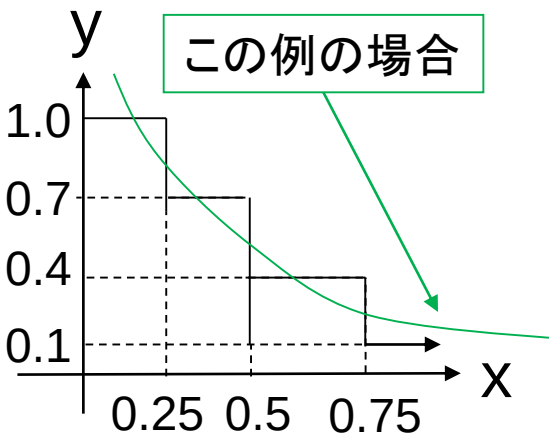
一般に、情報エントロピーが大きく減少する属性値から順番に分類することで
簡潔な表現を行うことができる

代表的な決定木による分類学習システムにID3（アイディースリー）がある
その改良手法にC4.5がある

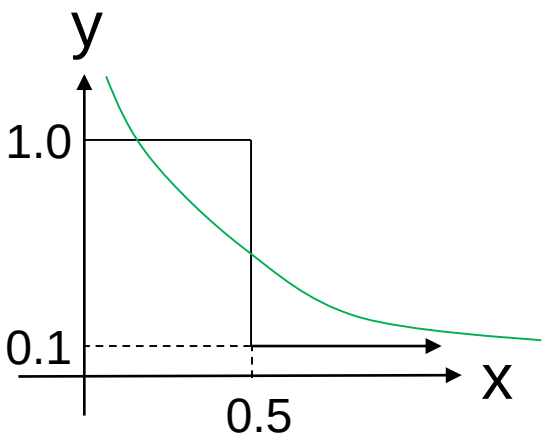
回帰木

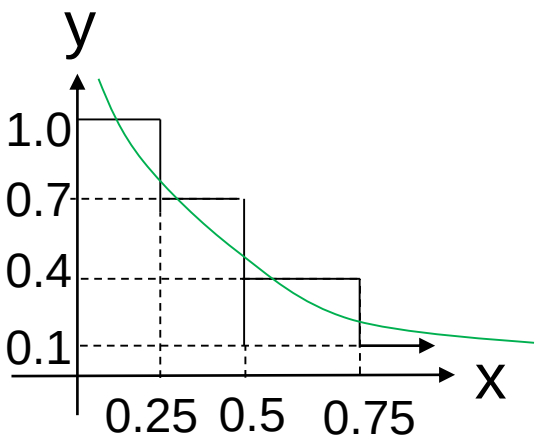


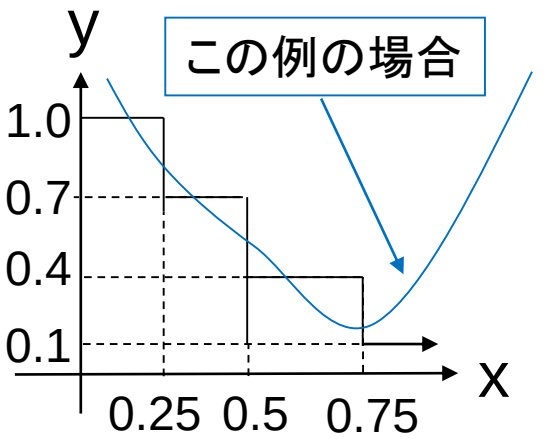




0.75を超えても良さそう







0.75を超えると悪い

木を深くしていくと当てはまりが良くなることがわかる

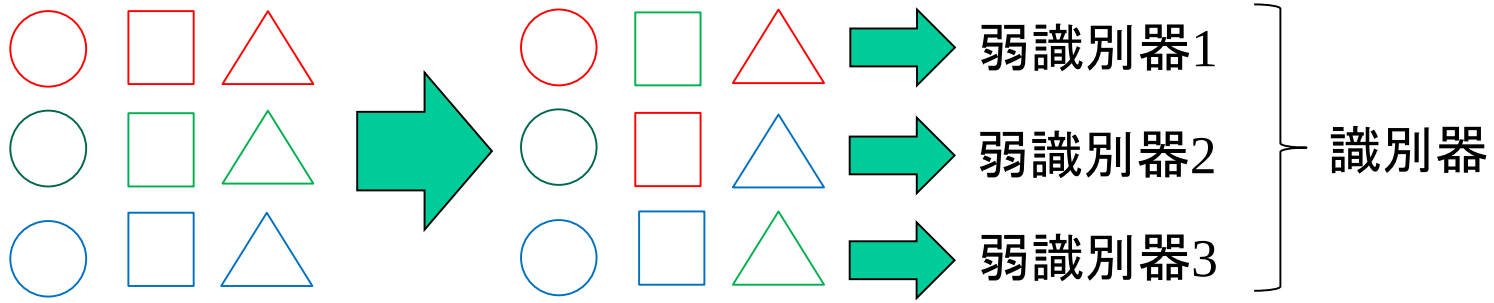
回帰木は良い相関が出やすいが、外挿領域では当てはまりが悪い理由が上記からわかる

アンサンブル学習 [2,3,4]

複数の弱学習器を融合して汎化性能を向上させる学習手法
バギング（Bagging）、ランダムフォレスト（Random Forest）、ブースティング（Boosting）

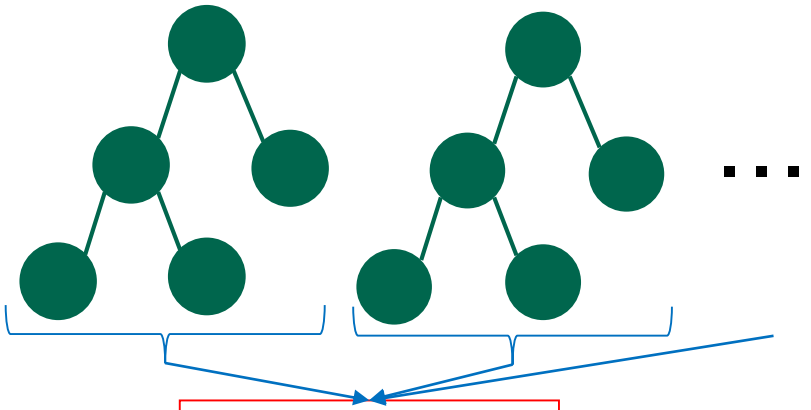
バギング（Bagging）

全体の学習データ　　ブートストラップサンプル（ランダムに取り出されたデータ）



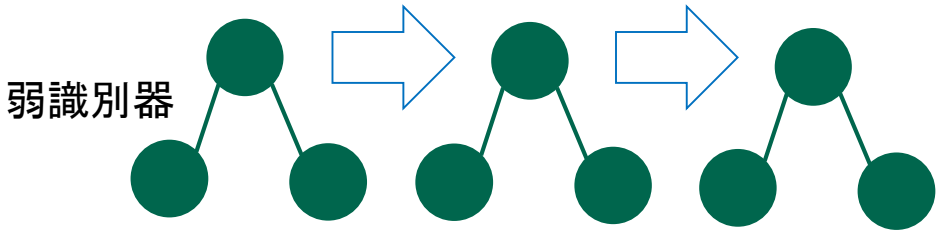
Random Forest（決定木のBagging）

複数の決定木をランダムに生成
各決定木の多数決で最終判断



AdaBoost（Boosting）

苦手な問題を他の弱識別器へ
他の弱識別器で誤分類したものを優先的に学習



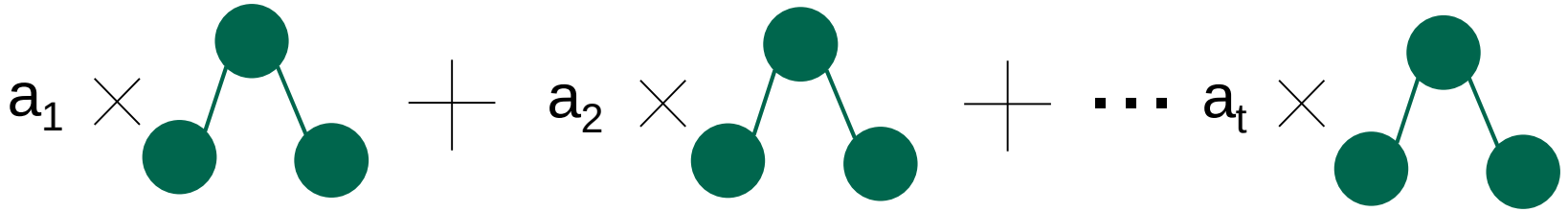
100個程度の弱識別器が必要（処理時間かかる）

弱識別器ごとに、どの学習データを重視するかを変えて学習

識別器を何個用いるかなど経験則が多い

AdaBoost と Gradient Boost

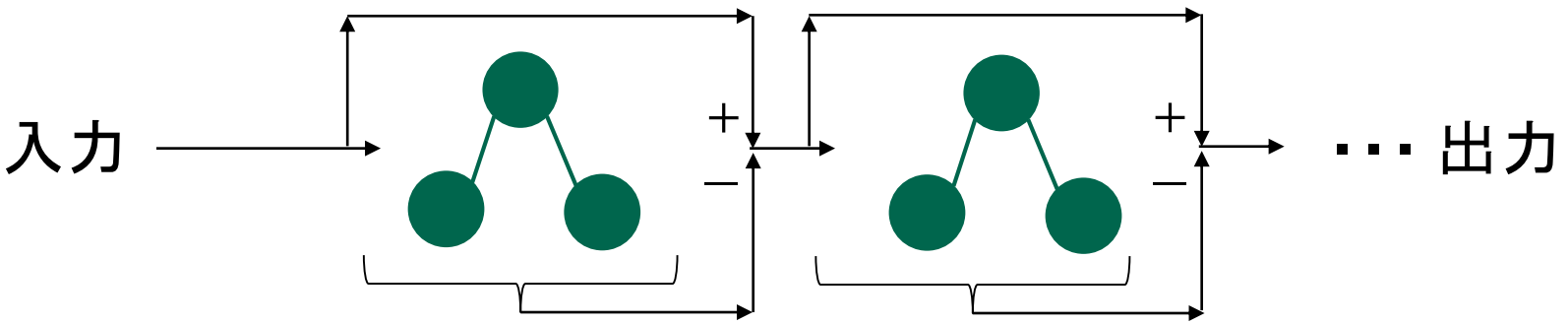
AdaBoost（Boosting）
苦手な問題を他の弱識別器へ
他の弱識別器で誤分類したものを優先的に学習



弱識別器

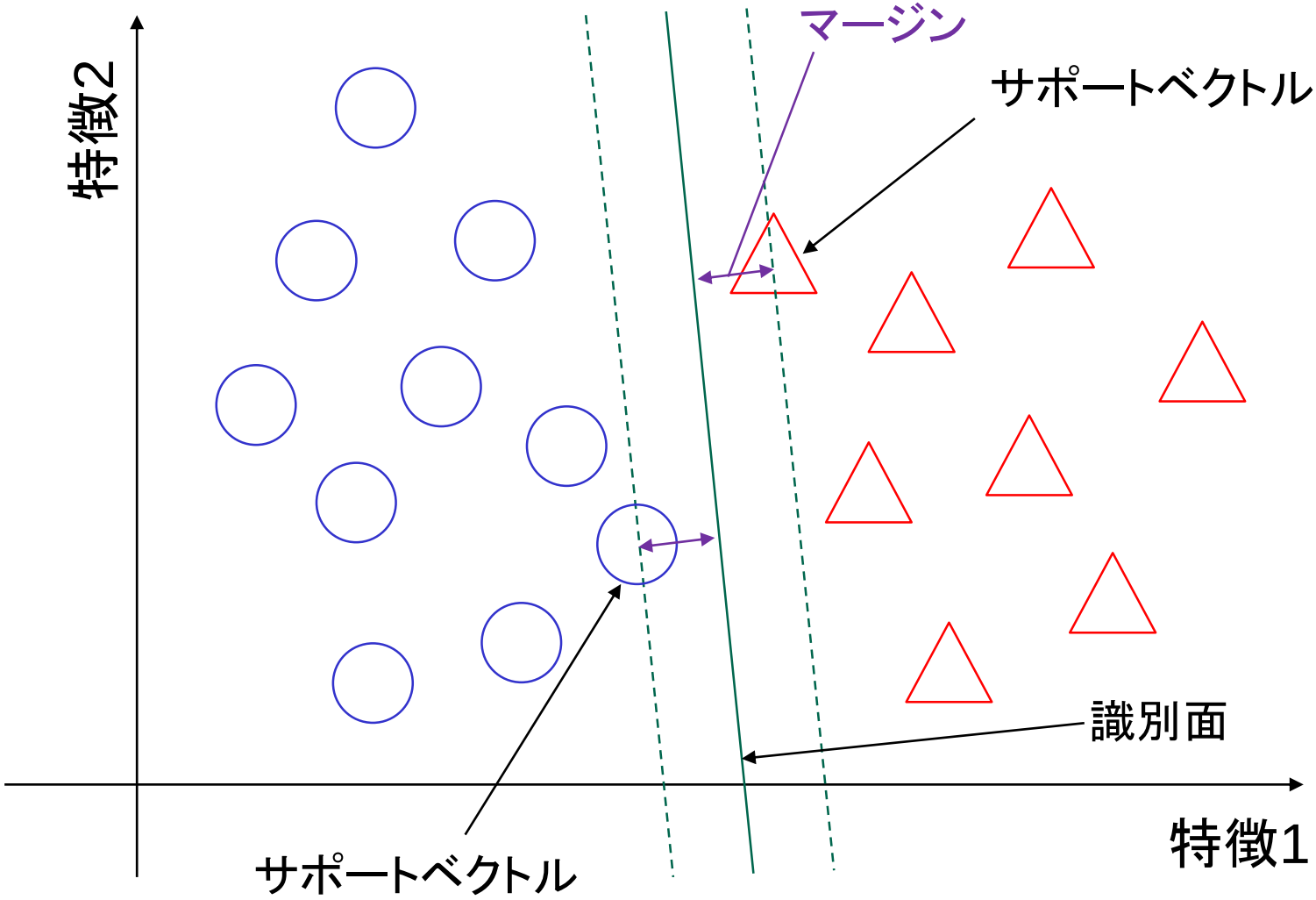
それぞれの弱識別器の重みa_tを変える

Gradient Boost（勾配ブースティング (Boosting)）
入力と弱識別機の差（残差）を次の弱識別機への入力へ



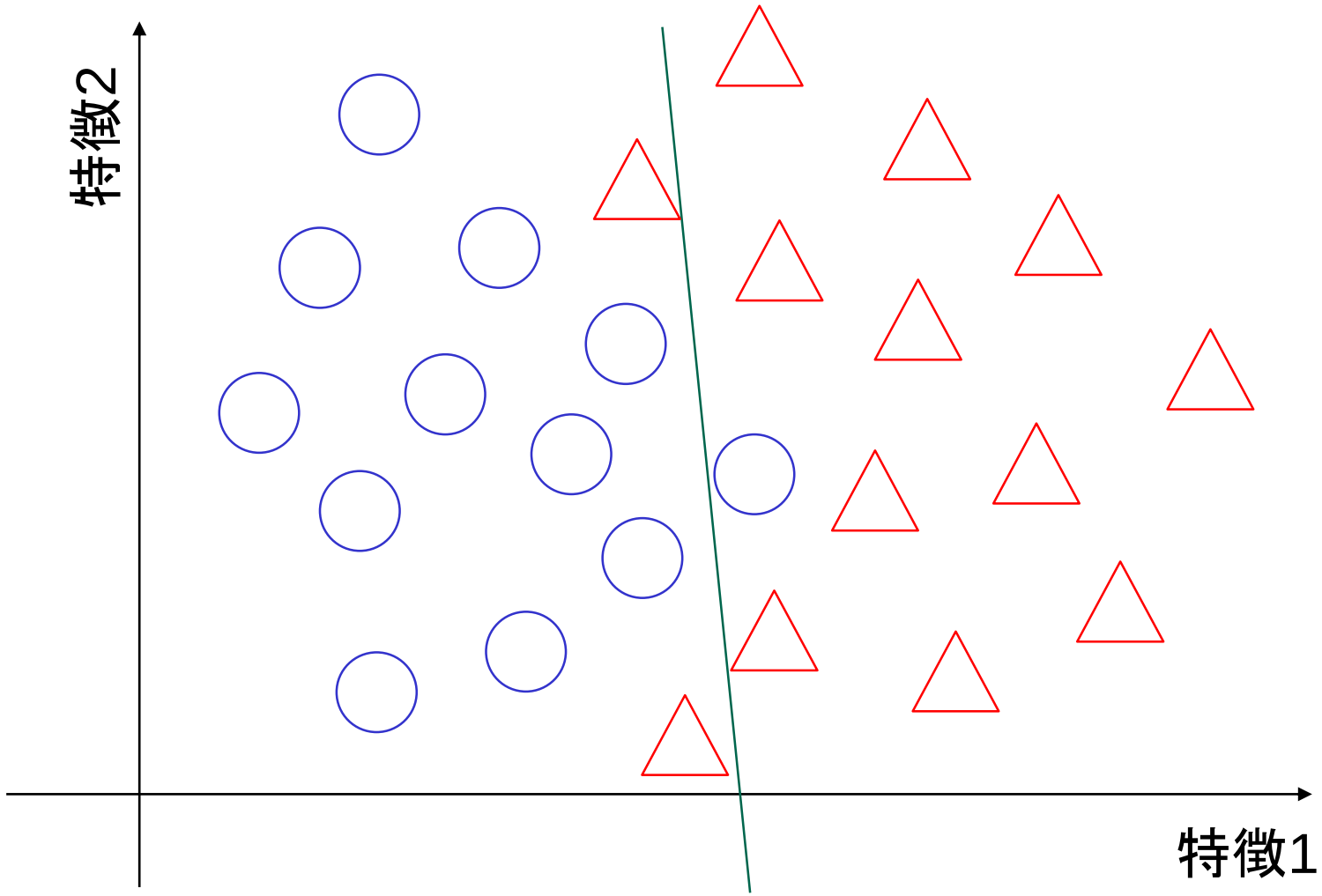
実用上、たいていの場合、勾配ブースティングが最も精度が良い
（ランダムフォレストの方が精度が良い場合も多くある）

サポートベクターマシーン（SVM）[3]



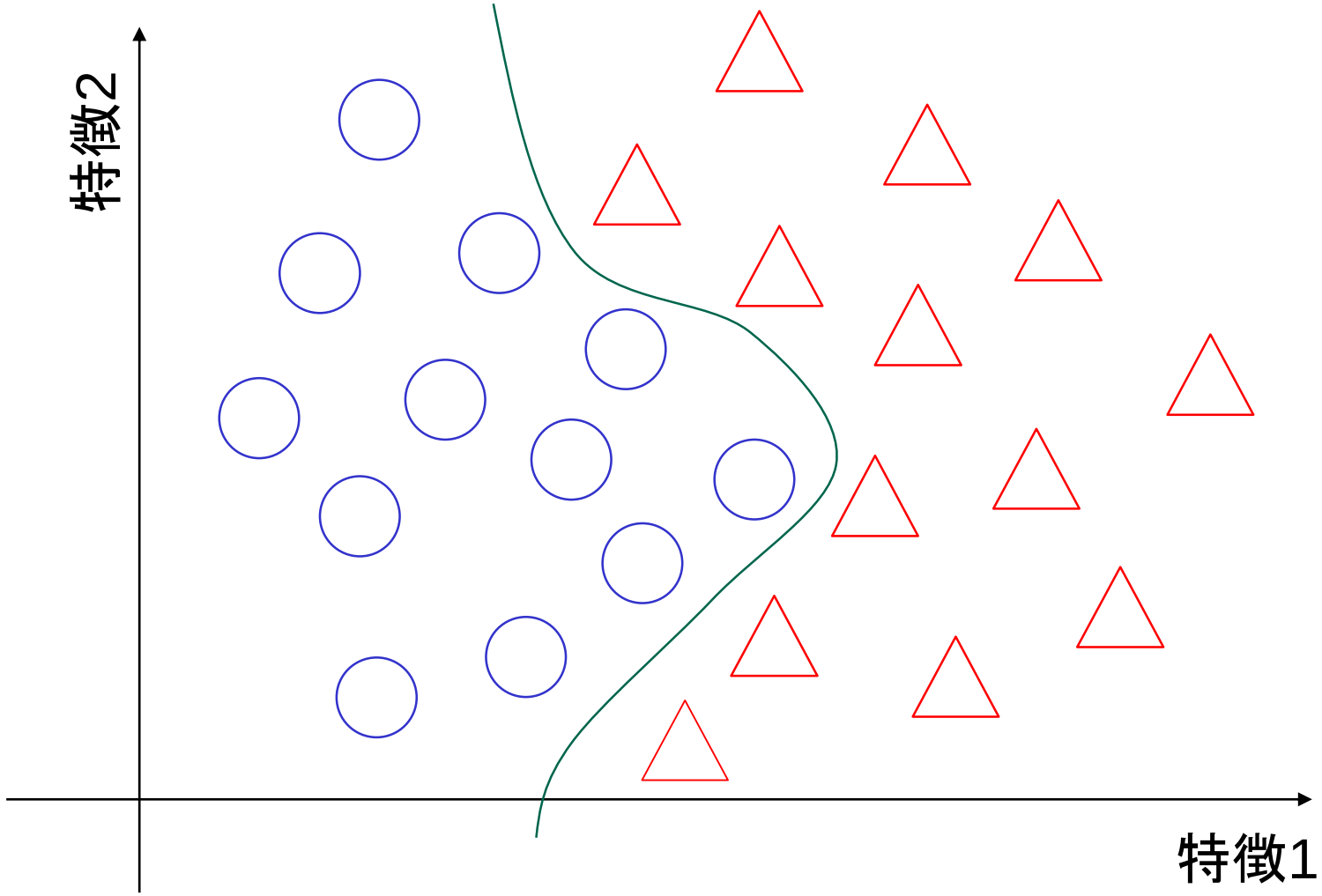
最も近いサンプル（サポートベクトル）からの距離（マージン）が
最大となるように分類する

ソフトマージンSVM [3]



完全に分離できない（線形分離不可能）場合
ある程度の誤差を許容するようにする

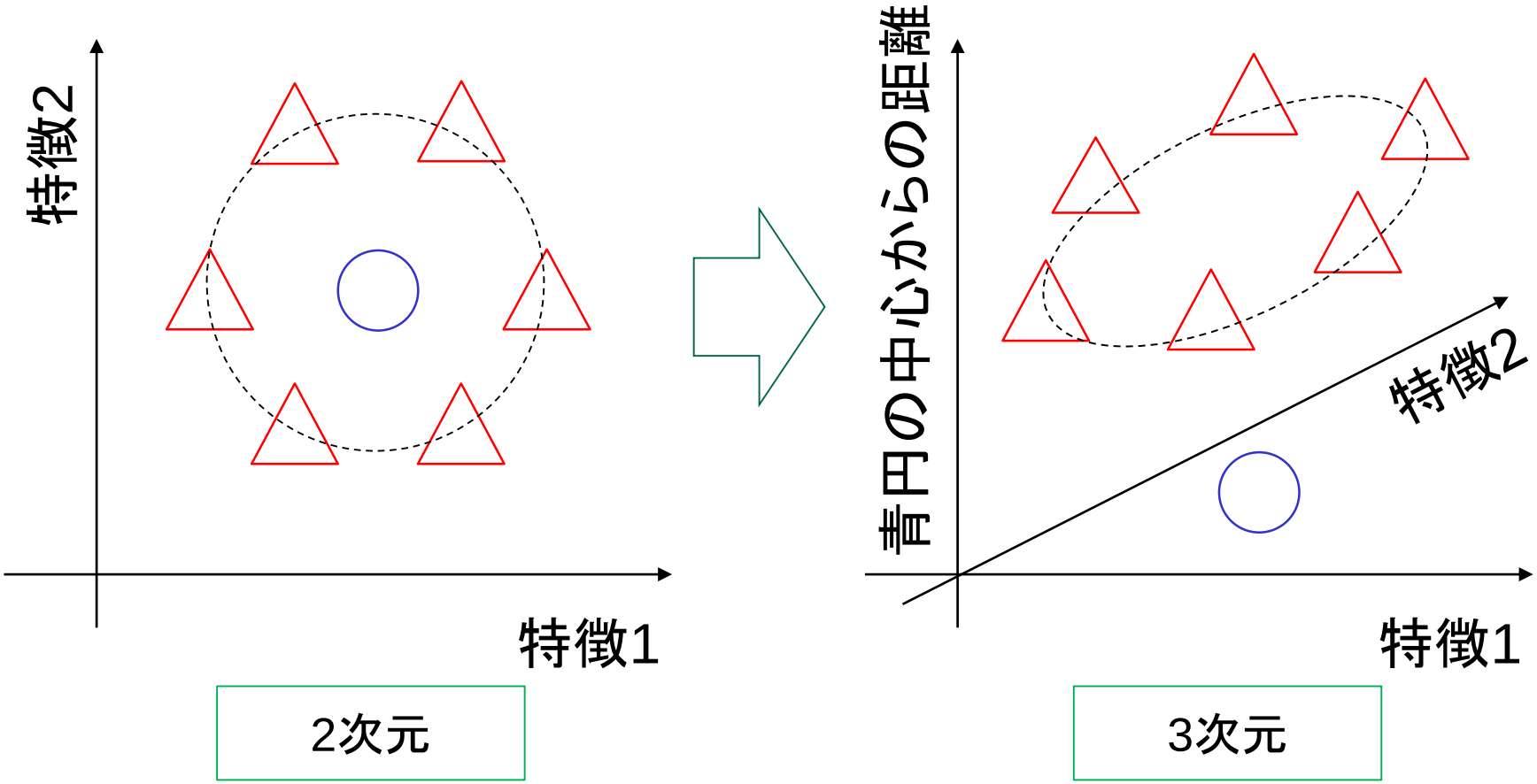
非線形SVM [2]



直線ではない形で分類する方法

データを見かけ上、次元を上げて超平面で分離する
（カーネルトリックと呼ばれる）

次元を上げて分類するとは？



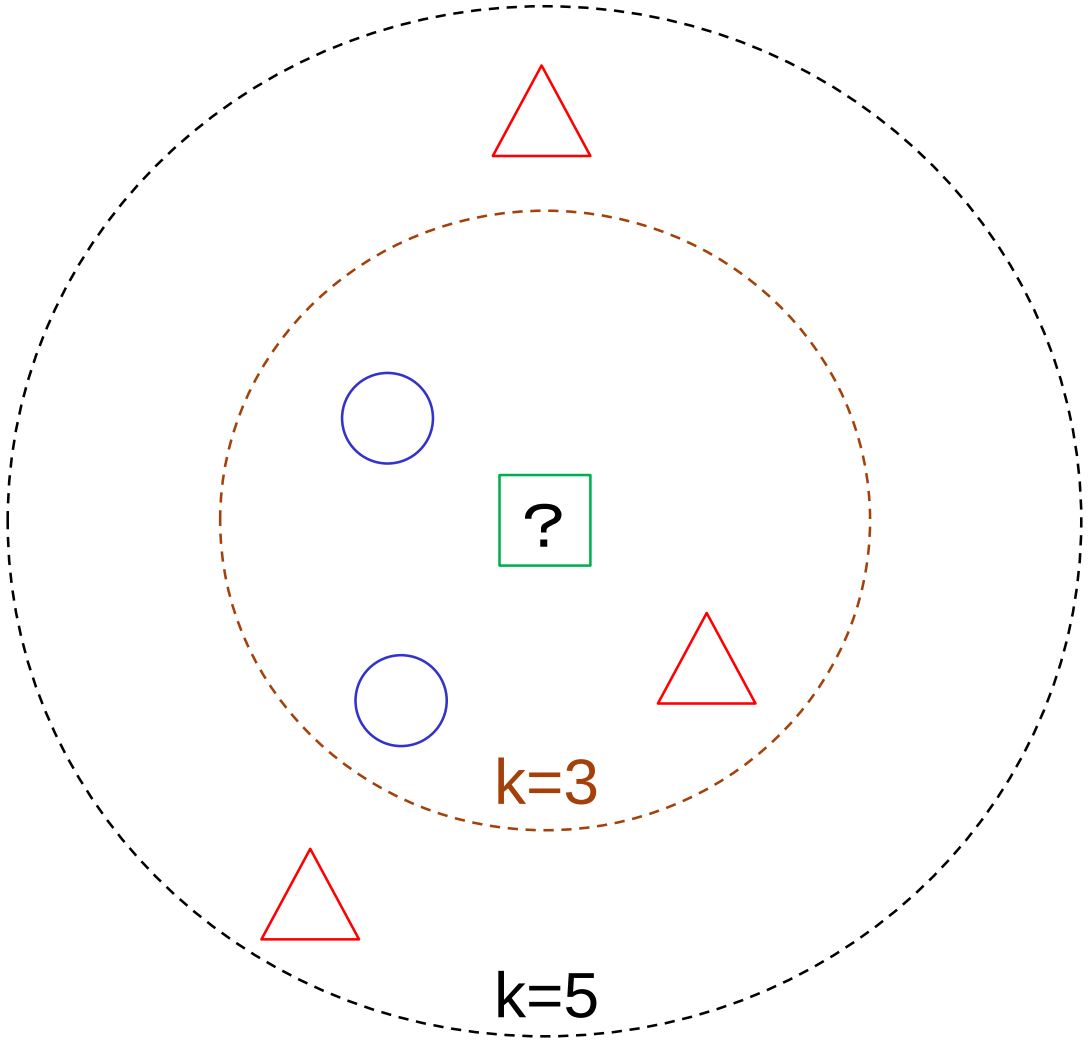
2次元

3次元

新たに軸を作る（次元を上げる）ことによって
直線（面）で分離できることがわかる

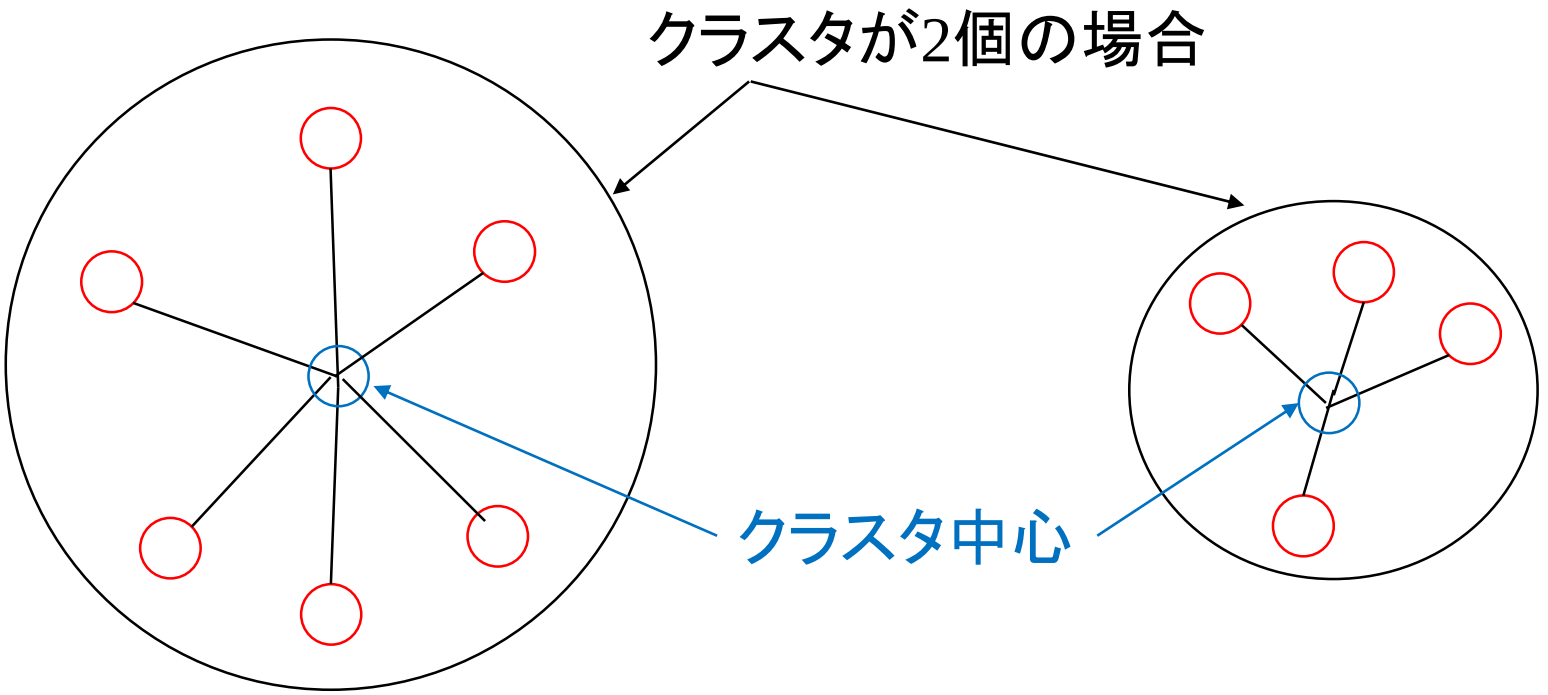
イメージでは「上の例のように変換して境界を引いて逆変換」となりますが
カーネルトリックはこれを数式上で行います

k-近傍法（k-NN）[3]



近くにあるk個のデータから多数決で属するクラスを決める
この図の場合、k=3なら○、k=5なら△になる

k-平均法（クラスタリング法）[2]



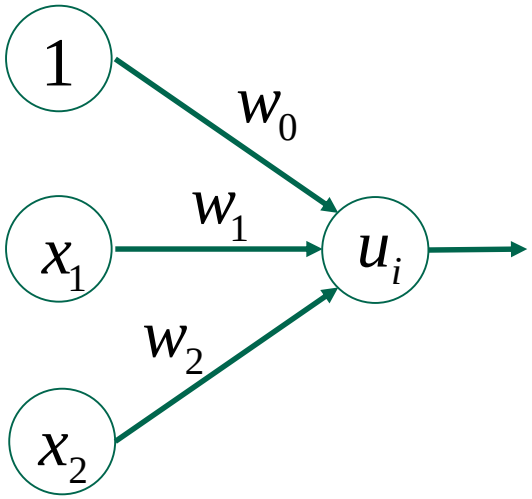
最終的に生成されるクラスタの数をユーザーが指定することができる

[各点に対してクラスタ中心からの距離を計算し、短い方のクラスタに所属させる
各クラスタ内の点からクラスタ中心を（算術平均などで）計算する] を繰り返す

点（事例）の分布に応じて適切なクラスを生成することができる手法として知られる

データ数が多いときは、クラスタが全て決まるまでに
何回も分類処理を繰り返す必要があり、処理時間がかかる

パーセプトロンと活性化関数 [3]



$$x = w_0 + w_1 x_1 + w_2 x_2$$ 重み付き和

※ $w_0 = b$

$$u_i = f(x)$$

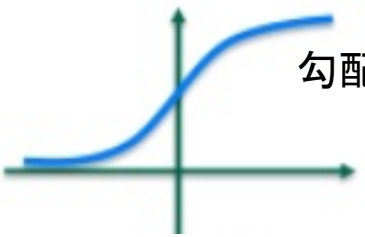
活性化関数（非線形関数）

$$f(x) = \frac{1}{1+e^{-x}}$$ シグモイド関数

ニューラルネットワークの計算で微分を用いることから計算が容易なシグモイド関数が多用された
しかし、0付近以外の勾配が小さく、誤差逆伝播法で更新する量が小さくなるため
学習がなかなか終わらずパーセプトロンの層数も増やすのが困難であった

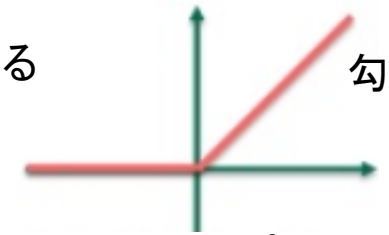
$$f(x) = \max(0, x)$$ ReLU（ランプ関数）

ReLU関数は微分しても値が変化せず、層数を増やすことが可能であった



勾配が小さくなる

シグモイド関数



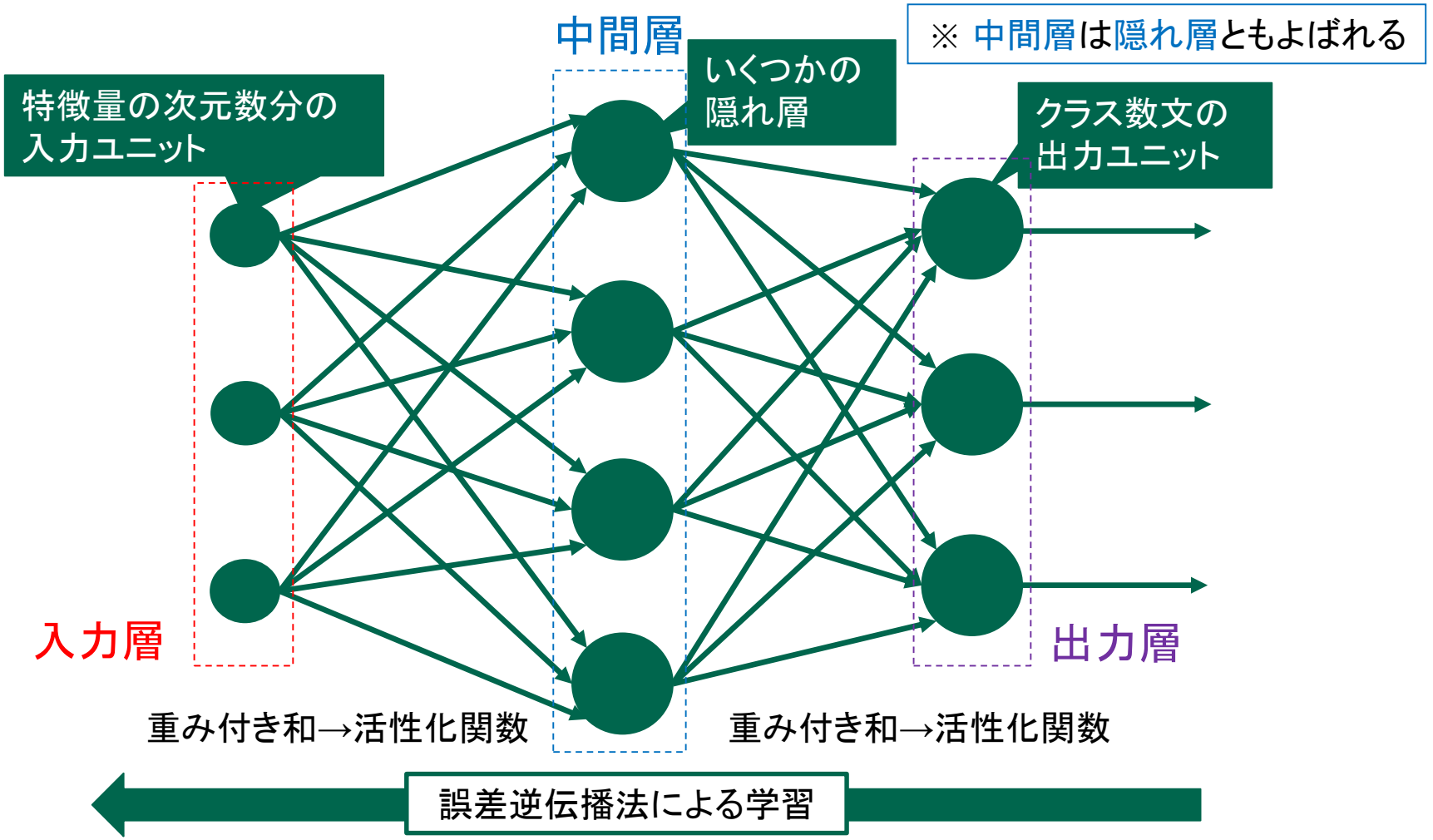
勾配一定

ReLU（ランプ関数）

他の重要な非線形関数
Softmax: 出力層で多出力の和が1
Tanh: DCGANの出力層
LeakyReLU: DCGANの識別器

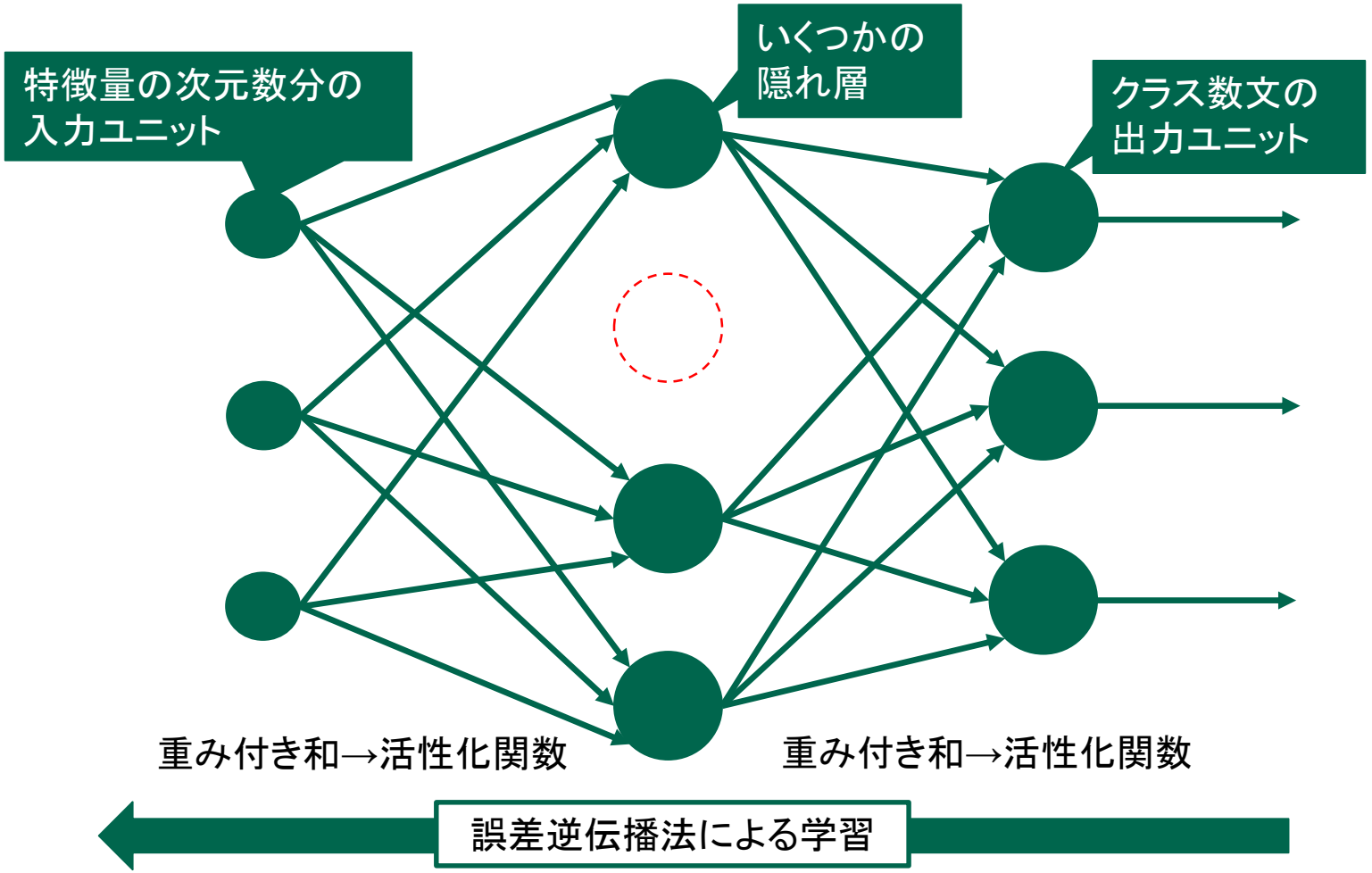
不必要なデータの忘却に用いられる
言語でのLSTMで使われている

ニューラルネットワーク（NN）[3]



パラメーターの更新（学習係数の更新）
SGD → AdaGrad → RMSProp → Adam の順で新しい方法が開発されてきた
一長一短があり、すべての問題で優れた手法はこれらの中に無い
SGDは現在でも使われ、Adamは研究者や技術者に好まれて使われている

ドロップアウト（Dropout）[3]

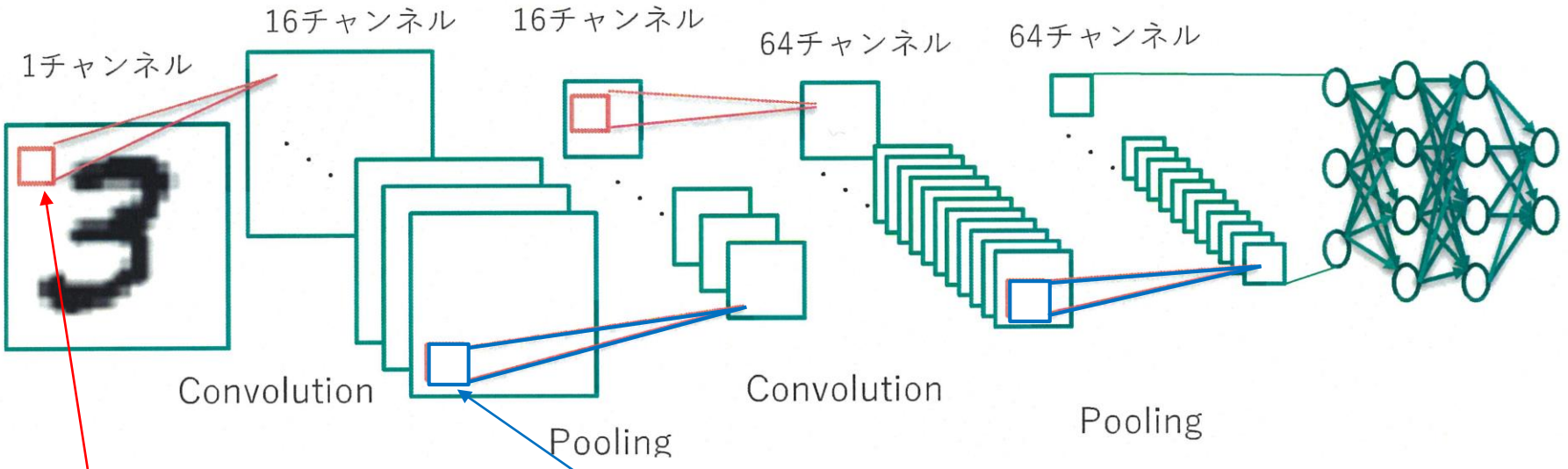


層が増えて複雑になるほど過学習（データに対してy=f(x)が複雑ということ）

学習時、ランダムに誤差伝播を0にする
未知のデータに関する分類の能力（汎化性能）を向上させる
（既存のデータに過剰に適合するのを防ぐ）

畳み込みニューラルネットワーク（CNN）[3]

Convolution層を持つニューラルネットワーク



局所的にのみ連結されたネットワーク
画像処理における畳み込みフィルタ（Convolution）に相当
例: 3x3のフィルタ

学会（2019年）でも
各層の節の数は
ケースバイケース

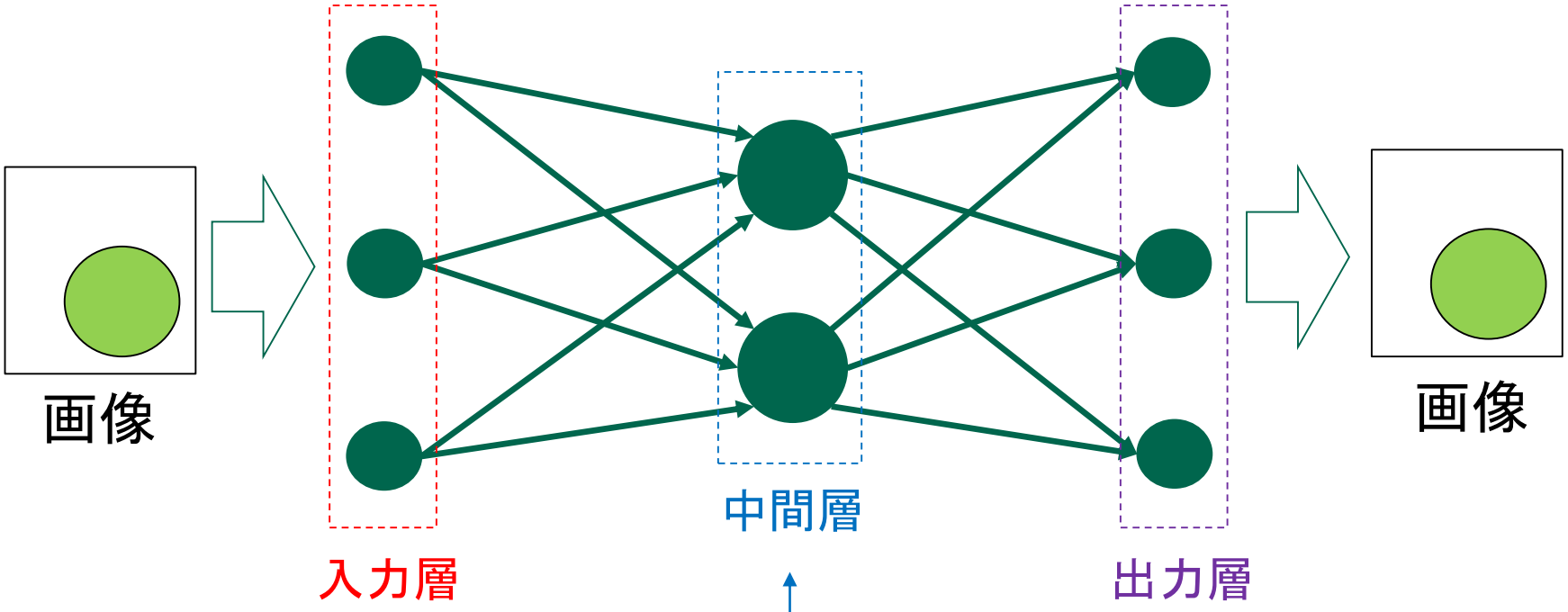
SEM像などでは手作業でフィルターをかけて
2値化やエッジ抽出して特徴量で分析
という手法を行う専門家の方もいる

位置ずれへの頑健性
Pooling: 局所領域の特徴を集約
例：3x3の領域内で最大値を返す

CNNは中間層が3～4層程度が学会（2019年）などでは多い
（CNNとは別の言語で用いられるニューラルネットワークのRNNでは10層程度が多い）

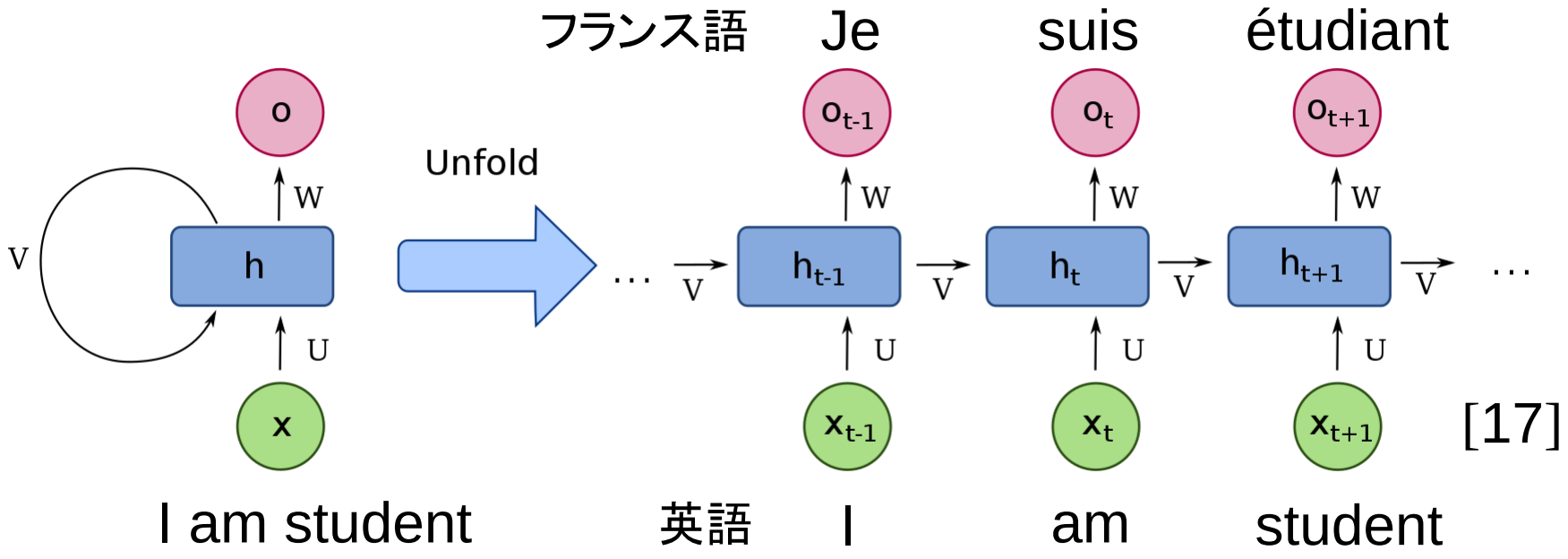
Auto-encoder [2]

出力 = 入力になるように学習する



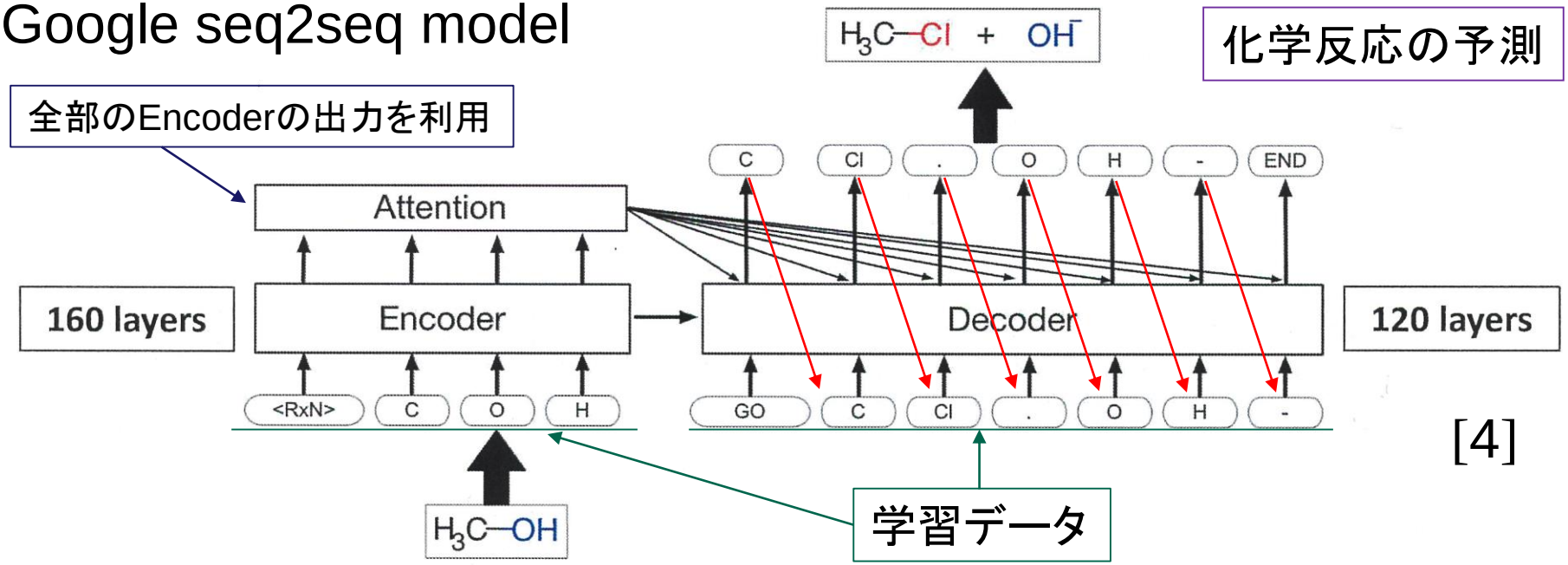
素子数の少ない中間層（隠れ層）に入力データの特徴が
圧縮・表現されていると考えられる

再帰型ニューラルネットワーク（RNN）[4, 17]



[17]

Google seq2seq model



[4]

敵対的生成ネットワーク（GANs）[4, 19, 20]

画像・音楽生成

ニューラルネットワーク（NN）

CNNを用いるなど
GANsを改良した
DCGANなどがある

Generator
（生成モデル）

Discriminator
（識別モデル）

一様乱数 → 偽物 → 偽物 → 偽物と判定

NN

NN

本物 → 本物と判定

交互に繰り返す

本物と偽者を判別できるように学習

一様乱数 → 偽物 → 偽物 → 本物との差

NN ← NN

Generatorを学習するとき、Discriminatorは学習させない

K-分割交差検証

対象データをK個に分割し、そのうちの1個をテストデータ
残りのk-1個を学習データとして学習する方法

例えば、K=5の場合

1回目	テスト	データ2	データ3	データ4	データ5
2回目	データ1	テスト	データ3	データ4	データ5
3回目	データ1	データ2	テスト	データ4	データ5
4回目	データ1	データ2	データ3	テスト	データ5
5回目	データ1	データ2	データ3	データ4	テスト

テストデータを変更しながらk回の学習と検証を繰り返す
このk回の検証結果を平均した結果を用いることが多い

Leave one out は K = 対象となるデータの数

分割交差検証ではScikit-learn だと cross_val_predict など
n_splits は10くらいが無難。ランダムの設定も忘れないように！
ランダムの設定を忘れると分割が上から順番に行われてツボにはまる

罰則項 [4]

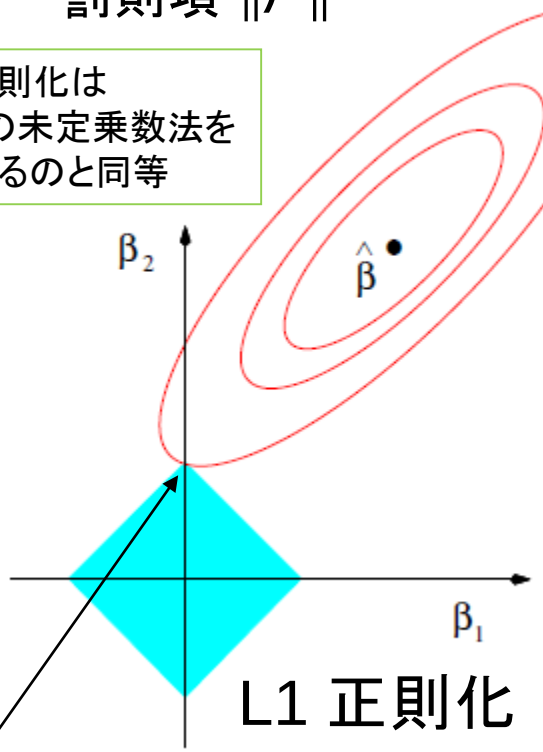
不自然な解をなるべく避けるために計算式に[λ×罰則項]が加算される

λは任意の正の定数（ハイパーパラメータとよばれる）

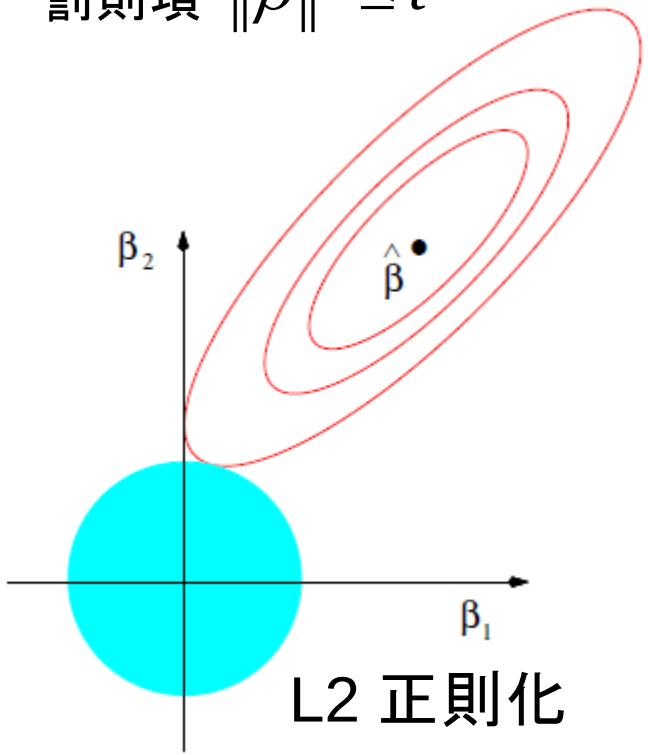
罰則項 $\|\beta\| \leq t$

罰則項 $\|\beta\|^2 \leq t$

L1正則化は
ラグランジュの未定乗数法を
解いているのと同等



L1 正則化



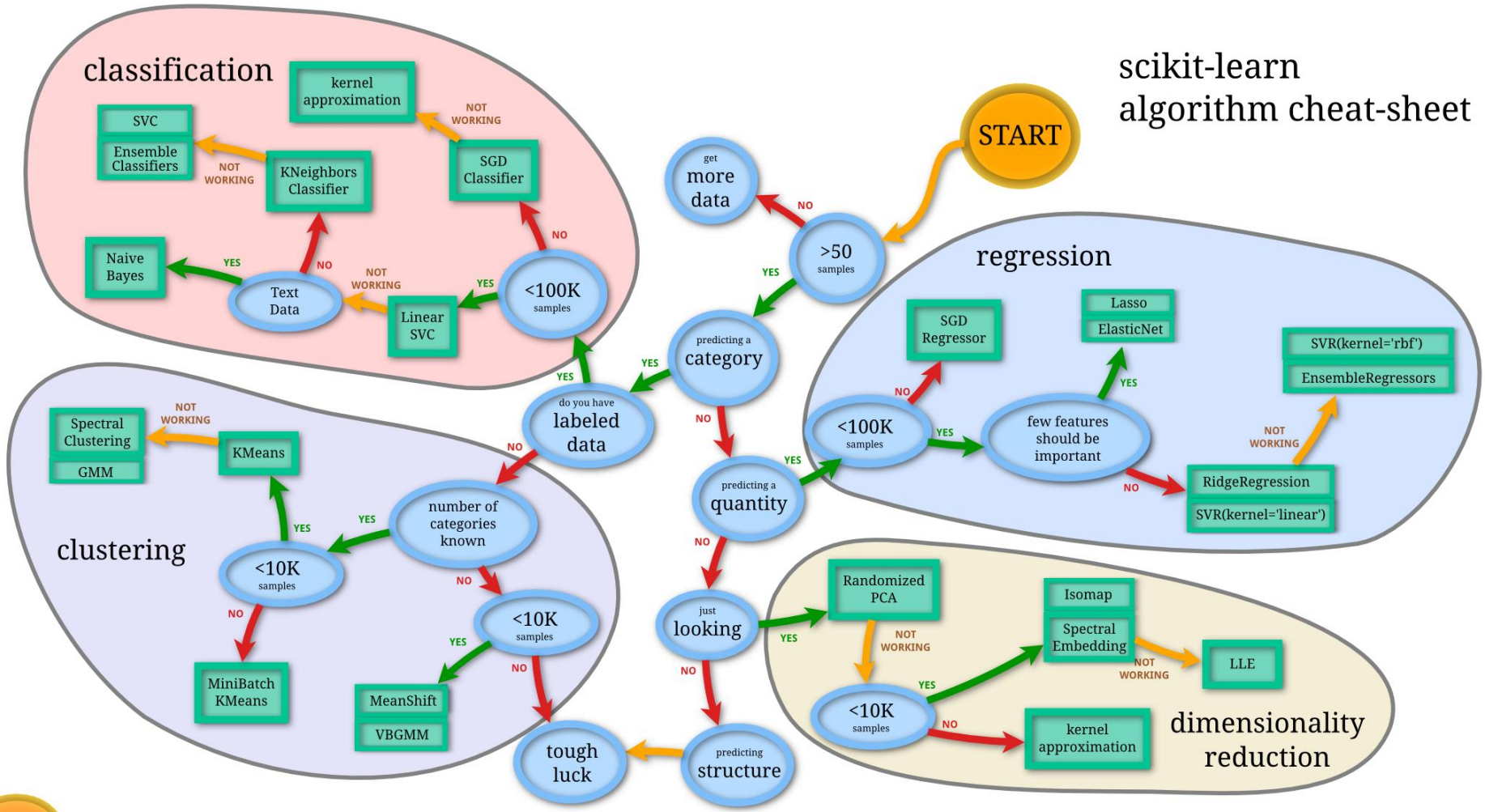
L2 正則化

https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf

図のように、L1正則化ではパラメーター（β_2）が0により近いところで接するため
L2正則化よりも0になるパラメータが多い

グリッドサーチ（GridSearchCV）を用いてハイパーパラメータや他のパラメータを
等間隔に区切って最適な値を調べていくことが行われる

各種方法の使い分け



https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

どのような手法を使えば良いかが書かれている

Classification: 分類問題、判別分析のアルゴリズム
Regression: 回帰分析のアルゴリズム（誤差が最小になるようにしている）
Clustering: クラスタリングのアルゴリズム

各種方法（回帰）の特徴 [4, 24]

□ 線形の回帰分析
・f(x)=wx+bとおいて最小二乗法でパラメータを求める
　誤差の二乗和が小さくなるように計算する
・正則化項（上記の方法でで上手くいかない場合の補正）
　リッジ回帰：パラメータの二乗和を加算。二次の正則化項を付けてパラメータが多い問題を回避している。Gaussian Process回帰とあまり変わらない結果になる
　LASSO: パラメータの絶対値の和を加算。パラメータをなるべく多く0にすることから変数選択の機能があるとされている

□ 回帰木（条件分岐のためf(x)は不連続な関数になる）
・パラメーターは各分岐において誤差が小さくなるように決定される
・入力データが高次元となってもあまり影響を受けない

□ ニューラルネットワーク（そこそこ精度が出る）
・誤差が最も少なるなるようにパラメータを調整
・組み合わせが多くなるほどパラメータの調整は難しい

□ Gaussian Process（カーネル法で非線形化も可能）

・予測の信頼度を示す確率分布が計算可能	カーネル関数の決め方が難しい
・確率が最も高い分布を予測値とする	逆行列があるため計算が重い
・ガウス分布同士の相関から新たな点を予測可能	精度を出すのが難しい

□ サポートベクター回帰
・ヒンジ関数を用いてカーネル法を適用すると計算時間を短縮できる

事例ベース学習の種類と手法 [2]

□ 事例ベース学習の種類

・教師あり学習
人間が用意した正解が付与されたデータを用いる

・教師なし学習
正解がないデータを用いる

強化学習
ときどき得られる（時間遅れのある）
報酬に基づいて自らの行動の
学習・最適化を行う枠組み
（例えば、将棋、迷路など）

・半教師あり学習
人工知能（AI）が判断が難しいと判定したものを人間が判断する、など

□ 事例ベース学習の方法（下記は代表的なもの）

・教師あり学習
サポートベクターマシン（SVM）
決定木
アンサンブル学習（AdaBoost、ランダムフォレスト）
ニューラルネットワーク：ディープラーニング（深層学習）はこの一種

・教師なし学習
クラスタリング法（k-平均法など）
自己組織化マップ（SOM）：NNの一種
　入力データの特徴を反映したマップを作成し、データがマップ上のどの位置に出力されたかによって、どのデータと類似した特徴を持つかを視覚的に理解できる

機械学習用のツール [3, 4]

□ エディタ
- Jupyter（ジュピター）
  ブラウザ上でIPythonを利用できる。グラフも描画できる
  Jupyter Notebook がオススメ
- Spyder（スパイダー）
  Matlabライクな開発環境。IPythonの画面も表示できる
- IPython（アイパイソン）
  シェルのように使える（コマンドプロンプトが好きな人向き）
- IDLE（アイドル）
  付属のエディタでコードを書き、F5キーで実行できる

□ 記述子
- RDKit
  構造式からの記述子の生成など
- XenonPy
  記述子の可視化など

□ 表
- Pandas
  表を扱う場合

□ ディープラーニング(DL)用のツール（ライブラリ）
- TensorFlow（テンソルフロー）: 開発元 Google（米国）
- Chainer（チェイナー）: 開発元 Preferred Networks（日本）
- Caffe（カフェ）: 開発元 UCバークレイのBVLC（米国）, C++（言語）でも可能
- PyTorch（パイトーチ）: プロ向け（Caffe → Caffe2 → PyTorch）

※ 基本的にOSはLinuxのUbuntuが推奨される。基本的に言語はpythonとなる

□ その他のツール
- scikit-learn（サイキットラーン）: 機械学習用のパッケージ（一番メジャー）
- Theano（テアノ）: DLの各種アルゴリズムがソースコード付きで解説されている
- Keras（ケラス）: TensorFlow と Theano のためのDL用ライブラリ（初心者向け）

[Jupyter Notebook + scikit-learn], [Jupyter Notebook + Keras + TensorFlow] など

機械学習の実際（scikit-learn）[3]

```
import numpy as np # 数値計算ライブラリ
from sklearn.datasets import load_digits # sklearnにある読み込みデータの指定
from sklearn.model_selection import train_test_split as spl # 訓練データとテストデータに分割の指定
from sklearn.metrics import confusion_matrix as cfm # 混同行列

from sklearn.svm import LinearSVC as classifier # 分類器(LinearSVM)の指定
clf = classifier(C=1.0) # 分類器(LinearSVM)の指定とパラメータの指定

digits = load_digits() # データの読み込み
x_train, x_test, y_train, y_test = spl(digits.data, digits.target, test_size=0.2, random_state=0)  # 訓練データとテストデータに分割
clf.fit(x_train, y_train) # 訓練データで学習
clf.score(x_test, y_test) # テストデータの評価

y_pred = clf.predict(x_test) # 予測
cfm(y_test, y_pred) # 混同行列で評価
```

たった十数行で機械学習ができてしまう！
文献[3]も試してみよう！

該当する部分を下記のように1から2行ほど書き換えれば様々な手法（識別器）が利用できる

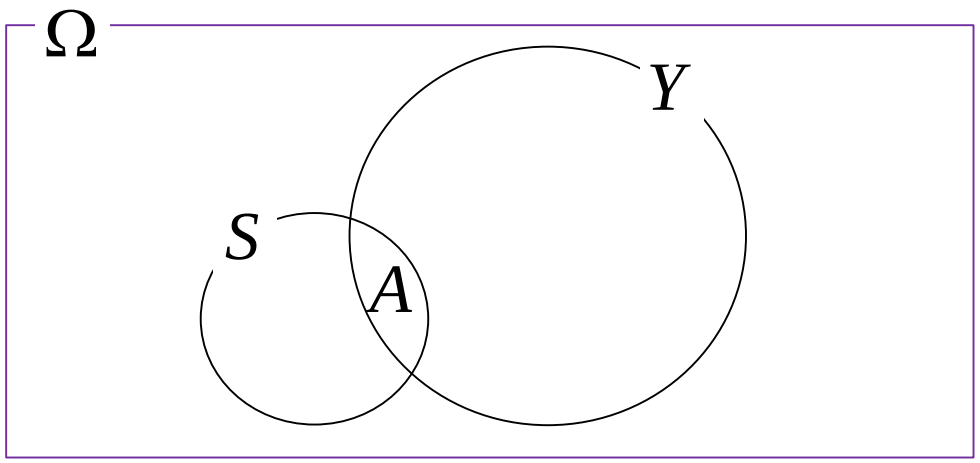
・ランダムフォレスト

```
from sklearn.ensemble import RandomForestClassifier as classifier
clf = classifier() # 分類器の指定とパラメータの指定(ディフォルトの設定を使用)
```

・k近傍法（※ k平均法とは異なるので注意）

```
from sklearn.neighbors import KNeighborsClassifier as classifier
clf = classifier() # 分類器の指定とパラメータの指定(ディフォルトの設定を使用)
```

ベイズの定理とベイズ推定 [4]



Sが起きたときに
Yが起こる確率 $P(Y|S)=\frac{P(A)}{P(S)}$

Yが起きたときに
Sが起こる確率 $P(S|Y)=\frac{P(A)}{P(Y)}$

$$P(A)=P(Y|S)P(S)=P(S|Y)P(Y)$$

ベイズの定理

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

この場合、P(Y)は
規格化定数の意味しか
持たないので省略

$$S \sim P(S|Y) \propto P(Y|S)\times P(S)$$

高い確率で
予測された構造

Yの性能を示す
Sの構造の確率

構造Sと性能Yのデータ

多くの構造と性能のデータを用いて、高い性能を示す構造を推定する

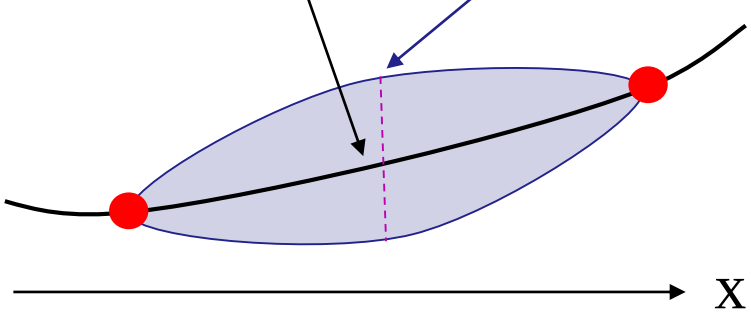
ベイズ最適化 [4, 16]

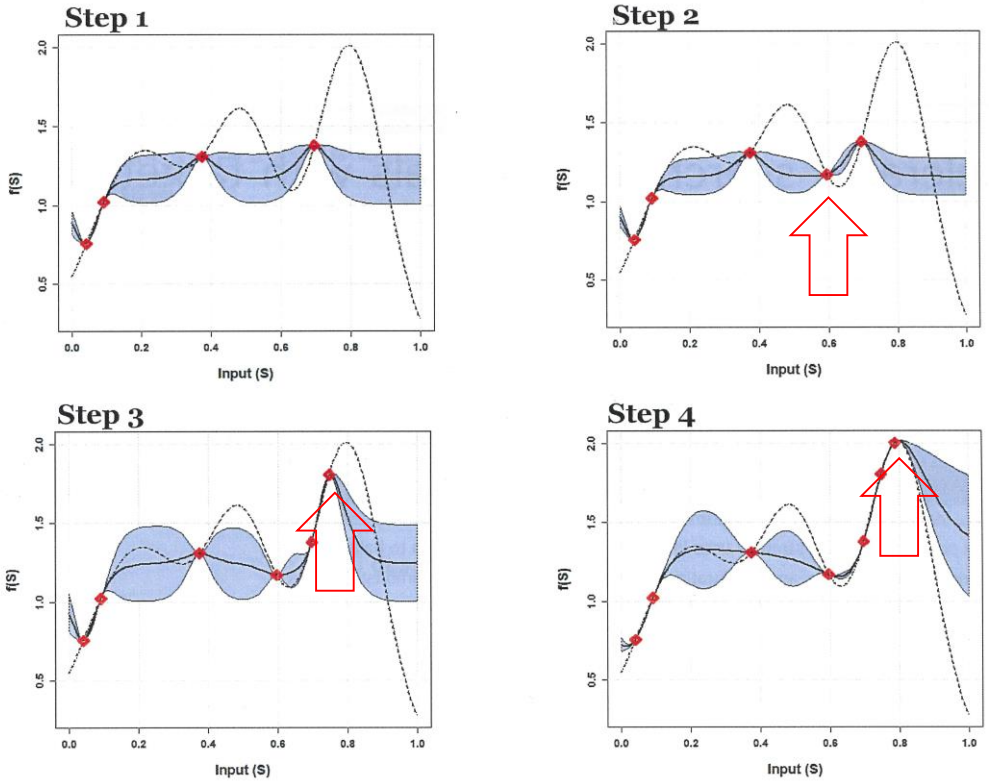
【活用】回帰関数から最大値が存在しそうな領域を選択
【探索】回帰関数の推定精度を上げるために分散が高い領域を選択

【実験計画】
平均と分散から獲得関数$a_n(x)$を導出
次に実験するxを決定

UCB (upper-confidence bound) criterion

$$a_n(x) = \underbrace{\mu_n(x)}_{\text{活用}} + \underbrace{\kappa\sigma_n(x)}_{\text{探索}}$$





■ 多くの場合、回帰モデルはガウス過程
■ 獲得関数のバリエーション
Expected Improvement, Probability of Improvement, Thompson Sampling など
■ R言語: rBayesianOptimization

機械学習における計算時間の考え方

CPUとGPU

ディープラーニングは16ビットの半精度浮動小数点でも問題なく学習ができることが分かっている[S1]

※ 機械学習：GPUの方がCPUよりも速い

16ビットのGPUは32ビットのGPUよりも2倍速い

GoogleはTensorFlow用に8ビット演算が可能なTPUを開発している
（8ビットは16ビットよりもさらに演算速度の高速化が期待される）

PCの構成（約18万円、2016年7月時点）[1]

名称	構成	カスタマイズ
マザーボード	インテルH170チップセット ATX （映像出力端子付き）	
メモリ	32 GB(DR4 SDRAM 16GBx2)	8 GBから32 GBへ変更
電源	700 W	500 Wから700 Wへ変更
CPU	インテル Core-i7 6700	
SSD	250 GB	
HDD	1TB (SATA3)	
光学ドライブ	DVDスーパーマルチドライブ	
GPU	NVIDIA GeForce GTX1070 8 GB	
LAN	ギガビットLANポート（オンボード）	
OS	Windows 10 HOME 64 bit	新しいHDDに変えて Ubuntu Desktopを インストール

[S1] S. Gupta et al., Deep learning with limited numerical precision. CoRR, abs/1502.02551 392(2015).

機械学習で分かること

内挿と外挿 [4]

- データ科学は基本的に内挿しかできない

- 内挿範囲か外挿範囲かを明らかにする方法はわからない
  新しいデータを入れてみて、それが上手くいくかを調べて評価するしかない

- 目的の性能が外挿範囲にある場合、下記の2の事例が起こることがある
  1) 外挿範囲に近い内挿範囲の端に予測が集まる
  2) もし、外挿範囲に予測がされたとしても目的の性能が出ない
  （予測された結果に対して、第一原理計算や実験をしてデータを追加し、またモデルを作って予測するというサイクルを繰り返して、少ないステップで目的の性能が得られるようにしていくことが必要となる）

- 失敗のデータは、その失敗の範囲を避けるために必要
  成功のデータから成功の範囲外を失敗として失敗のデータを作ったりする

- 偏りのあるデータは、偏った結果しか生み出さない

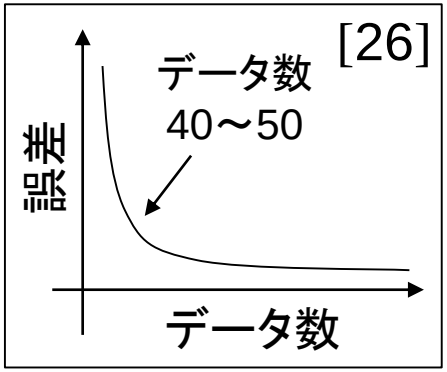
機械学習によって得られた値が
内挿か外挿であるかをよく検討することが必要

機械学習で重要な点 [4]

- 機械学習で得られたモデルは、理解したいがそれは難しい
- 同じ予測をするために様々な解がある（多様性がある）
- 統計学の共通原理：汎化性能の高いモデルが最良
  同等の汎化性能を達成するモデルは数多く存在する
- 機械学習の有用性：（複雑な）経験則の発見（因果関係でない）
- 欠損にバイアスがあるとき、穴埋めしてはいけない
  値を変えて結果を検証することが必要
- 似たような変数はロバスト性のために重要となる
  独立性の高い入力変数だけにすればよいというわけではない
- ニューラルネットワーク（NN）のデータ数の目安はない
  ノード（節）の数や層の数も最適なものが分からない
  自動でそれらの値やパラメータを変えて、最適なものを探すプログラムを作る必要がある

機械学習で重要な点 [24, 25]

- y=f(x)の関数フィッティング（統計処理であったりする）
- y=f(x)の複雑さによってデータ数は異なる
- 多くの場合、数百から数千件のデータが必要



- “データ数は同類の問題を解いた経験“から経験的に見積もる
- データにあるが気づいていないことを知る手法としての利用もある
- 人が行うことが困難な大量のデータや試行回数、試行時間などを処理できる

◇ マーカス理論の式をデータから機械学習で再現しようとした場合 [26]

- 誤差はデータ数に対して指数関数的に減少
- 記述子が良ければデータ数は40程度から良い

Marcus theory (Wikipedia)

$$\Delta G^{\ddagger} = \frac{(\lambda_o + \Delta G^0)^2}{4\lambda_o}$$

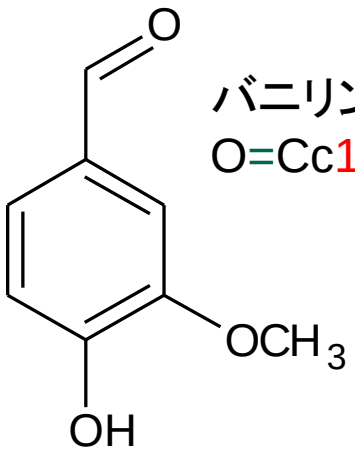
- 物理や理論にもとづいて記述を書くと良くなる（原子の構造情報は重要）

第一原理計算との連携

SMILESによる化学構造の表現 [4]

Simplified Molecular input Line Entry Specification Syntax
(http://www.daylight.com/smiles/incex.html)

- 環の始点と終点は同じ数字
- 側鎖は括弧
- C, N, O, P, S, Br, Cl, I以外の元素は角括弧
  例えば [Au]
- 芳香環は小文字の元素記号
- = 二重結合
- # 三重結合
- @ 絶対配置



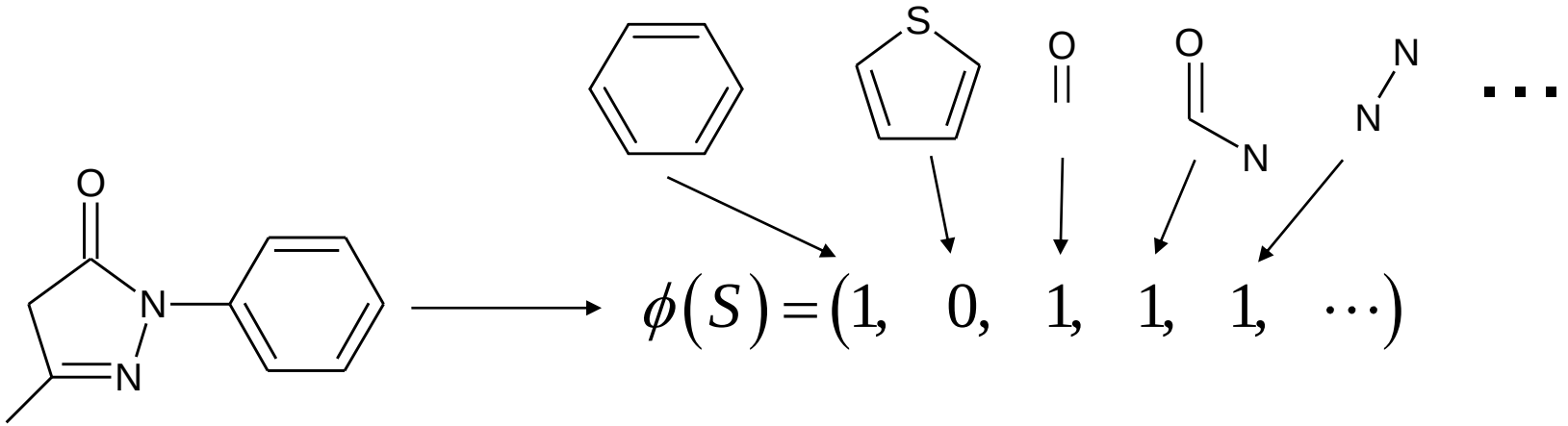
バニリン（vanillin） $C_8H_8O_3$
O=Cc1ccc(O)c(OC)c1

- SMILES表記法に基づき 化学反応の入出力を文字列で表現

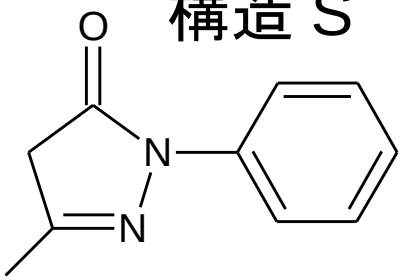
Input			Output
Reactant 1	Reactant 2	Catalyst	Product
Nc1nnc(C2CCC2)s1	. O=C(Cl)Cl	> A_CCOC(C)=O	O=C=Nc1nnc(C2CCC2)s1

- 機械翻訳用のニューラルネットワークを訓練し、高精度な反応予測モデルを導く
  Sequence-to-Sequence (seq2seq)
  反応中心部位の予測精度は～90%
  生成物ペアの反応の有無を予測するモデル: Accuracy = 86.4%

有機材料での入力データ（descriptor）の例 [4]



構造 S



記述子（descriptor）

1011110000001101001

物性 Y

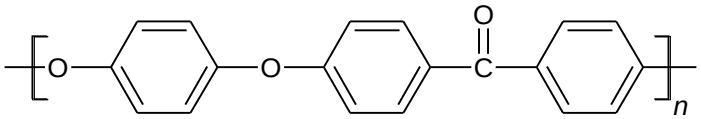
生理活性
毒性
バンドギャップ
エネルギー
など

Molecular fingerprint
PubChem
MDL MACCS keys
ECFP6
Kolekota-Roth
など

SVM
ランダムフォレスト
ディープラーニング
Gradient Boosting
Elastic Net
など

有機材料での入力データ（descriptor）の例 [4]

構造



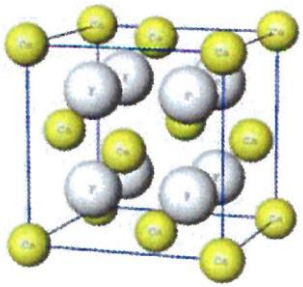
組成　$A_aB_bC_cD_dE_e$

記述子（descriptor）　$\phi_i(S) = f\left(a,b,c,d,e,\phi_A^i,\phi_B^i,\phi_C^i,\phi_D^i,\phi_E^i\right)$

平均、分散、最大、最小など　組成比　原子番号、原子半径など

無機材料での入力データ（descriptor）の例 [4]

Crystal structure



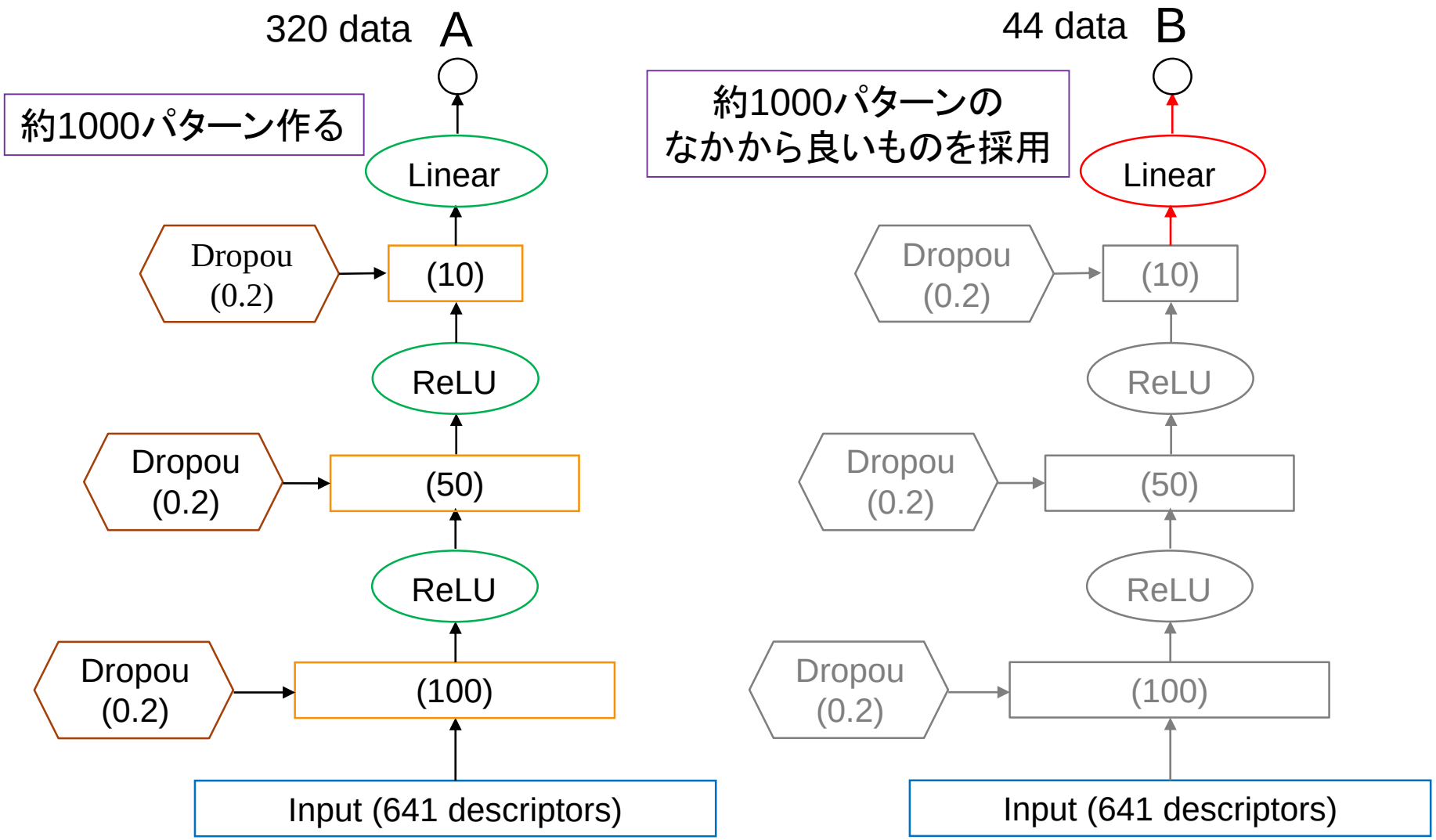
Composition
$Ni_aTi_bCu_cFe_dPd_e$

$$\phi_i(S) = f\left(a,b,c,d,e,\phi_{Ni}^i,\phi_{Ti}^i,\phi_{Cu}^i,\phi_{Fe}^i,\phi_{Pd}^i\right)$$

組成比　原子番号、原子半径など

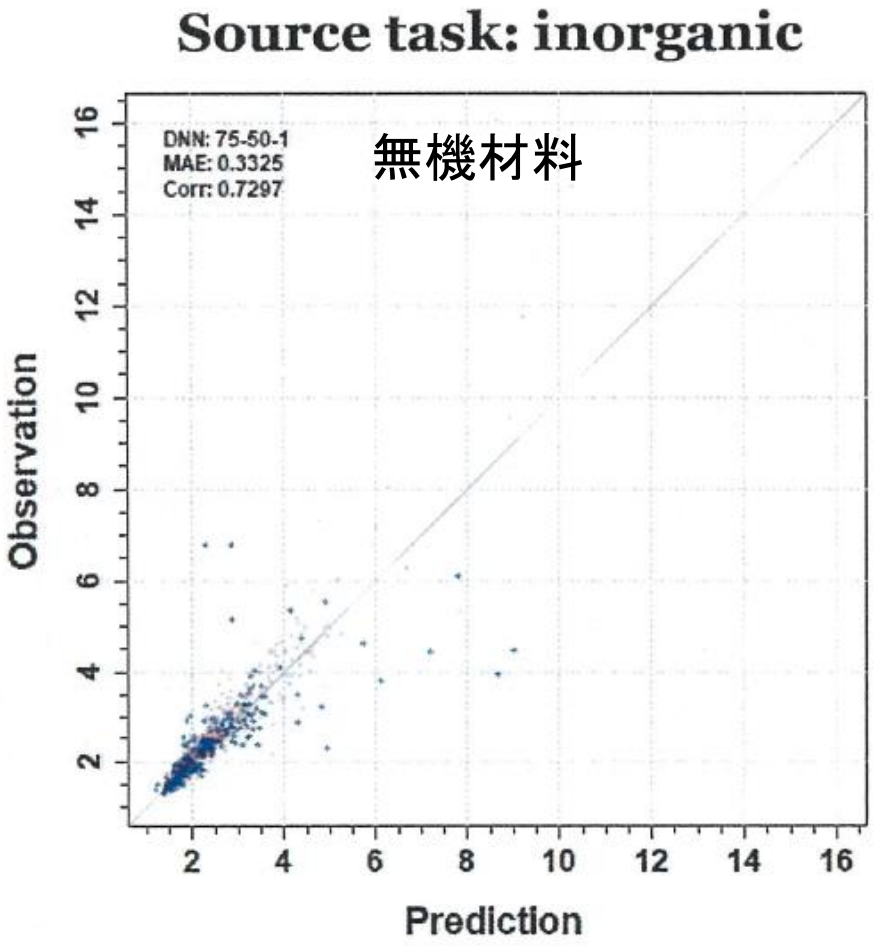
平均、分散、最大、最小など

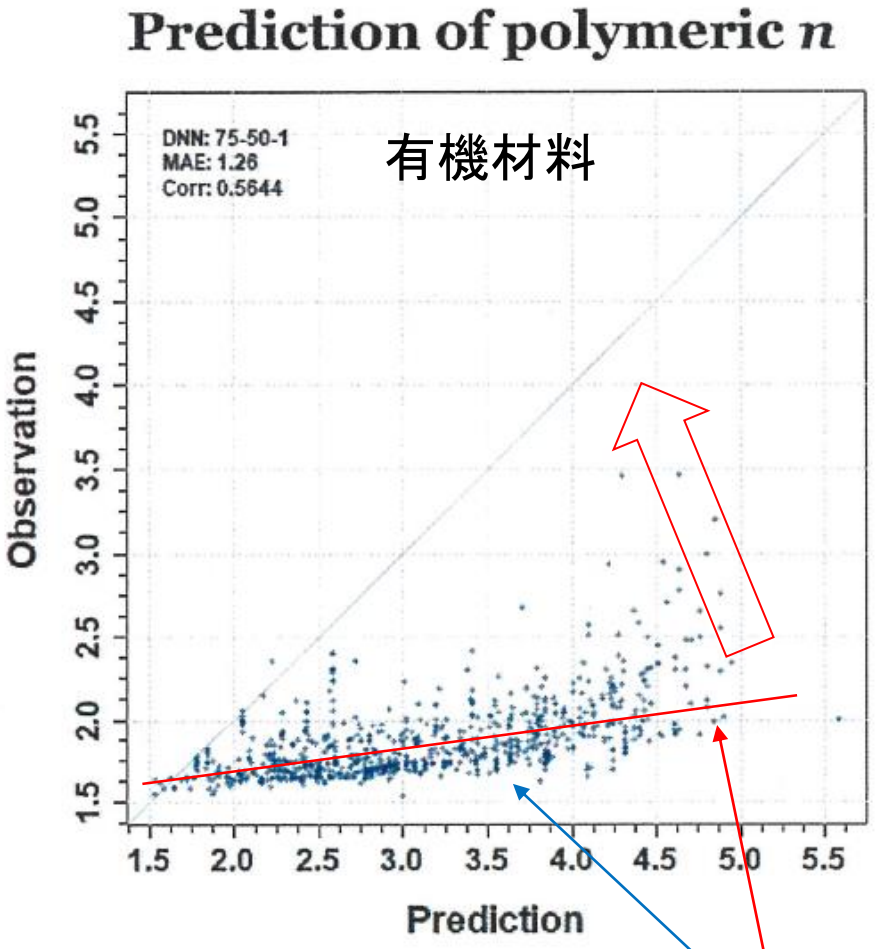
転移学習 [4]



物理量AとBに比例関係がある場合
多くのデータがある物理量Aで学習させ、それを物理量Bの予測に利用する

無機材料で学習したときの有機材料の予測 [4]



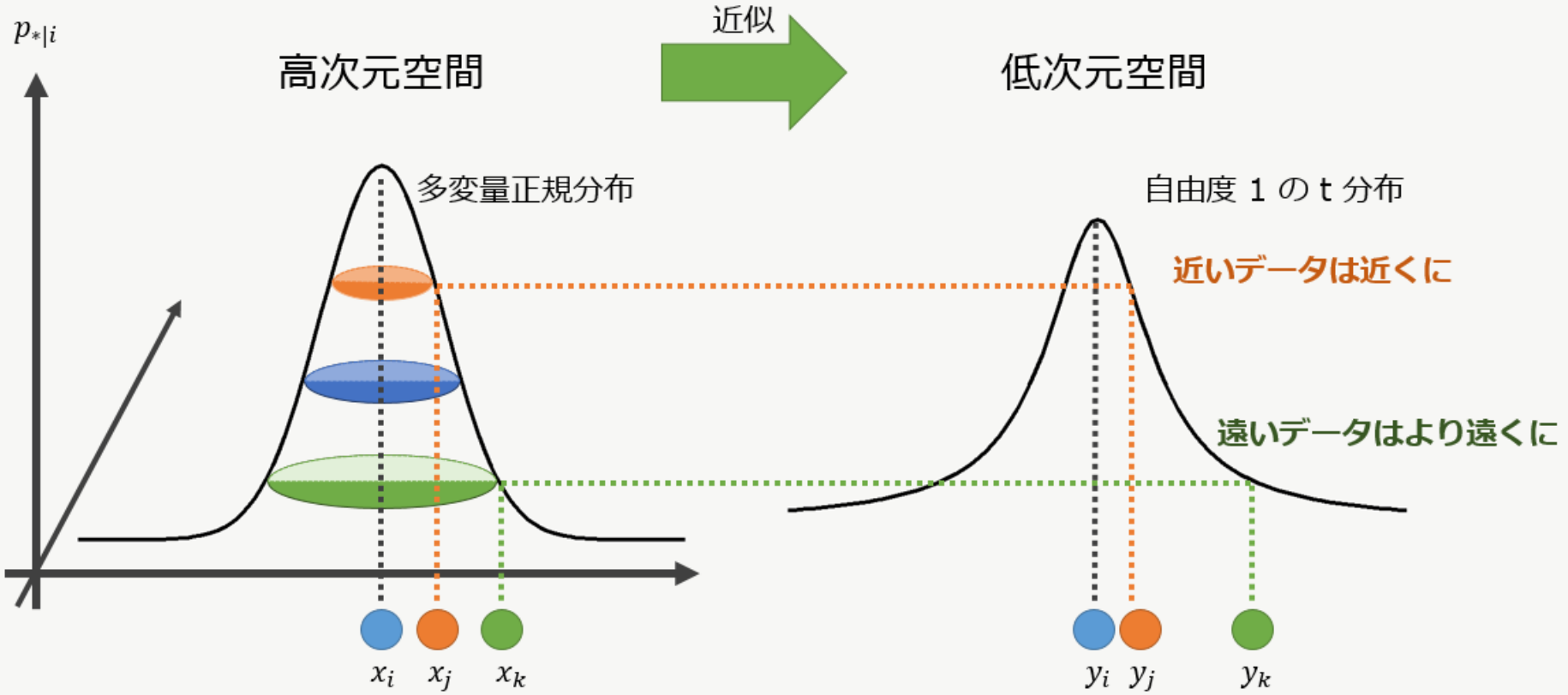


無機材料で学習したものをそのまま有機材料に適用しても上手く予測できない

傾きを補正すれば比較的上手く予測できる
（どのような理由でこのような結果になっているかは現時点（2018年）ではわからない）

t-SNE (t-distributed Stochastic Neighbor Embedding) [4]

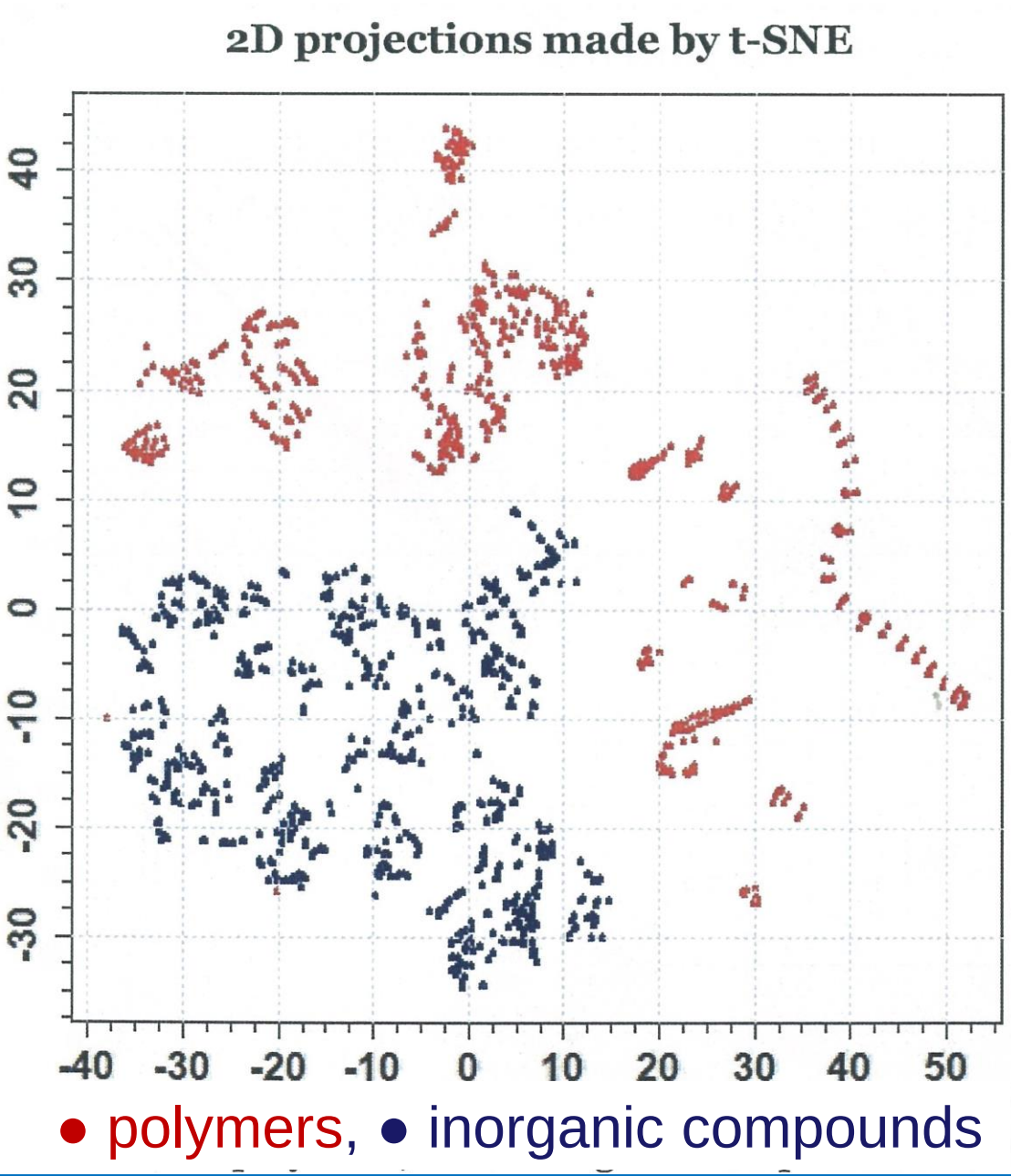
Van der Maaten and Hinton, JMLR, 2018



https://blog.albert2005.co.jp/2015/12/02/tsne/

2点間の「近さ」を確率分布で表現する

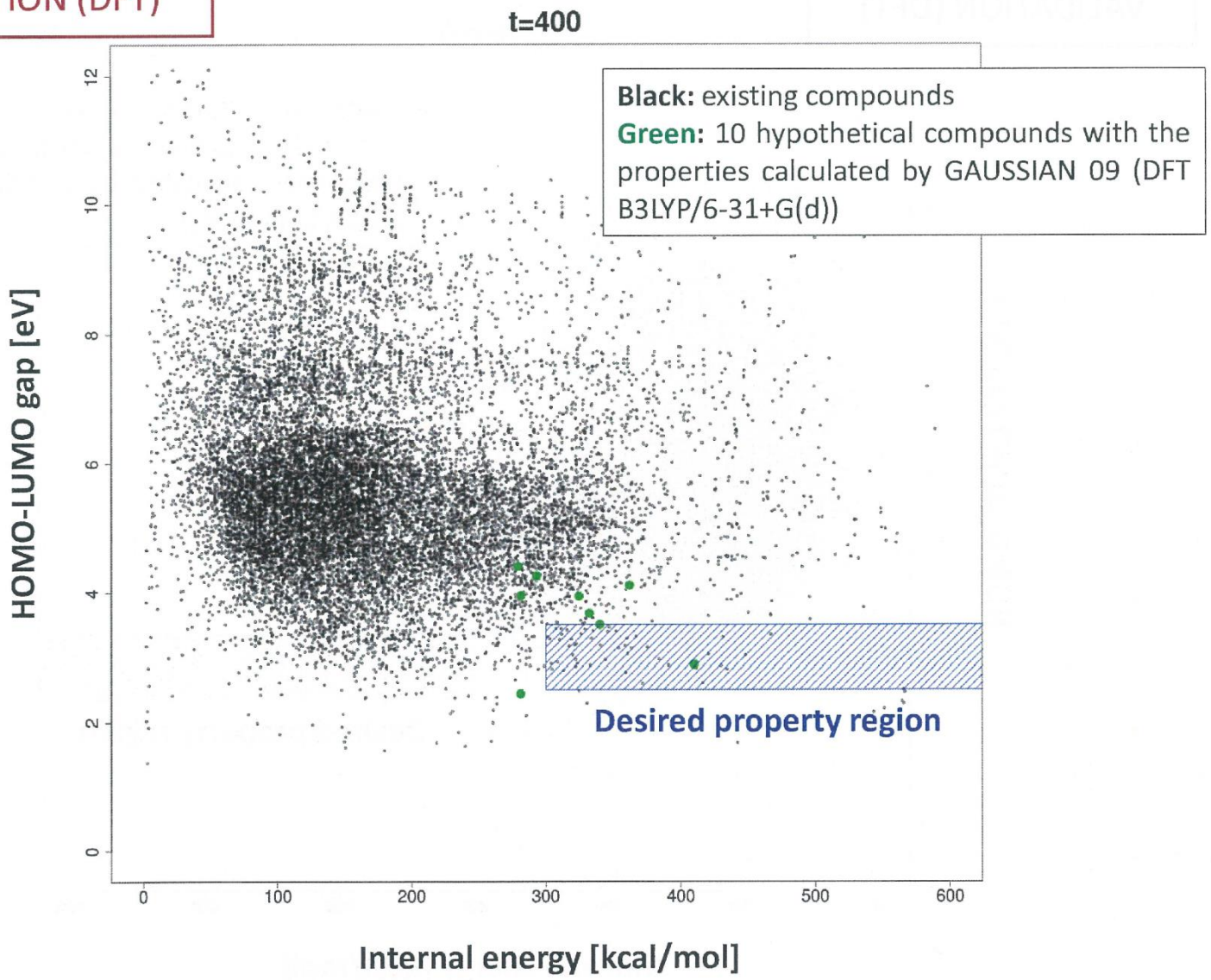
t-SNEがメジャー。この他には自己組織化マップがある



ポリマーのデータは、無機化合物のデータに関して外挿領域にあることがわかる

内挿的予測の限界を超える [4]

VALIDATION (DFT)



機械学習による予測（構造）と第一原理計算を繰り返すことで目標に近づく

データベース

◇ 構造や基礎的な特性に関するデータベース (計算も含む)

・COD (Crystallography Open Database)：結晶構造のデータ

・AMCSD：結晶構造のデータ

・Citrination：実験や計算のデータ

・ESP：電子構造

・MRL Datamining Chart：熱電特性

・CCCBDB：実験および計算での気相原子および小分子の熱化学データ

◇ 第一原理計算によるデータベース

・Material Project：VASPでのデータ（フォノンはAbinit, X線吸収はFEFF）

・AFLOW：VASPやQuantum Espressoのデータ

・OQMD（オーキューエムディー）：VASPのデータ

・NoMaD（ノマド）：様々な第一原理計算コードのデータ

・PubChemQC (http://pubchemqc.riken.jp/)：HOMO-LUMO gap

・TE Desing Lab：熱電特性

・Catalysis-Hub (https://www.catalysis-hub.org/)：触媒研究用

・COMPUTATIONAL MATERIALS REPOSITORY (RMC)

◇ データを自作する場合の例（物理系）[26]

・ USPEX + VASP (PBE-GGA, HSE06, GW_0@PBE) & VASP + Phononpy

欠陥を含む場合はsxdefectalign, PyCDTなどを用いた計算

◇ 文献からのデータ抽出 [22]
・ChemDataExtractor
(https://pubs.acs.org/doi/10.1021/acs.jcim.6b00207)
(https://pubs.acs.org/doi/10.1021/acs.chemrev.6b00851)

◇ 記述子の例 [22]
・fingerprint: NMR, IRに有効とされる
・原子間の結合をグラフとして取り扱う方法
(https://www.sciencedirect.com/science/article/pii/S135964461200275 9)
(https://pubs.acs.org/doi/abs/10.1021/ci400403w)
・各原子の記述に周囲の原子の影響の重みを含めて取り扱う方法
・原子配置の3次元構造情報を取り入れる方法
(https://pubs.acs.org/doi/abs/10.1021/acs.jmedchem.7b00696)
などがある

◇ 反応中の構造と安定性の変化をほぼ自動で調べ上げる方法 [22]
・人工力誘起反応（Artificial Force Induced Reaction: AFIR）法
(https://pubs.rsc.org/en/content/articlelanding/2013/cp/c3cp44063j#!divAbstract)

◇ 原子間ポテンシャル（https://www.jsme.or.jp/kikainenkan2019/chap05/）[23]
・Behler-Parrinello neural networks

Descriptors & ML algorithms in R [4]

Regression and classification

Method	Package
Elastic-net	glmnet
Deep Learning	h2o, mxnet
Support Vector Machine	kernlab
Gaussian Process regression	GPfit, laGP
Boosting	Boost
Bagging	ipred
Random Forest	randomForest, ranger, Rborist
Gradient Boosting	xgboost

Hyperparameter optimization

Method	Package
Bayesian Optimization	rBayesianOptimization

Descriptors in the *rcdk* package (*iqspr*)

Name	Description
standard	Paths of a given length
extended	standard + ring + atomic property
maccs	166 bits MDL MACCS keys
circular	ECFP6
pubchem	881 bits PubChem fingerprint
graph	standard + connectivity
kr	4860 bits Kolekota-Roth
hybridization	standard + Info. on hybridization
shortestpath	shortest paths between atoms
signature	count type of fingerprint

+ more than 200 physical and chemical descriptors

Materials Property Database

Green et al., Appl. Phys. Rev. 4, 011105 (2017)

TABLE III. Partial list of computational and/or experimental materials property databases.

Name	Owner	Content	Fee? (Y/N)
Aerospace Structural Metals Database (ASMD)[136]	CINDAS LLC	Properties of 255+ high strength, lightweight alloys	N
Automated Interactive Infrastructure and Database for Computational Science (AiiDA)[137]	École Polytechnique Fédérale de Lausanne (Switzerland) The Bosch Research and Technology Center (USA)	Informatics infrastructure to manage, preserve, and disseminate the simulations, data, and workflows of computational science.	N
ASM Online Databases[138]	ASM International	Metals and alloys databases (e.g., phase diagrams, mechanical properties, corrosion, and micrographs)	Y
Automatic Flow for Materials Discovery (AFLOW)[139]	Duke University	Property data for ~1 500 000 materials, as well as ~150 000 000 calculated properties	N
Citrination[140]	Citrine	Structure-property-process relationships for over 17 million materials	N
CRC Handbook of Materials Properties[141]	CRC Press	Comprehensive physical and structural properties data for engineering materials	Y
Crystallography Open Database (COD)[142]	Materials Design	Open-access collection of ~300 000 crystal structures of organic, inorganic, metal-organic compounds and minerals	N
Electrolytic Genome[143]	Joint Center for Energy Storage Research (DOE)	Project to accelerate the discovery of battery electrolytes by computer simulation and experimental validation	N
Granta Data Series[144]	Granta	Extensive catalog of property data for a wide range of materials classes	Y
Hydrogen Storage Materials Database[145]	Department of Energy	Properties data for adsorbents, chemical hydrides, metal hydrides	N
Infotherm[146]	John Wiley and Sons	Thermodynamic and physical properties of organic, inorganic, and organometallic compounds	Y
Inorganic Crystal Structure Database (ICSD)[147]	Fachinformationszentrum Karlsruhe—Leibniz Institute for Information Infrastructure (FIZ)	~177 000 peer-reviewed inorganic crystal structure data entries including atomic coordinates	Y
MARVEL[148]	Swiss National Science Foundation	Accelerated design and discovery of novel materials via a materials informatics platform of database-driven, high-throughput quantum simulations	N

Materials Property Database

Green et al., Appl. Phys. Rev. 4, 011105 (2017)

Materials Database (MatDB)[149]	National Renewable Energy Laboratory	Computational materials database focusing on materials for renewable energy applications such as photovoltaic materials, catalysts, thermoelectrics; Also, DFT relaxed crystal structures, thermochemical properties, and quasiparticle energy calculations providing accurate band-gaps and dielectric functions	N
Materials and Processes Technical Information System (MAPTIS)[150]	NASA	Physical, mechanical, and environmental properties for metallic and non-metallic materials used in space and aerospace applications	N
Material Property Data (MatWeb)[151]	MatWeb	Data sheets of polymers, metals, ceramics, semiconductors, fibers, and other engineering materials	N
Material Properties Database and Estimation Tool (MatDat)[152]	Matdat.com	Metals properties databases with ~800 datasets on steels, aluminum alloys, titanium alloys, weld materials, and other alloys	N (additional data for purchase)
The Materials Project[153]	Lawrence Berkeley National Laboratory	Access to computed information on known and predicted materials, as well as analysis tools to design novel materials	N
MedeA[154]	Materials Design Inc.	Software package for atomistic scale simulation of materials properties, using ICSD, Pearson and Pauling databases	Y

Materials Property Database

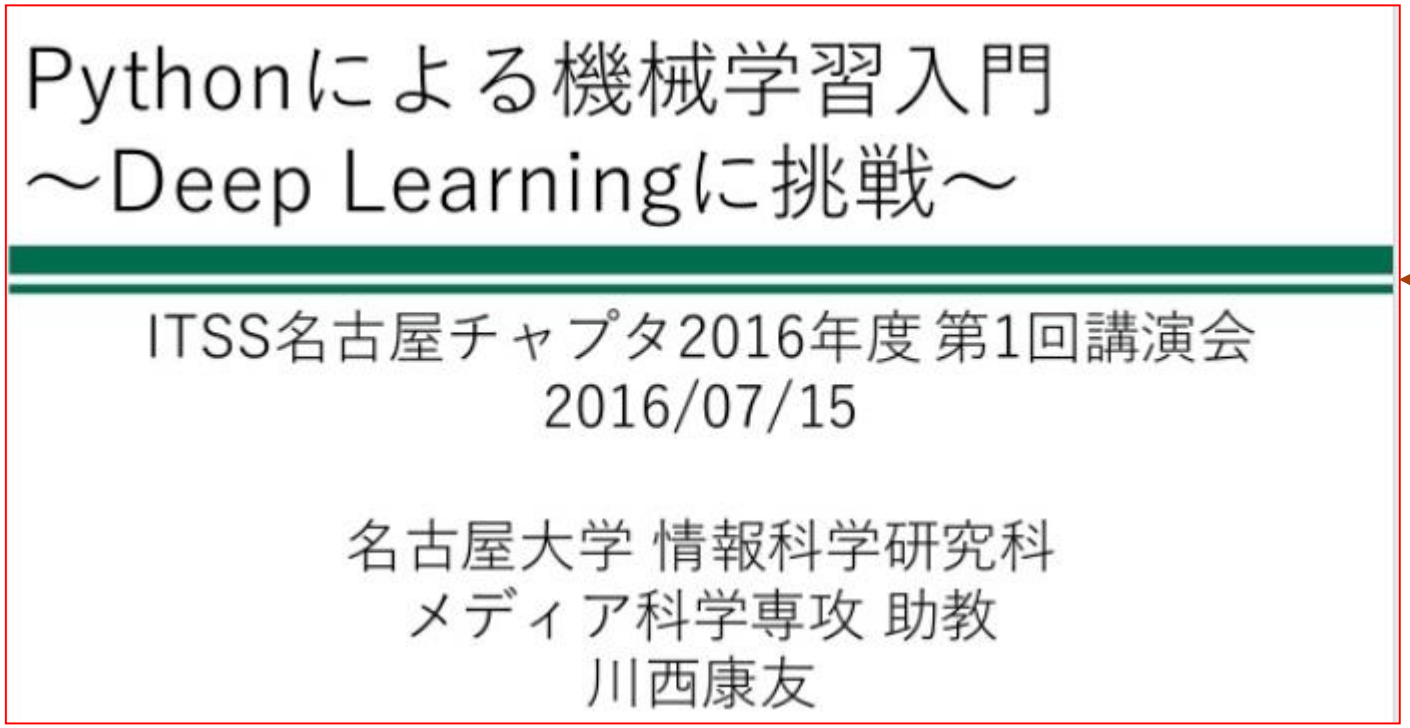
Green et al., Appl. Phys. Rev. 4, 011105 (2017)

TABLE III. (*Continued.*)

Name	Owner	Content	Fee? (Y/N)
Microelectronics Packaging Materials Database (MPMD)[155]	CINDAS LLC	Properties of 1025+ electronics packaging materials	Y
NIMS Materials Database (MatNavi)[156]	National Institute of Materials Science (Japan)	Scientific and engineering materials database including crystal structures, diffusion data, creep and fatigue data	N
NIST Alloy Data[157]	National Institute of Standards and Technology	Thermophysical property data with a focus on unary, binary, and ternary metal systems	N
NIST Data Gateway[158]	National Institute of Standards and Technology	Properties data for a broad range of materials and substances from many different scientific disciplines	N (additional data for purchase)
Novel Materials Discovery Laboratory (NOMAD)[159]	European Union Center of Excellence	Materials Encyclopedia, Big-Data Analytics and Advanced Graphics Tools for materials science and engineering	N (additional data for purchase)
Open Quantum Materials Database (OQMD)[160]	Northwestern University	DFT calculated thermodynamic and structural properties for ~475 000 materials	N
Pauling File[161]	Material Phases Data System (MPDS)	Phase diagrams, crystal structures, and physical properties databases for inorganic compounds	Y
Pearson's Crystal Data (PCD)[162]	ASM International Materials Phases Data System	Crystal structure database for inorganic compounds, including ~165 000 chemical formulas	Y
Prospector[163]	UL	Materials and ingredients search engine offering technical information for commercial products	N (additional data for purchase)
SpringerMaterials[164]	Springer Nature	Physical and chemical properties of ~250 000 materials and chemical systems	Y
Substances and Materials Databases (Knovel)[165]	Elsevier	Mechanical, chemical, corrosion, etc., properties data for a wide range of materials and coatings	Y
Thermodynamics Research Center[166]	National Institute of Standards and Technology	Thermodynamic properties tables, thermophysical properties data, models, and standards for a wide variety of compounds, binary mixtures, ternary mixtures, and chemical reactions	N (additional data for purchase)
Thermophysical Properties of Matter Database (TPMD)[167]	CINDAS LLC	Thermophysical properties of 5000+ materials	Y

参考文献

[1] 伊庭斉志「進化計算と深層学習」オーム社
[2] 長尾智晴「『機械学習』超入門セミナー」情報機構（2017）
[3] 川西康友「Pythonを用いたパターン認識・機械学習超入門」情報機構（2017）
https://www.slideshare.net/yasutomo57jp/python-deep-learning
https://www.slideshare.net/yasutomo57jp/pythondeep-learning-60544586
[4] 吉田亮「マテリアルインフォマティクスの最前線」情報機構（2018）
[5] 荒木雅弘「フリーソフトではじめる機械学習入門」森北出版株式会社
[6] 小高知宏「はじめての機械学習」オーム社
[7] 株式会社システム計画研究所「Pythonによる機械学習入門」オーム社
[8] 金森敬文「機械学習のための連続最適化」講談社
[9] 株式会社フォワードネットワークら「実装ディープラーニング」オーム社
[10] 斎藤康毅「ゼロから作るDeep Leaning」オライリー・ジャパン
[11] 人工知能学会「深層学習」近代科学社
[12] 石村貞夫「「超」入門ベイズ統計」ブルーバックス
[13] Sebastian Raschkaら「達人データサイエンティストによる理論と実践 Python機械学習プログラミング」インプレス
[14] クジラ飛行机「Pythonによるスクレイピング&機械学習」ソシム
[15] Caffeで始めるディープラーニング: https://www.slideshare.net/KotaYamaguchi1/caffe-71288204
[16] ベイズ的最適化: https://www.slideshare.net/issei_sato/bayesian-optimization
[17] Recurrent neural network
https://en.wikipedia.org/wiki/Recurrent_neural_network#/media/File:Recurrent_neural_network_unfold.svg
[18] DeepLearning における会話モデル: Seq2SeqからVHREDまで
https://qiita.com/halhorn/items/646d323ac457715866d4
[19] 日経ソフトウェア（2018年11月号）
[20] 今さら聞けないGAN（1） 基本構造の理解: https://qiita.com/triwave33/items/1890ccc71fab6cbca87e
[21] 溝口照康「機械学習を用いたナノ構造解析」JFCC（ファインセラミックスセンター）
[22] 応用物理 第88巻 第10号 (2019)
[23] 機械工学年間2019 -機械光学の最新動向- : https://www.jsme.or.jp/kikainenkan2019/chap05/
[24] 淺原彰規「さらっと学ぶマテリアルインフォマティクス」情報機構（2019）
[25] 自己学習モンテカルロ法: https://www.hpci-office.jp/invite2/documents2/ws_material_190208_nagai.pdf
[26] 2019年理研シンポジウム、埼玉、2019年12月23日



関連するセミナーなら
約5万円はする内容

https://www.slideshare.net/yasutomo57jp/python-deep-learning

https://www.slideshare.net/yasutomo57jp/pythondeep-learning-60544586

Qiita コミュニティ キーワードを入力

@ctgk 2016年12月10日に更新

PRML第6章 ガウス過程による回帰 Python実装

Qiitaなどで関連する記事を読むと良い

AIST

サポートベクターマシン

赤穂昭太郎
産業技術総合研究所

2006.7.6～7 統数研公開講座
「カーネル法の最前線—SVM, 非線形データ解析, 構造化データ—」

NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

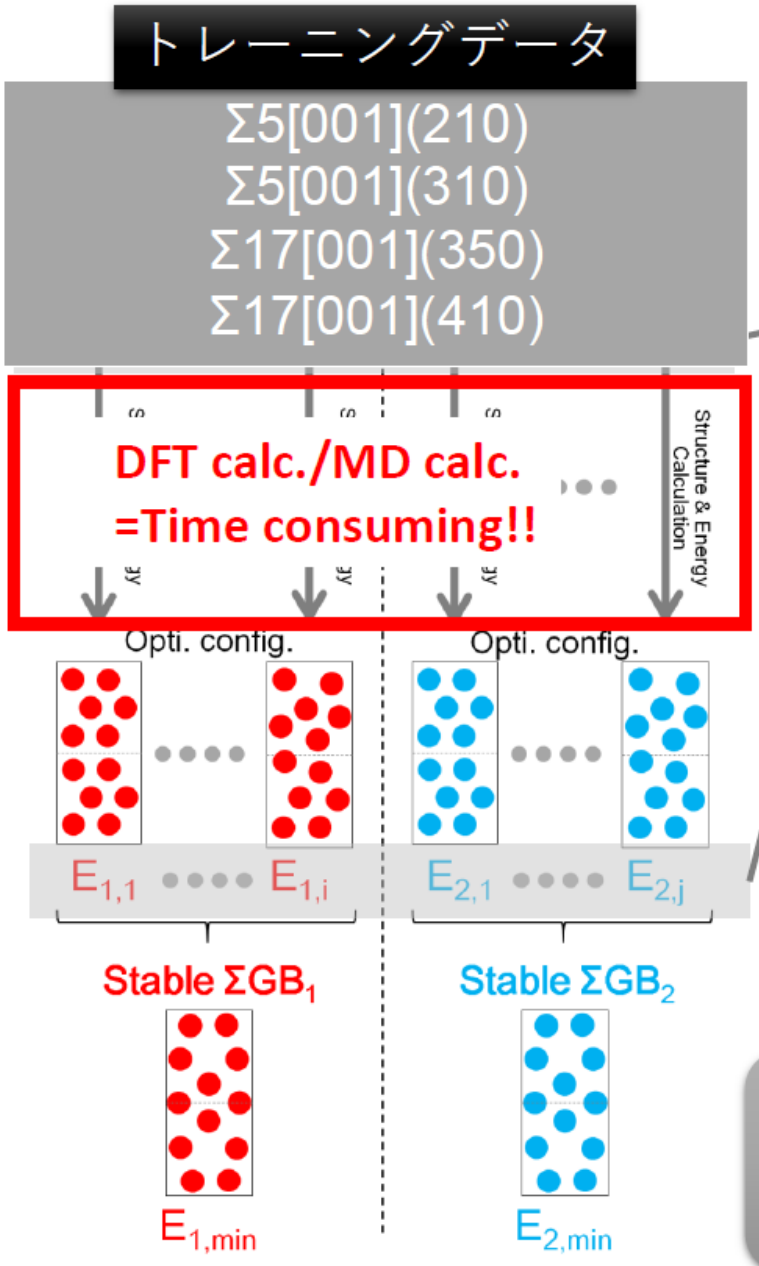
https://www.ism.ac.jp/~fukumizu/ISM_lecture_2006/svm-ism.pdf

https://www.youtube.com/channel/UCh5M2YUAPW7HnpfTUv7XHmA

入力ファイル例あり

Appendix
（無機材料の記述子の例）

回帰器(Predictor)の作り方



Predictor

GB energy = **f** (Descriptor (x) 1, x2, x3, ……)

List of Descriptors
tan(θ/2)
sin(θ/2)
Atomic density around GB
Average 1st NN (Near Neighbor) bond length
Average 2nd NN bond length
Average 1st NN bond length around GB
Average 2nd NN bond length around GB
Number of shorter bond length
Number of longer bond length
Average shorter bond length
Average longer bond length
Shortest bond length
Number of dangling bond around GB
Relative translation distance along x direction
Relative translation distance along y direction
Relative translation distance along z direction

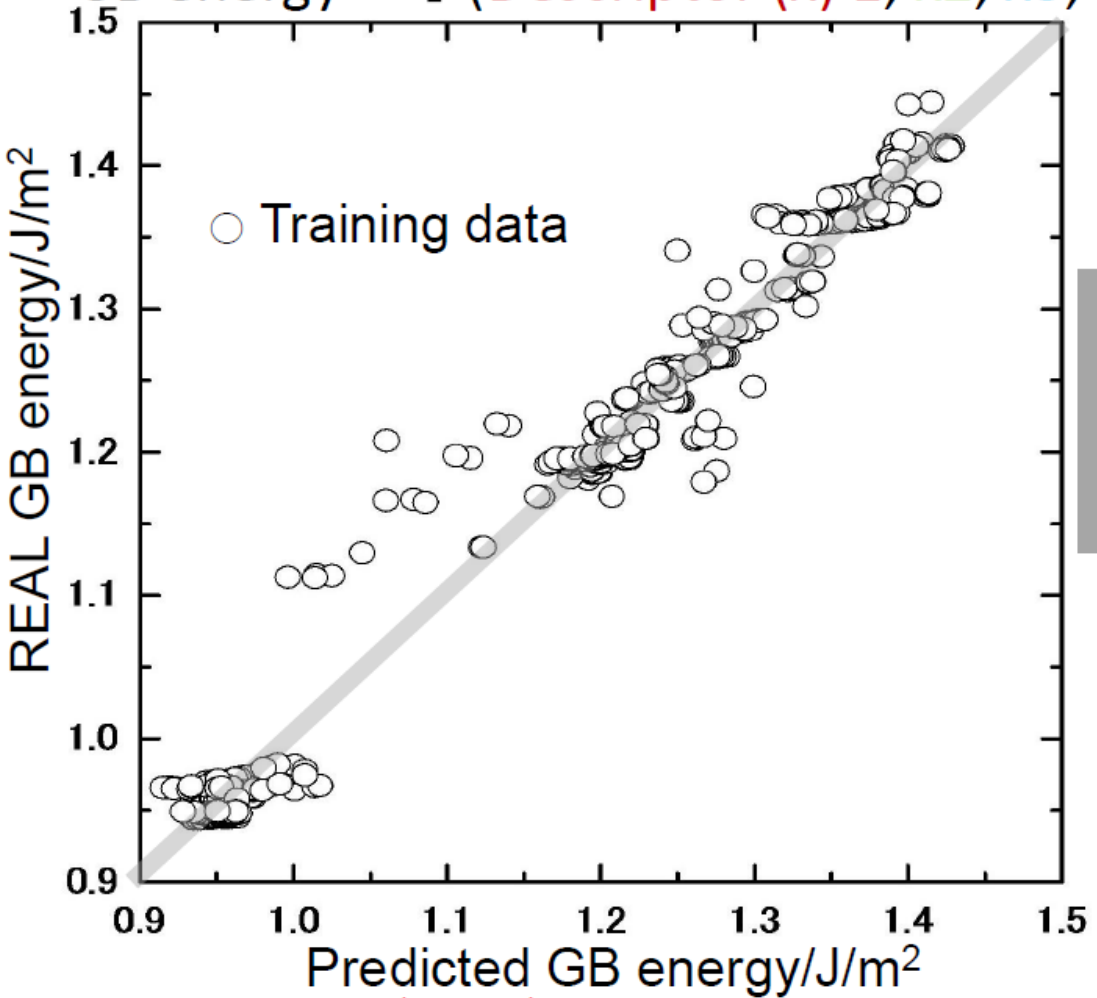
回帰: サポートベクトル回帰
対象物質: **fcc-Cu** [001]Symmetric tilt GBs
EAM-Pot. Calc.: GULP, EAM potential

S. Kiyohara et al. *Science Adv.* (2016)

回帰器(Predictor)の作製

Predictor

GB energy = **f** (Descriptor (x) 1, x2, x3, ...)

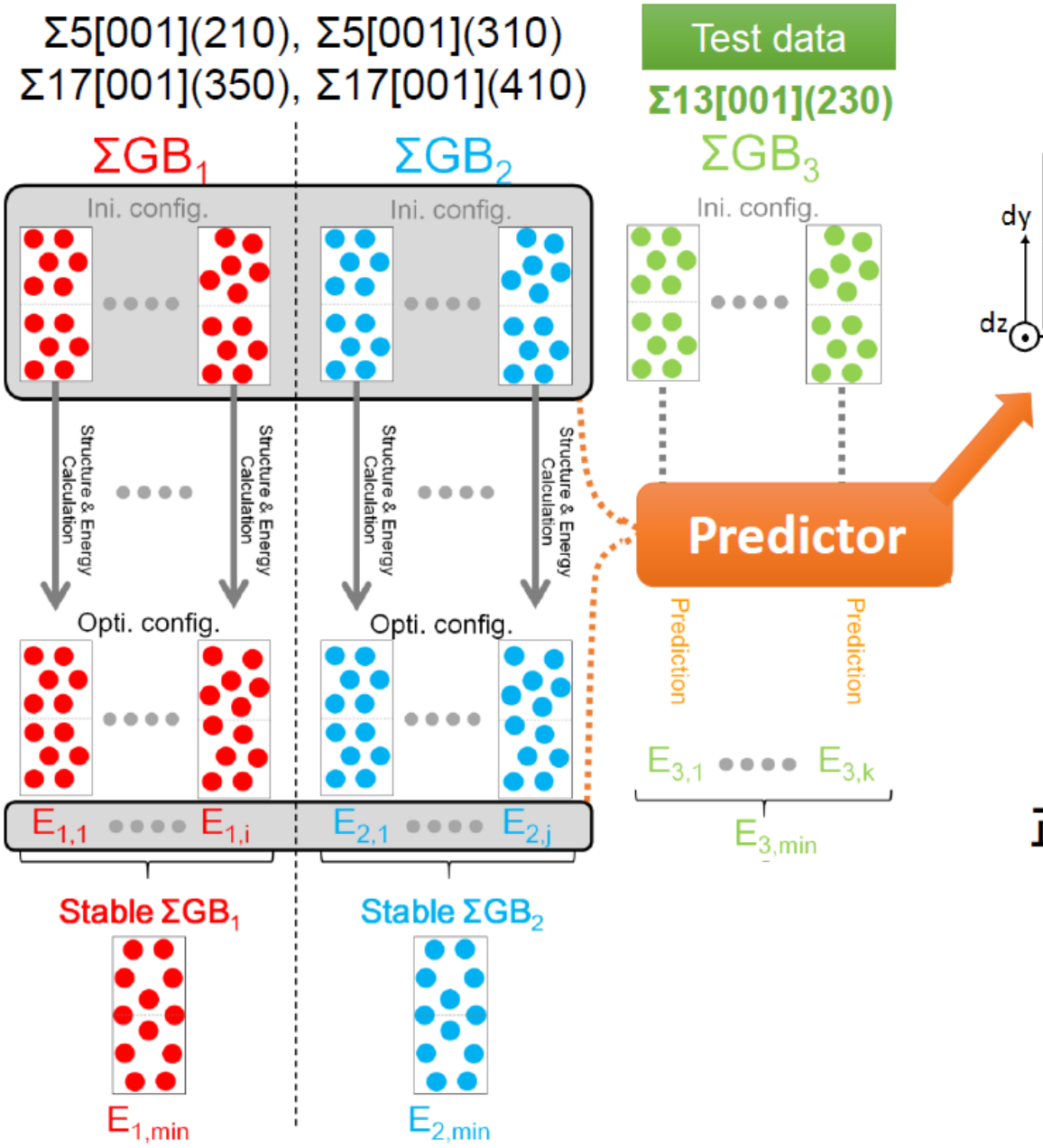


トレーニングデータ

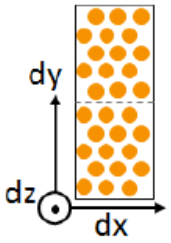
Σ5[001](210)
Σ5[001](310)
Σ17[001](350)
Σ17[001](410)

粒界エネルギーが記述子により表現されている
→ 記述子が適切に選択されている

回帰器(Predictor)のチェック：test data



銅 [001]対称傾角粒界の系統的決定



dx, dy, dz= 5.0, 1.0, 0 ang.

GB Energy = 0.96 J/m²

答え合わせ！

dx, dy, dz= **5.0, 1.0, 0** ang.

GB Energy = **0.84** J/m²

数千個の候補構造から
正確に1つをスクリーニング
↓
計算コストを大幅に削減