学習

15 分類による予測

研修内容

研修の目的

- 分類の意味とその方法について理解し，生徒に情報の特性について考えさせる授業ができるようになる。
- オープンデータの整理，加工を行い，分析を行った結果，解釈や予測の評価を行うことができるようになる。
- 数学の知識を前提とすることなく，実際の興味あるデータに関して，機械学習の考え方を用いて，分類を行うことができるようになる。

この学習項目で使用するプログラミング言語は R です。

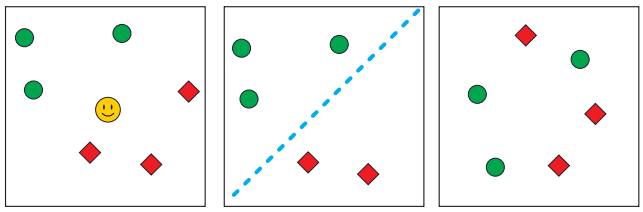
1 分類とは何か

分類(classification)とは，教師あり学習の一つである。教師あり学習(supervised learning)とは，訓練データに正解のラベルを付けて訓練する方法である。教師あり学習は，「学習13」の重回帰分析などをはじめとする「回帰」と本学習で学ぶ「分類」に分けることができる。

「回帰」と「分類」が扱う問題はほぼ共通であるがアルゴリズムやそのアプローチは異なる。例えば，次の二つは異なるアプローチと考えることができる。

- 明日の最高気温は，何℃まで上がるか→[回帰]
- 明日の最高気温は，今日の最高気温より高いか低いか→[分類]

分類は，教師あり学習の中で，有限個の選択肢の一つをラベルにして考える。分類は，単純な直線や超平面で分割できることもあるが，そうでない場合も多い 図表 1。

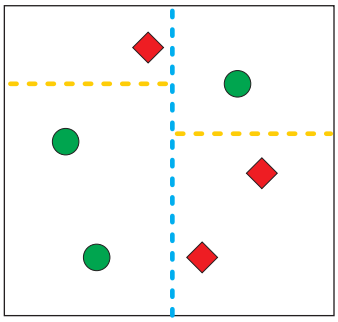


図表 1　分類問題（左），直線で分割できる場合（中），1 本の直線で分割できない場合（右）

2 決定木による二値分類

決定木とは，分類においてはモデルである。訓練データを複数の属性を基に分割し，予測に活用する。決定木には，回帰木と分類木の2種類があるが，ここでは分類木を基に決定木として説明していく。

右図のように1本の直線では分類できないデータを複数の属性(条件)を基に分割し 図表 2，その条件を演習1の最後の結果にあるような木構造で表現する 図表 6。



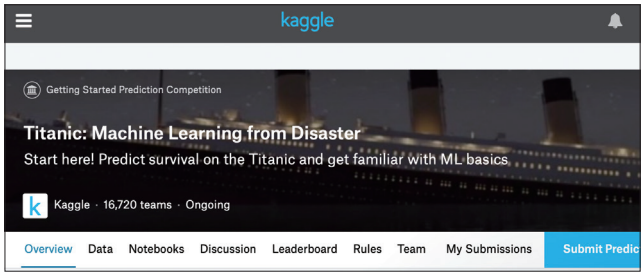
図表 2　属性ごとに境界線を引いたデータの分類

演習 1

EXERCISE

kaggle から，titanic データをダウンロードし図表3，タイタニック号に乗っていた乗客の年齢，性別，客室に関して，その生存の有無を表す決定木を作成してみましょう。

kaggleからtitanicデータをダウンロードする。タイタニック号に関しては，映画にもなったのでご存じの方も多いと思うが，1912年4月15日に氷山と衝突し，沈没した旅客船であり，事故の結果，乗組員と乗客合わせて2,224人のうち1,502人が亡くなった痛ましい事故で有名である。「train.csv」は，その一部の乗客に関するデータである。データの読み込みから始める。



図表3 「kaggle」 https://www.kaggle.com/c/titanic

```
01  titanic.train<-
      read.csv("c:/Users/shinya/Desktop/titanic/train.csv")   # データの場所を指定
02  str(titanic.train)
```

train.csvを読み込むと，12個の属性を持つ891人の乗客のデータであることが分かる。乗客の名前なども含まれているが，ここでは，Pclass（客室の等級），Sex（性別），Age（年齢），Survived（生存1,死亡0）の情報のみが必要なので，必要な部分を取り出す。

```
01  titanic.data<-titanic.train[,c("Pclass","Sex","Age","Survived")]
02  titanic.data
```

出力結果

```
  Pclass    Sex Age Survived
1      3   male  22        0
2      1 female  38        1
3      3 female  26        1
4      1 female  35        1
5      3   male  35        0
6      3   male  NA        0
```

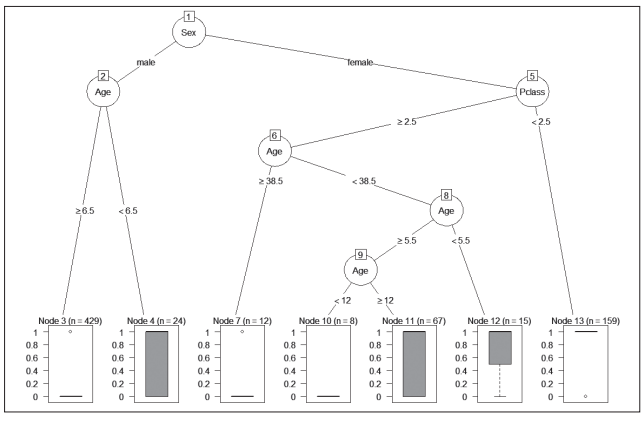
欠損値(NA)があるので，欠損値を取り除く。

```
01  titanic.data<-na.omit(titanic.data)
```

分類木を描くには二つのライブラリを読み込む。もしライブラリを初めて使用する場合は，install.packages ("partykit")のようにインストールをしてから行う。method="class"は，分類木の作成に必要なオプションである。

```
01  library(rpart)
02  library(partykit)
03  titanic.ct<-rpart(Survived~.,data=titanic.data, method="class")
04  plot(as.party(titanic.ct),tp_arg=T)
```

実行すると，右のような図が表示されるが，どのように解釈すればよいだろうか図表4。何が生死を決めた最大の要因となっているだろうか。分類木が複雑なので，少し剪定(pruning)を行う。そのままでは，全ての訓練データに当てはまるモデルを作成するため，過学習(over fitting)となり，訓練に使用していないテスト用のデータを使う場合に，予測が合致しない可能性がある。そのため，適度なところで，分類木の枝を剪定し，予測モデルとして有効な分類木モデルを作成することが必要になる。



図表4 titanic データによる分類木（剪定前）

この分類木のCP（複雑度：complexity parameter）と呼ばれるパラメータを表示してみよう。printcpの結果を見るとCP=0.027を超えたあたりで収束しているのが分かる図表５。

```
01 printcp(titanic.ct)
```

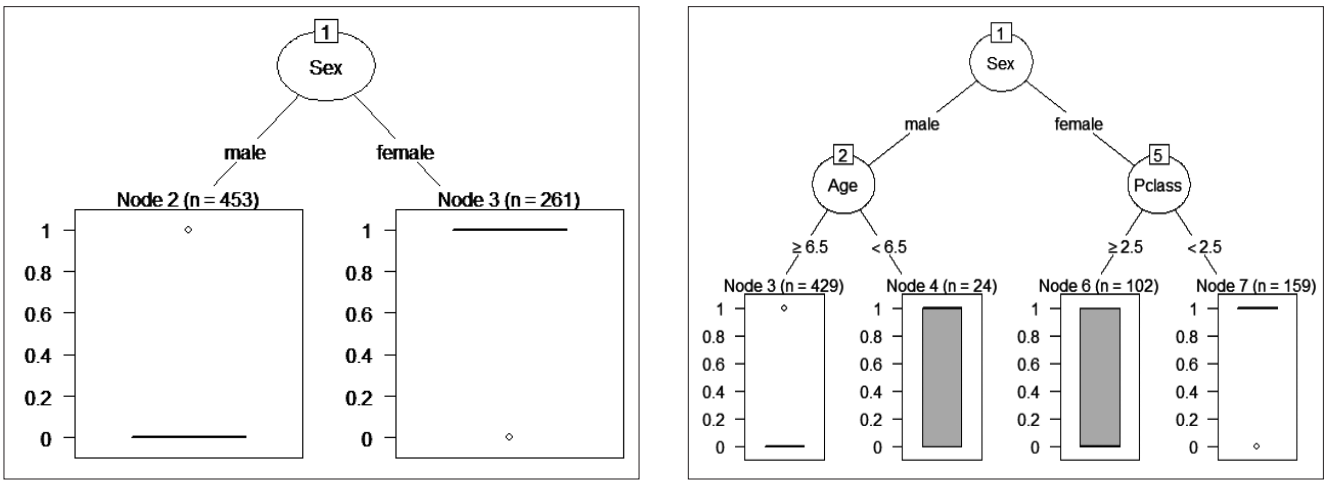
```
Classification tree:
rpart(formula = Survived ~ ., 
data = titanic.data, method = "class")

Variables actually used in tree construction:
[1] Age    Pclass Sex

Root node error: 290/714 = 0.40616

n= 714

        CP nsplit rel error  xerror     xstd
1 0.458621      0   1.00000 1.00000 0.045252
2 0.027586      1   0.54138 0.54138 0.038162
3 0.012069      3   0.48621 0.56897 0.038840
4 0.010345      5   0.46207 0.54483 0.038249
5 0.010000      6   0.45172 0.54483 0.038249
```

図表５　分類木の剪定の解析結果

CPを0.028にした場合の分類木を出力する。

```
01 titanic.ct2<-rpart(Survived~.,data=titanic.data,
     method="class", CP=0.028)
02 plot(as.party(titanic.ct2))
```

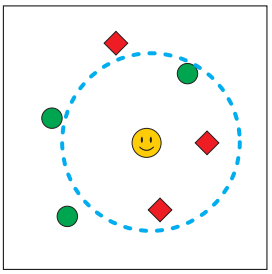
この事故の生死を決める最大の要素は，性別であった図表６（左）。乗務員が積極的に女性や子供を救助したことも読み取れる図表６（右）。また，船室の優劣は生死を決める要因にはなっていないようである。分類木は，このような分析をするだけでなく，別のデータに関する予測も行うことができる。このデータをこの学習で採用したのは，データサイエンスの分野において，よく知られたデータであるだけでなく，このように分類しやすく，分類木によって，ここにあるようなデータを眺めるだけでは見つけにくかった結果を導き出すことができるためである。データを分析することにより人間の尊厳に気付かされることもある。



図表６　２つの分類木CP=0.028（左），CP=0.027（右）

3 k-近傍法による分類

k-近傍法(k-nearest neighbor method，kNN)とは，予測したい値に最も距離が近いk個を考え，その中で多数決をとり，多い値をその予測値とする考え方である。右の図では，スマイルマークが予測したい値の位置で，k=3としたときの近傍の範囲を表している。この場合は，◆がスマイルマークの予測値となる図表7。

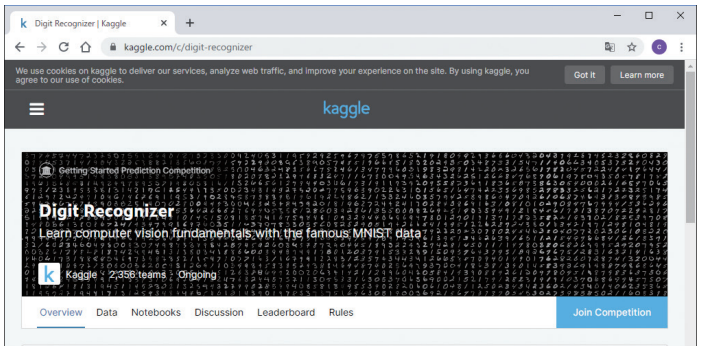


図表7　スマイルマークを予測する k=3 の近傍範囲

演習2

EXERCISE

kaggle から，digit-recognizer のデータ(MNIST データ) をダウンロードし図表8，train.csv の一部を訓練データとして，k-近傍法を用いて学習させ，train.csv の一部をテストデータとして，その正答率を調べましょう。



図表8　「kaggle」https://www.kaggle.com/c/digit-recognizer

kaggleから手書き数字データ(MNISTデータ)を読み込む。

```
01 mnist<-read.csv("c:/Users/shinya/Desktop/digit-recognizer/train.csv")
```

このtrain.csvデータは，42,000個の手書き文字が，正解ラベルのlabel（1列目）と784（28×28）個のピクセルの階調データ(0 ～ 255)が収録されている。このままでは，大きすぎて処理の時間がかかることから，一部のみを利用する。訓練データとして1,000個，テストデータとして100個のデータを使用する図表9。訓練データとテストデータを作成する。

図表9　テストデータ（抜粋）

```
01 mnist.light<-mnist[1:1000,]
02 mnist.light.test<-mnist[1001:1100,]
```

k=3としたk-近傍法をもとに，テストデータ100個の正解候補を作らせる。mnist.light[,-1]は，訓練データのピクセルデータ，mnist.light.test[,-1]は，テストデータのピクセルデータ，mnist.light[,1]は，訓練データのラベルデータ，k=3は，kNNのkの値である。classパッケージは，標準で入っているので，呼び出すだけですぐに利用することができる。最初の出力は，100個のテストデータの推定値である図表10(6~9行目)。attr(,"prob")は，確率を表す。近傍の点3つのうちの採用した値の割合を表している図表10。

```
01 library(class)
02 mnist.rp<-
     knn(mnist.light[,-1],mnist.light.test[,-1],
         mnist.light[,1],k=3,prob=T)
03 mnist.rp
```

```
  [1] 1 5 1 7 9 8 9 5 7 4 7 2 8 1 4 3 8 6 2 7 2 6 7 8 1 8 8 1 9 0 9 4
 [33] 6 6 8 2 3 5 4 5 4 1 3 7 1 5 0 0 9 5 5 7 6 8 2 8 4 2 3 6 4 8 0 2
 [65] 4 7 3 4 4 5 4 3 3 1 5 1 0 2 2 2 9 5 1 6 6 9 4 1 7 2 2 0 7 0 6 8
 [97] 0 5 7 4
attr(,"prob")
  [1] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
  [7] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
 [13] 0.6666667 1.0000000 0.6666667 0.6666667 0.6666667 1.0000000
 [19] 1.0000000 0.6666667 0.3333333 1.0000000 1.0000000 1.0000000
 [25] 1.0000000 0.6666667 1.0000000 1.0000000 1.0000000 1.0000000
 [31] 0.6666667 0.6666667 1.0000000 1.0000000 1.0000000 1.0000000
 [37] 1.0000000 1.0000000 1.0000000 1.0000000 0.6666667 1.0000000
 [43] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
 [49] 0.6666667 1.0000000 0.6666667 1.0000000 1.0000000 0.6666667
 [55] 1.0000000 1.0000000 0.6666667 1.0000000 1.0000000 1.0000000
 [61] 0.3333333 1.0000000 0.3333333 1.0000000 0.6666667 1.0000000
 [67] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.6666667
 [73] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
 [79] 0.6666667 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
 [85] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000
 [91] 1.0000000 1.0000000 0.7500000 1.0000000 1.0000000 1.0000000
 [97] 1.0000000 0.6666667 1.0000000 1.0000000
Levels: 0 1 2 3 4 5 6 7 8 9
```

図表10　100個のテストデータの推定値

実際のテストデータのラベルと比較してみよう図表11。

```
01 mnist.rp==mnist.light.test[,1]
```

```
  [1]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
 [11]  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
 [21]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [31] FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [41]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
 [51] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [61] FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [71]  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [81]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [91]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

図表11　テストデータのラベルとの比較結果（正解はTRUE，不正解はFALSE）

実際の正答率はいくつだろうか。次のプログラムで確認する。

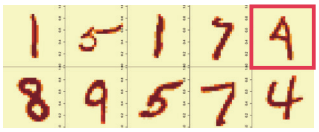
```
01 mean(mnist.rp==mnist.light.test[,1])
```

```
[1] 0.88
```

図表12　正答率

88%の正答率であることが分かる図表12。訓練データが小さいがそこそこ良好な結果が出ている。五つ目のテストデータが誤って認識されている。実際にどんな手書き文字なのか見てみよう図表13。次のプログラムで最初の10個を表示してみる。ピクセル行列とグラフの座標の向きが異なるので，表示を少し工夫している。

```
01  par(mfrow=c(length(1:10)/2, 5))
02  par(mar=c(0,0,0,0))
03  for(i in 1:10){
04    m<-matrix(data.matrix(mnist.light.test[i,-1]),28,28)
05    image(m[,28:1])}
```



図表13　手書き文字の描画（最初の10個）

手書き文字の描画の上段の一番右が誤った数である図表13。読者のみなさんは，いくつに見えるだろうか。kNNは「9」と判定したが，実際ラベルは「4」だったようだ。

次に混同行列(Confusion matrix)を作ってみよう図表14。

縦軸が予測値，横軸が正解ラベルである。先ほどの予測が9，正解が4であるものが最下行(予測9の行)に見つけられる。左上から右下への対角線上の数の合計が正答の個数である。

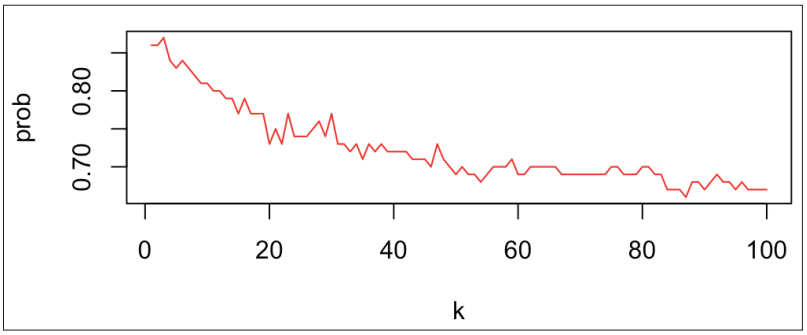
それでは，kの値を増やせば，より正答率は上がるのだろうか。実験した結果が次の図である図表15。どうやらk=3としたのは，悪くはなかったようである。より，正解率を上げるには，訓練データの件数を多くするとよいだろう。ここでは，k-近傍法を扱ったが，分類のアルゴリズムは，この他にも様々な手法がある。また，MNISTデータの分析では，学習17でも扱うニューラルネットワークによる学習が有名である。ここでの近傍を求める距離は，784次元(147ページ参照)のユークリッド距離を計算しているが，距離が何であるかを生徒に理解させるには，階調(0～255)ではなく，0と1で2値化されたデータと考えると分かりやすくなる。

実際の正答率はいくつだろうか。次のプログラムで確認する。

```
01  cfm<-table(mnist.rp,mnist.light.test[,1])
02  cfm
```

```
mnist.rp   0  1  2  3  4  5  6  7  8  9   ← 正解ラベル
        0  7  0  0  0  0  0  0  1  0  0
        1  0 10  0  0  0  0  0  0  0  1
        2  0  0 12  0  0  0  0  0  0  0
        3  0  0  0  5  0  0  0  0  2  0
        4  0  0  1  0 11  0  0  0  0  1
        5  0  0  0  1  0 10  0  0  0  0
        6  0  0  0  0  0  0  9  0  0  0
        7  0  0  1  0  0  0  0 10  0  0
        8  0  0  1  0  0  0  0  0 10  0
        9  0  0  0  0  1  0  0  1  1  4
   ↑ 予測値          ↑ 9と予測したが正解は4
```

図表14　混同行列



図表15　kの値と正答率の関係

【参考文献・参考サイト】
●「kaggle」 https://www.kaggle.com/
●「IPython データサイエンスクックブック第 2 版」 Cyrille Rossant 著　菊池彰 訳　オライリージャパン(2019)
●「R ではじめるデータサイエンス」 Hadley Wickham, Garrett Grolemund 著　大橋真也 監修，黒川利明 訳　オライリージャパン(2017)
●「データサイエンスのための統計学入門」 Peter Bruce, Andrew Bruce 著　大橋真也 監修　黒川利明 訳　オライリージャパン(2018)

学習活動と展開

学習活動の目的

- 分類について，その目的や方法が理解できる。
- 分類木について，その解釈方法や剪定の方法が理解できる。
- k- 近傍法について，その原理と検証方法，学習効果の向上の方法に関して理解できる。
- 分類が，予測に活用できることが理解できる。

学習活動とそれを促す問い

	問　い	学習活動
展開 1	分類や教師あり学習について調べてみよう。	教師あり学習や分類の原理や社会での活用事例について調べる。
展開 2	分類木について調べてみよう。	決定木や分類木について，その方法と活用方法，解釈方法などを理解する。
展開 3	k- 近傍法について調べてみよう。	k- 近傍法について，その原理や予測の正確さに関して，k の値や訓練データの個数を変化させて調べる。

展開 1	
問　い	分類や教師あり学習について調べてみよう。
学習活動	教師あり学習や分類の原理や社会での活用事例について調べる。
指導上の留意点	活動につまずいている生徒に，必要に応じてヒントを与え，生徒自らが分類の有用性を理解するよう促す。

展開 2

問　い	分類木について調べてみよう。
学習活動	決定木や分類木について，その方法と活用方法，解釈方法などを理解する。
指導上の留意点	● データを分析に活用できるようにデータクリーニングができるように指導する。 ● 分類木の木構造の読み方や解釈の仕方，その原因についての独自の解釈等ができるように促す。

展開 3

問　い	k- 近傍法について調べてみよう。
学習活動	k- 近傍法について，その原理や予測の正確さに関して，k の値や訓練データの個数を変化させて調べる。
指導上の留意点	● k の値の意味がきちんと理解できているか，訓練データの個数やテストデータの個数が適正であるか，考えさせる。 ● k- 近傍法以外の方法での予測と関連付ける。また他の方法について調べさせる。

まとめ

まとめ	データの分類に関する様々な方法とその解釈や社会における分類の活用方法に関して理解させる。

学習

16 クラスタリングによる分類

研修内容

研修の目的

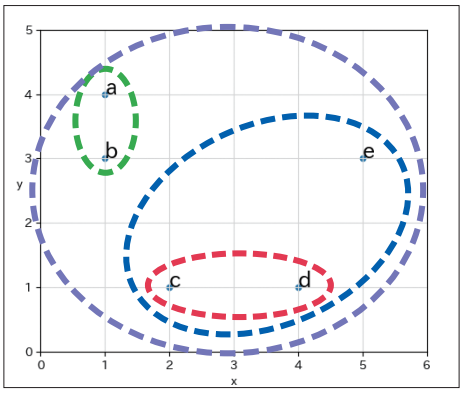
- 教師なし学習によるクラスタリングやアソシエーション分析の手法について理解し，データを用いたクラスタリングやアソシエーション分析を行う授業ができるようになる。
- 教師なし学習によるクラスタリングやアソシエーション分析による結果を評価し，必要に応じてアルゴリズムを選択するなどの分析方法の改善について理解し，アルゴリズムの評価や改善について考えさせる授業ができるようになる。
- 教師なし学習によるクラスタリングやアソシエーション分析の有用性を理解し，これらの分析手法が適用できる場面を考えさせたり，問題解決として活用できる場面を考えさせたりする授業ができるようになる。

この学習項目で使用するプログラミング言語は Python です。

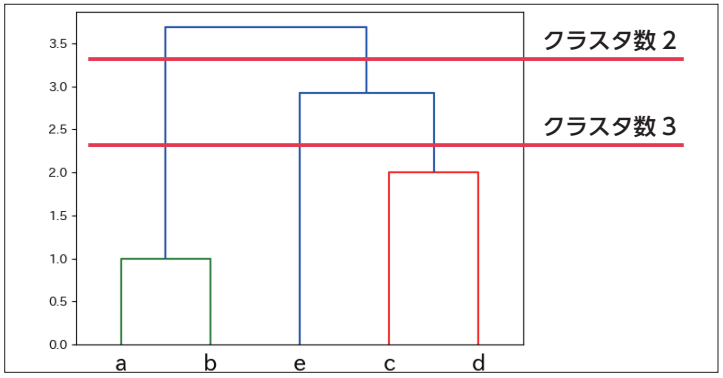
1 階層的クラスタリング

一般に分類とは，人間が与える正解より得られる特徴から，データを分析することで，その特徴に基づいて新しいデータを予測する教師あり学習の一つである。それに対して，特に正解を与えずに(教師なし学習という)，似ているデータをまとめて，いくつかのクラスタといわれるグループに分割する手法をクラスタリングという。一つ一つのデータを一つのクラスタとし，距離が最も近いクラスタ同士を併合して階層的にクラスタを形成する方法を(凝集型)階層的クラスタリングという図表１。クラスタリングの過程において，併合の様子を木として図示したものをデンドログラム(樹状図・樹形図)という図表２。

図表１で示した点について階層的クラスタリングを行う方法を示す。最も距離が近い二つのクラスタa, bをまとめて，新しいクラスタとする。このように，距離が近いクラスタ同士を順次まとめて，最終的に一つのクラスタになるまで繰り返す。このときのまとまる手順とそのクラスタ間の関係を図にしたものが図表２のデンドログラムである。この図において，縦軸はクラスタ間の距離である。このデンドログラムに水平線を描いたとき，デンドログラムと水平線の交点の数がクラスタ数になる。階層的クラスタリングでは，あらかじめクラスタ数を決める必要がなく，結果を解釈する際に決めることができる。



図表１　クラスタリングの様子



図表２　クラスタリングによりできたデンドログラム

ここで，併合された新しいクラスタの代表点を決める方法には，最短距離法，最長距離法，群平均法，ウォード法などがある。分類結果として比較的良好なものが得られる群平均法やウォード法が使われることが多い。距離については，ここでは通常の距離（ユークリッド距離）を用いるが，他にはマンハッタン距離やジャッカード係数，コサイン類似度などがあり，データの特性に応じて選択する。ユークリッド距離を求めるときに，データの値の尺度やデータの単位が異なる場合などは，そのまま距離を求めることが適切でない場合もある。このような場合には，スケーリングしてから距離を求める必要がある。分類されたクラスタを解釈する際は，クラスタの代表的なデータを提示したり，平均的な特徴を提示したり，クラスタの特徴を表す名前を付けたりして，クラスタリングの結果を活用する。

演習 1

EXERCISE

e-Stat で公開されている家計消費状況調査の「特定の財（商品）・サービスの 1 世帯当たり 1 か月間の支出を全国・地方・都市階級別にまとめたのデータ（表番号 3-1）」を用いて，地方ごとの支出額を基にクラスタリングをし，家計の傾向についての地方間の類似性やその傾向について考えてみましょう。

e-Statで「都道府県の指標　基礎データ　人口・世帯　2020」とキーワード検索を行い，表計算ソフト形式のデータをダウンロードする。ダウンロードしたファイルを，表計算ソフトを用いて不要な列を削除する。また，品目が50種類に分類されているが，データ中の品目の分類（通信，旅行関係，教育・娯楽など）ごとに支出傾向が似ている地方を調べられるよう，分類ごとに小計を求める。小計を求めたら，品目ごとの行を削除する。地方について，分類ごとの支出額により階層的クラスタリングを行うには，一つの地方について 1 行にデータが配置されている必要がある。表計算ソフトの行と列を入れ替えて貼り付ける操作により，行と列を転置する。データを加工したらCSV-UTF8形式で保存し，このデータを基にクラスタリングを行う。

階層的クラスタリングを行うには，クラスタとして分類したいデータを1行に配置しておく必要がある。入手したデータがロングフォーマットの場合には学習12の演習5の方法を用いてワイドフォーマットに変換する必要がある。

```
01  import pandas as pd
02  df = pd.read_csv("household_economy.csv", index_col=0)
03  df.head()
```

※ファイル名を一部修正しています。

品目区分（平成29年改定）	通信費用	旅行費用	教育費用	衣料費用	医療費用	家具費用	家電等費用	家屋費用	自動車費用	冠婚葬祭費用	仕送り金
北海道	13065	5547	6633	1559	1877	1166	3361	7768	19632	4030	2529
東北	14672	4985	6414	1726	1367	970	3358	7674	26949	7447	2664
関東	15770	8264	12569	2703	1913	1123	4724	9865	18813	6294	1925
北陸	14751	4882	8216	2116	1579	1319	4291	13954	24862	8369	2837

図表 3　特定の財（商品）・サービスの 1 世帯当たり 1 か月間の支出（e-Stat のデータを加工して利用）

クラスタリングを行う前に，分類ごとに金額を確認する。分類ごとの支出額について地方間の差を比較すると，自動車費用は他の分類の差よりも大きな差になっている図表３。このままクラスタリングを行ってしまうと，分類ごとの支出額の差を用いて距離を求めることになり，差が大きい自動車費用の影響が大きくなると考えられる。そこで，次のプログラムにより，クラスタリングの前に分類ごとの影響が同等になるようデータの基準化を行う。

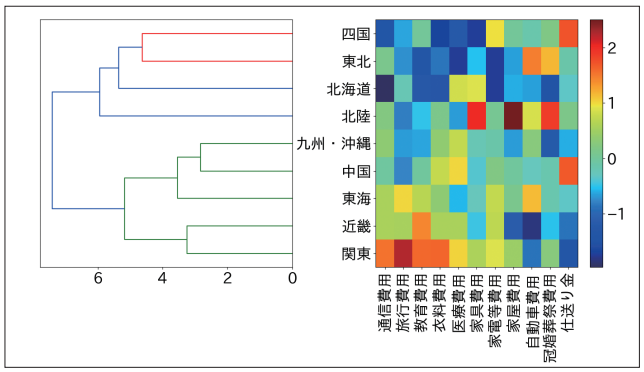
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
sc.fit(df)
df_std = pd.DataFrame( data=sc.transform(df), index=df.index,
columns=df.columns)
```

次に，基準化した値を用いてクラスタリングを行う。

```
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram,linkage
z = linkage( df_std, method='ward', metric='euclidean' )
den = dendrogram( z, labels=df.index, orientation='left',
distance_sort='descending' )
```

クラスタリングの結果，デンドログラムが得られる図表４。このプログラムには載せていないが，項目ごとの値が分かるようヒートマップを追加している(デンドログラムの色はデフォルトのもので，特にクラスタ数を意識したものではない)。

デンドログラムやヒートマップを見ることにより，いくつのクラスタに分割することが適当か，また，それらのクラスタにはどのような特徴があるかなどを考えることができる。



図表４　結果として得られるデンドログラム

2 k-means法によるクラスタリング

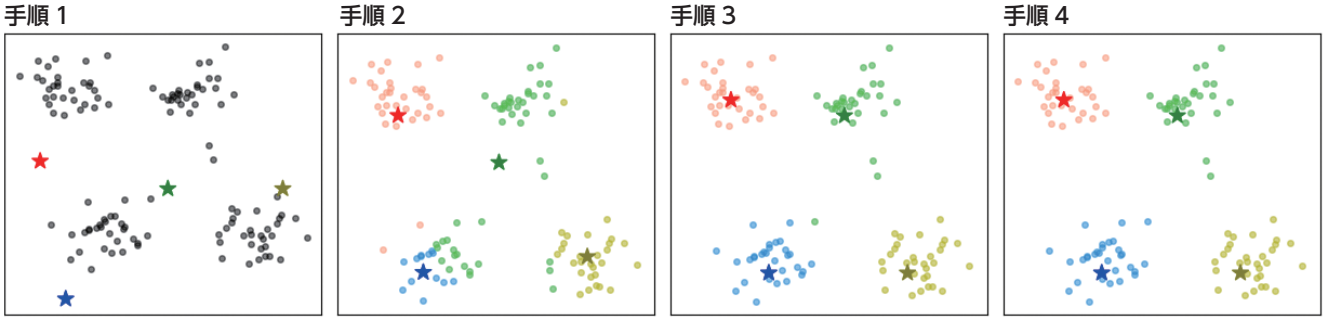
クラスタリングの別の方法として，k-means法(k平均法)がある図表５。k-means法では，次の手順によってクラスタリングする。

1) あらかじめ分割するクラスタ数を決めておき，ランダムに代表点(セントロイド)を決める。
2) データと各代表点の距離を求め，最も近い代表点のクラスタに分類する。
3) クラスタごとの平均を求め，新しい代表点とする。
4) 代表点の位置が変わっていたら2に戻る。変化がなければ分類終了となる。

1)によりランダムに代表点を決めることによって，結果が大きく異なり，適切なクラスタリングとならない場合もある。何回か繰り返して分析をしたり，k-means++法を用いたりすることにより改善することができる。

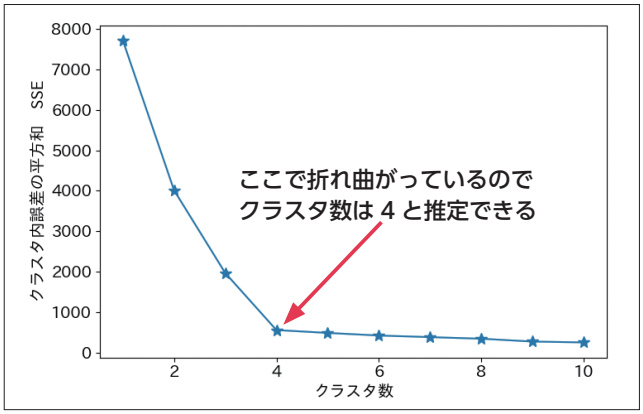
k-means++法では代表点の初期値の決め方1)を，次の1')に変更したアルゴリズムになる。

1')データの中からランダムに一つの代表点を選び，その点からの距離の2乗に比例した確率で残りの代表点を選ぶ。k-means法ではクラスタ数をあらかじめ

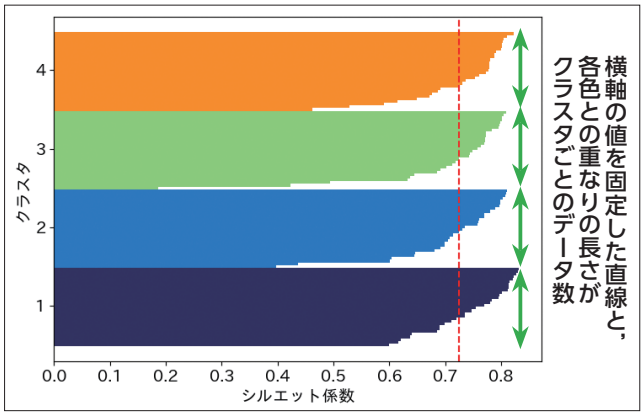


図表５　k-means法によるクラスタリング（★印は代表点）

決める必要があるが，その適切なクラスタ数の推定方法としてエルボー法がある。クラスタ数とSSE（クラスタ内誤差の平方和）を折れ線グラフで表したとき図表6，肘のように曲がっているところのクラスタ数にするものである。データによっては折れ曲がる点がはっきりしない場合があり，その場合にはシルエット図という図を用いたシルエット分析といった手法も併せて使われる図表7。



図表6　クラスタ数と SSE によるクラスタ数の推定



図表7　シルエット図

演習2

EXERCISE

カリフォルニア大学アーバイン校（UCI）が運営する機械学習用データ配布サイトで公開されている卸売業者の顧客データを用いて，顧客をクラスタリングし，どのような顧客が購入しているか分析してみましょう。

UCI（http://archive.ics.uci.edu/ml/datasets/Wholesale+customers）から，卸売業者のデータのファイル名をWholesale_customers_data.csvに変更してダウンロードする。そのファイルから，プログラムを用いてデータを読み込む。

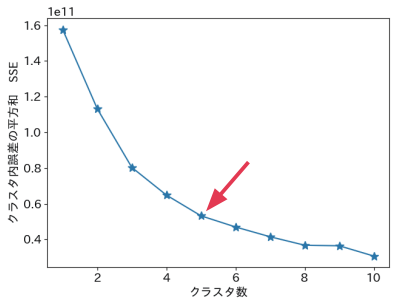
```
01 import pandas as pd
02 df = pd.read_csv("Wholesale_customers_data.csv")
```

このデータは，8個の属性から構成されており，1行1顧客のデータとなっている。このうち，6個の属性が年間注文額となっており，これらの属性を用いてクラスタリングを行う。

```
01 sub_cols=df[['Fresh','Milk', 'Grocery','Frozen', 'Detergents_Paper','Delicassen']]
```

金額を用いてクラスタリングを行いたいので，値の基準化は行わない。次にクラスタ数とSSEの関係をグラフに描き，クラスタ数をエルボー法で推定する図表8。グラフを見ると，極端に折れ曲がっているクラスタ数はないが，クラスタ数が5の場合の点からゆるやかに減少していることから，クラスタ数を5とする。

```
01 import matplotlib.pyplot as plt
02 from sklearn.cluster import KMeans
03 dist_list=[]
04 for i in range( 1, 11 ):
05     kmeans = KMeans( init='random',
       n_clusters=i,random_state=0 )
06     y_km = kmeans.fit( sub_cols )
07     dist_list.append( kmeans.inertia_ )
08 plt.plot(range(1,11),dist_list,marker='*',
   markersize=10)
09 plt.xlabel('クラスタ数')
10 plt.ylabel('クラスタ内誤差の平方和　SSE')
```



図表8　クラスタ数の推定

次にk-means法によりクラスタリングする。

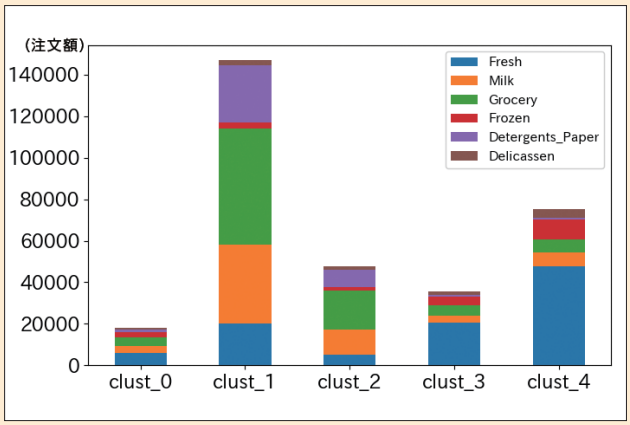
```
kmeans = KMeans( init='random', n_clusters=5, random_state=0 )
pred = kmeans.fit_predict( sub_cols )
df['cluster_id'] = pred
```

このようにクラスタリングしたデータの特徴を，属性ごとに平均値を求めて調べる(凡例の位置は調整してある)図表９。

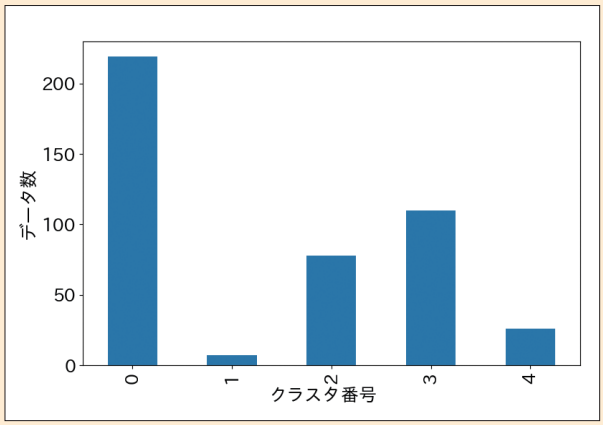
```
clusterinfo = pd.DataFrame()
for i in range(5):
    clusterinfo['clust_' + str(i)] = df[df['cluster_id'] == i].mean()
clusterinfo = clusterinfo.drop(['Channel','Region','cluster_id'])
clusterinfo.T.plot.bar(stacked=True)
```

また，各クラスタに含まれるデータ数を次のプログラムで調べる図表10。

```
labels = pd.Series(kmeans.labels_,name='cluster_number')
ax=labels.value_counts(sort=False).plot(kind='bar')
ax.set_xlabel('クラスタ番号')
ax.set_ylabel('データ数')
```



図表９　クラスタごとの注文額の平均値



図表 10　クラスタごとのデータ数

演習 3

EXERCISE

各クラスタの特徴とこの卸売業者が売り上げを伸ばすための方策を考えましょう図表９図表10。また，クラスタ数を５として分析を行いましたが，クラスタ数を変えた場合はどのように変化するかを考えましょう。

3 アソシエーション分析

ネットショッピングでは，商品の閲覧や購入の履歴を基にお薦めの商品が提案される。また，SNSでのお薦めのユーザーや動画サイトでのお薦めの動画などの提案もなされている。履歴データなどを用いてデータの結び付きの強さを求める分析をアソシエーション分析という。アソシエーション分析の事例として「おむつを買った人は缶ビールを買う傾向にある」という分析を紹介されることが多い。一人の客による1回の購入データをマーケットバスケットデータといい，これを基に分析するマーケットバスケット分析を取り上げる。

ここでは，次のような購入データがある場合についてマーケットバスケット分析により分析する図表11。

マーケットバスケット分析では，支持度，確信度，リフト値という３つの値が評価指標としてよく用いられる。

レシート番号	購入したもの
1	コーヒー，パン，弁当
2	コーヒー，パン，弁当
3	コーヒー，パン
4	コーヒー，弁当
5	お茶，弁当，パン
6	お茶，弁当
7	紅茶，弁当
8	紅茶，パン，弁当

図表 11　購入データの例

(1) 支持度：全ての商品を買った人のうち，特定の商品を買った人の割合

supp(コーヒーとパン)= 3/8 = 0.375
supp(お茶とパン)= 1/8 = 0.125

支持度が高いものは，全体のデータの中で多く現れる。コーヒーとパンの組はよく買われているが，お茶とパンの組はあまり買われていない。

(2) 確信度：商品 X を買った人のうち，商品 X も商品 Y も両方とも買った人の割合

conf(コーヒー→パン)= 3/4 = 0.75
conf(パン→コーヒー)= 3/5 = 0.6
conf(コーヒー→弁当)= 3/4 = 0.75

コーヒーを買った人はパンを買っているが，パンを買ったからといってコーヒーを買うとは限らない。コーヒーとパンを一緒に買う人とコーヒーと弁当を一緒に買う人の割合は同じであることが分かる。

(3) リフト値：X → Y の確信度を商品 Y の支持度で割った値

lift(コーヒー→パン)= conf(コーヒー→パン)/supp(パン)= (3/4)÷(5/8) = 6/5 = 1.2
lift(コーヒー→弁当)= conf(コーヒー→弁当)/supp(弁当)= (3/4)÷(7/8) = 6/7 ≒ 0.857

リフト値が大きいパンはコーヒーと一緒に売れている割合が高く，リフトが小さい弁当はコーヒーと一緒ではない割合が高い。このことから，コーヒーを買った人の確信度はパンと弁当は同じであるが，よりコーヒーと一緒に買われているのはパンであり，コーヒーを買う人に勧めることが適切なのはパンの方である。商品Xと組み合わせた場合の比率が高いときリフト値は1より大きくなり，リフト値が1を下回るときは推奨する根拠となりにくいと考えられる。実際には商品は膨大な種類があるため，全ての組み合わせを考えることは計算量が多くなり，計算時間も現実的な時間で収まらなくなる。そこで，一定の基準以上の支持度や確信度の場合だけ計算するなどの工夫が必要である。このような工夫により計算量を減らす方法をアプリオリアルゴリズムという。

演習 4　EXERCISE

購入データの例で図表11，パンを買った人に薦めるものを，支持度，確信度，リフト値を基に検討してみましょう。

【参考文献・参考サイト】
●「政府統計の総合窓口(e-Stat)」https://www.e-stat.go.jp/
●「UCI Machine Learning Repository」http://archive.ics.uci.edu/ml/datasets/Wholesale+customers
●「Python でデータサイエンス」
https://pythondatascience.plavox.info/scikit-learn/%E3%82%AF%E3%83%A9%E3%82%B9%E3%82%BF%E5%88%86%E6%9E%90-k-means
●「東京大学のデータサイエンティスト育成講座」塚本邦尊，山田典一，大澤文孝 著　中山浩太郎 監修　松尾豊 協力　マイナビ出版(2019)

学習活動と展開

学習活動の目的

- 教師なし学習のアルゴリズムの考え方を理解する。
- 教師なし学習のアルゴリズムを活用して，大量のデータをクラスタリングしたり，アソシエーション分析をしたりする技能を身に付ける。
- 教師なし学習によるクラスタリングをどのような場面で活用できるか考えられるようにする。

学習活動とそれを促す問い

	問　い	学習活動
展開 1	教師なし学習の考え方を理解しよう。	少量のデータを用いて，教師なし学習のアルゴリズムにより，分類したりアソシエーションルールを見つけたりする。
展開 2	教師なし学習によりデータを分析してみよう。	大量のデータを用いて，教師なし学習によりデータを分析する。
展開 3	教師なし学習を用いて，できることを考えよう。	学習したアルゴリズムを適用できる場面を考える。

展開 1	
問　い	教師なし学習の考え方を理解しよう。
▼	
学習活動	少量のデータを用いて，教師なし学習のアルゴリズムにより，分類したりアソシエーションルールを見つけたりする。
▼	
指導上の留意点	数学的な内容に深入りせず，アルゴリズムの考え方が理解できるようにする。

展開 2

問　い	教師なし学習によりデータを分析してみよう。
学習活動	大量のデータを用いて，教師なし学習によりデータを分析する。
指導上の留意点	●大量のデータを扱うことができるよう，ファイルから読み取り，処理し，結果を得る一連のプログラムを活用する。 ●データから得られた結果を解釈できるよう，得られた結果を基に考えて，言葉で表現させる。

展開 3

問　い	教師なし学習を用いて，できることを考えよう。
学習活動	学習したアルゴリズムを適用できる場面を考える。
指導上の留意点	分類した結果の活用の方法を考えたり，似たようなデータが得られる場面を考えたりできるようにする。

まとめ

まとめ	教師なし学習によるデータの分類の方法とその活用場面について整理する。

学習

17 ニューラルネットワークとその仕組み

研修内容

研修の目的

- 生徒に「自律性」と「適応性」の観点からAIの定義を考えさせられるようになる。
- 機械学習，AI，ディープラーニングの言葉の違いとそれぞれの関連を生徒に示せるようになる。
- ニューラルネットワークの概念と仕組みを生徒に理解させられるようになる。特に「学習」について実習を交えた授業を展開し，生徒がより理解を深められる授業をできるようになる。
- 生徒とニューラルネットワークによる手書き文字認識等のサンプルプログラムを実行し，仕組みを確認し，ニューラルネットワークの有用性とAI技術の活用を考えられるようになる。

この学習項目で使用するプログラミング言語は Python です。

1 人工知能(AI)とは何か，AIの活用例

この学習項目では，人工知能(Artificial Intelligence，AI，以後AIと表記)の定義や活用事例からAIがどのようなものかを捉えることを目標とする。

AIは人によって想像するものが異なる幅広い意味を含んだ語であり，明確な定義はない。例えばAIのキーワードに自律性(Autonomy)と適応性(Adaptivity)がある。以下の活用例から自律性と適応性を確認する。

- 自動車の自動運転技術
- 写真に写っている顔への自動タグ付け
- Webの閲覧状況からのお薦め表示

自動車の自動運転では，突然前に障害物が飛び出してきたときやカーブした道等の様々な状況下で，自動車が自分で判断して適切にブレーキを踏みハンドルを切る。写真の自動タグ付けではポインターを指定せずに撮影した画像から顔を認識する。自律性とはこのように，人の判断なしに状況に応じて動作する能力である。適応性とは，大量のデータから特徴を見つけ出し状況判断ができる，あるいは与えられた正解データと新たなデータを照合することで自らのプログラムの精度を上げていくことができる(学習)能力である。前述の例ではWebの閲覧状況から閲覧者に合わせた情報を表示する。

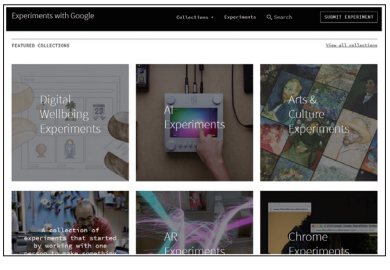
演習 1

EXERCISE

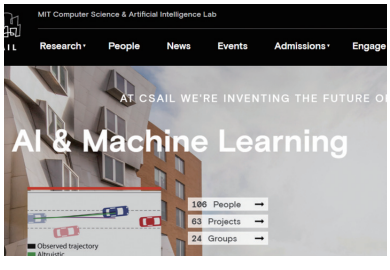
既に AI が使われている例を挙げます。いくつか試して更なる活用方法を考えましょう。
またこれらの技術の使用で注意すべき点について近くの人と共有しましょう。

AI の活用例

- CLARA: A NEURAL NET MUSIC GENERATOR：AIによる自動作曲と演奏，Claraの利用。(http://christinemcleavey.com/clara-a-neural-net-music-generator/)
- HUMAN OR AI (By mcleavey Posted August 27, 2018)：人間が作曲したか，AIが作曲したかの判断を試せる。(http://christinemcleavey.com/human-or-ai/)
- Experiments with Google：AI ExperimentsからAIを使った様々なアプリケーションを体験できる 図表 1。
- MIT Computer Science & Artificial Intelligence Lab：AIと機械学習の最新の研究内容が読める 図表 2。



図表 1 https://experiments.withgoogle.com/collection/ai

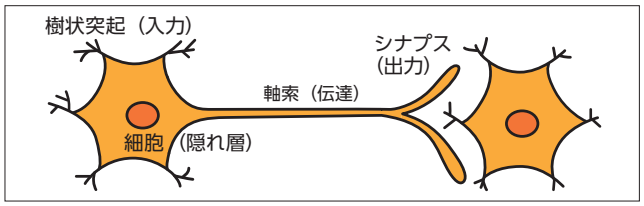


図表 2 https://www.csail.mit.edu/

2 ニューラルネットワークの由来

この学習項目では，ニューラルネットワーク，AI，ディープラーニング，機械学習の関連を整理する。ニューラルネットワークとは人間の神経細胞とその仕組みに似せて考えられたコンピュータの処理である。人間の神経細胞(ニューロン)は，受容体を持つ樹状突起と軸索，シナプスの組み合わせ図表3で構成され，シナプス間を神経伝達物質といわれる化学物質が伝達されることで成り立っている。

このモデルを使ったネットワークの原形が1943年に発表された。ニューラルネットワークは，膨大な計算量とその計算時間が大量であること，隠れ層の重み付けを決めるアルゴリズムの難しさから長らく非現実的とされた(冬の時代)。その間，人が特徴量を指定して行う機械学習が活用されるようになったが，人によるコンピュータの学習であり，限界が見られていた。その後，計算処理を分散して実行できるGPUの登場と，コンピュータの計算処理速度の向上により重み付けの調整が人の手から離れ，コンピュータが計算を繰り返すことで最適解を探すこと(自律学習)が可能となった。このコンピュータによる学習の繰り返しを多層ニューラルネットワークで表したものをディープラーニング(深層学習)という。これによりAIの予測分類精度がそれまでのものから劇的に上がり，ニューラルネットワークを基にしたAI技術が大きく注目されるようになった図表4。



図表3　人間の神経細胞（ニューロン）

図表4　AIと機械学習，ディープラーニング，ニューラルネットワークの関係

3 ニューラルネットワークの概念

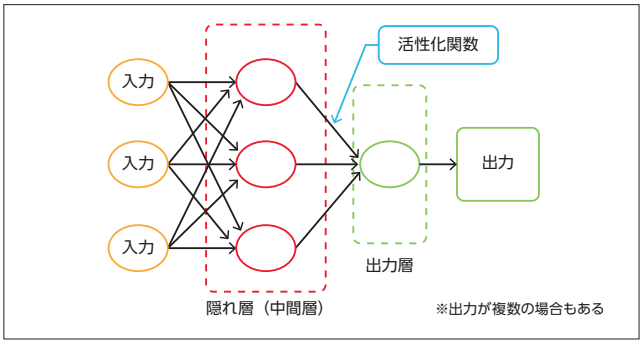
この学習項目では，ニューラルネットワークの概念と仕組みを演習しながら学び，理解することを目標とする。

ニューラルネットワークは大きく分けて「入力」「隠れ層(中間層)」「出力層」で構成されている図表5。隠れ層を次の入力として別の隠れ層に渡していくことを繰り返すと図表6，複数の層の隠れ層をもつ深層学習(Deep Learning)となり，より複雑な出力に対応できる。一般に隠れ層と出力層を合わせてネットワークの層の数を表す。

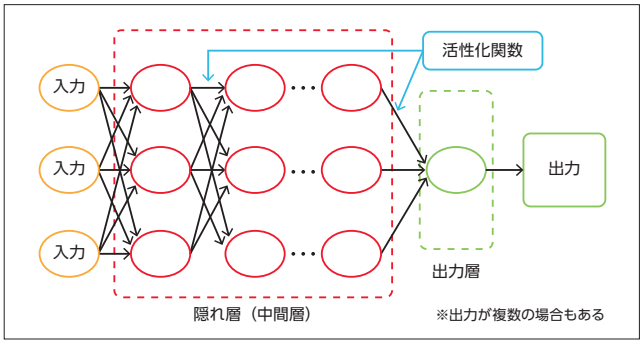
各ニューロンでは下記のように計算を行っている。

(入力)　×　(重み付け)　＋　(バイアス)

その後，各ニューロンの結果を活性化関数に入れて(発火という)出力とする。



図表5　入力，隠れ層，活性化関数，出力



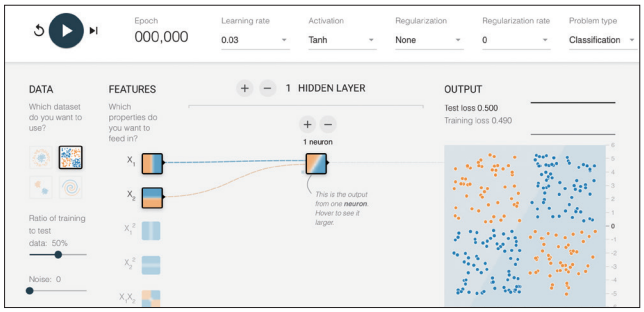
図表6　多層構造のニューラルネットワークのイメージ

例えば手書き文字が4か9か認識したいとする。入力は手書き文字画像のピクセルごとのグレースケールの階調を値にして表した配列である。出力は手書き文字が4であれば0，9であれば1とする。各ピクセルの値に重み付けとバイアスの値を加えることでどちらかの出力値に寄せていく。最初の重み付けとバイアスはコンピュータがランダムに行い，学習の過程で自律的にそれぞれ修整していく。活性化関数は，隠れ層の計算結果が一定の値(閾値)を超えたときに大きな値を得られる関数である。具体的な活性化関数については次節で扱う。

演習 2 EXERCISE

TensorFlow Playgroundで，を複数組み合わせ，同じものを重ねて，FEATURES や HIDDEN-LAYER を操作して組み合わせることにより，オレンジ色の点とブルーの点の集まりを明確に分けることができるか，試してみましょう。

TensorFlow公式サイトでは，デモアニメーションを実際に操作し，隠れ層を深くしていくことによる学習効果を確かめられる図表7。このサイトは，オレンジの点とブルーの点の集まり(画面左のDATAで増やしたり減らしたりできる)を学習に使う入力データとし，画面中央左側の(隠れ層の様々なフィルターパターンを表しているFEATURES)と隠れ層の数(HIDDEN LAYERの左の＋－ボタンで数の増減を調整)を組み合わせて，画面左上の実行ボタンを押すと，画面右のOUTPUTにオレンジ色の領域とブルーの領域に分かれていく学習の様子が確認できる。



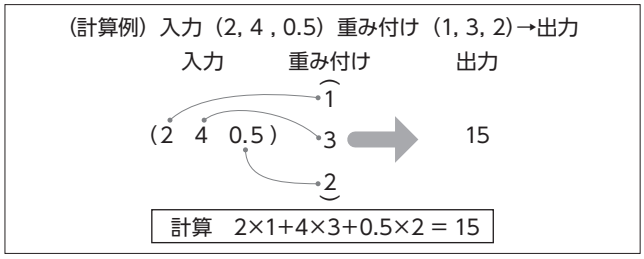
図表 7　TensorFlow サイト画面例　「TensorFlow」 http://playground.tensorflow.org

4 ニューラルネットワークの仕組み

この学習項目では，ニューラルネットワークの仕組みを計算とプログラムで表現し，ニューラルネットワークの計算や活性化関数の働きを理解しよう。

(1) ニューラルネットワークの計算

入力値が複数の場合や隠れ層が多層になったときは行列の乗算(内積，ドット積)を行う図表8。



図表 8　行列計算

演習 3 EXERCISE

ニューラルネットワークの計算を確認しましょう。

入力 (0.3，0.8，1.2，0.7)，重み付け (1，2，1，1)，バイアス 0.2 のときの出力結果を求めましょう。

演習3で行った計算をPythonのプログラムで表現する。

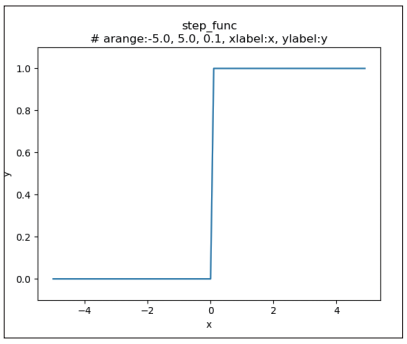
```
01 import numpy as np
02 X = np.array([0.3, 0.8, 1.2, 0.7])
03 W = np.array([1, 2, 1, 1])
04 B = 0.2
05 A = np.dot(X,W) + B
06 print(A)
```

出力結果は 4

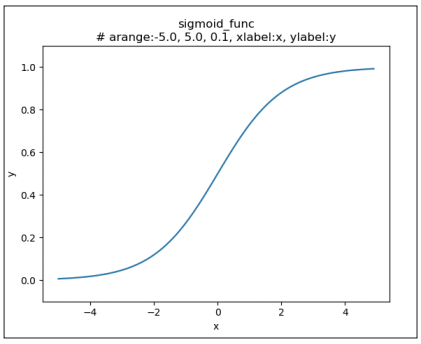
Xは入力，Wは重み付け，Bはバイアスを表す。1行目では，Pythonで計算処理ができるパッケージnumpyをnpとして使えるようにしている。2～4行目で演習3と同じ値をそれぞれに用意している。5行目でnumpyパッケージにあるdot関数を使って行列計算をしてバイアスを加え，出力Aとしている。最後にprint文で計算結果を出力している。この隠れ層での計算の後，活性化関数を用いる。

(2) 活性化関数

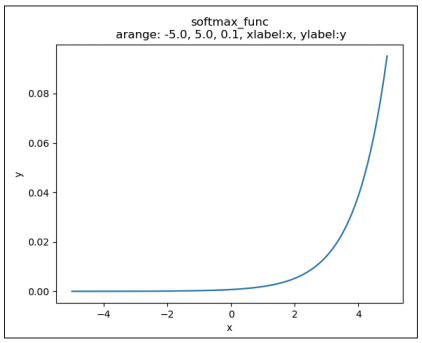
各ニューロンの出力に作用させる関数を活性化関数という。活性化関数は各ニューロンの結果の正確さを確率で表したものである。論理演算のように入力に対して0か1を返すような2値に分類ができる場合はステップ関数図表9を用いる。はっきりとした分類ができる分類器をパーセプトロンという。ニューラルネットワークでは論理演算のように2値に分けられるとは限らない。猫か犬かのような2値分類の場合には，シグモイド関数図表10やReLU（ランプ）関数を使い，猫か犬か鳥かといった複数に分類する場合はソフトマックス関数が多く用いられている図表11。



図表 9　ステップ関数



図表 10　シグモイド関数



図表 11　ソフトマックス関数

5 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で学習する際に必要な技術を確認する。

(1) 損失関数

ニューラルネットワークの性能の良し悪しを測る指標を損失関数という。損失関数では予測データと教師データとの誤差を表し，この値が0に近いほど性能がよい。ニューラルネットワークの学習は，この損失関数の結果が最小となるような重みを探していくことである。学習に使われる計算については次項「(2)勾配降下法」で説明する。損失関数には状況に応じて様々な関数を使う。2乗和誤差や交差エントロピー誤差が有名である。2乗和誤差とは学習結果と教師データの差分を2乗した合計を2で割ったものである。

$$E = \frac{1}{2}\sum_{k=1}^{n}(y_k - t_k)^2$$

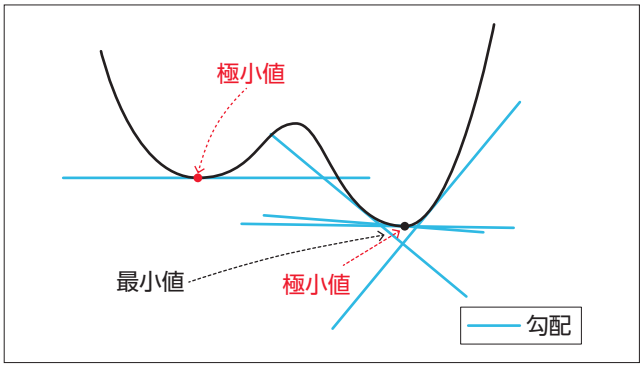
2乗和誤差を表す式

(2) 勾配降下法

ニューラルネットワークは学習を繰り返し，損失関数の結果が最も小さくなる重みやバイアスを探している。ニューラルネットワークの学習では，最適な重みやバイアスを勾配降下法(最急降下法)によって探すことが多い図表12。

勾配降下法は，勾配を使って損失関数の値を最も減らす方向を求める方法である。勾配はある地点での各重みにおける損失関数の傾きである。全ての重みで勾配が0となる地点を極値といい，局所的な最小値(極小値)の候補となる図表13。ただし極小値が全体の最小値となるとは限らないため，地点を変えながら繰り返し極小値を探り，最小値を求めている。

図表 12　勾配降下法



図表 13　極値

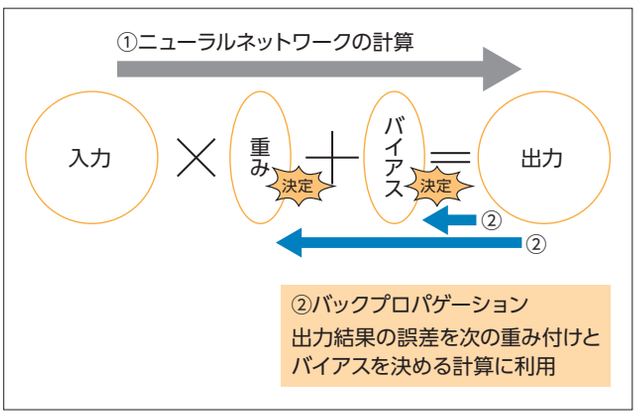
(3) バックプロパゲーションによる学習

バックプロパゲーション(誤差逆伝播法)は前項(2)勾配降下法をもとにしたニューラルネットワークの学習で最適な重みとバイアスを探す代表的な手法である図表14。

(4) オーバーフィッティングとその防止

オーバーフィッティング(過学習)とは学習の際に特定のデータにだけ過剰に対応し，学習に用いていない他のデータでは正しくならない状態のことである。

これらの技術をPython等のプログラミング言語で実装することもできるが，ここでは，ニューラルネットワークを容易に構築するためにNeural Network Console（SONY）を利用する。Neural Network Consoleには，Webブラウザで動作するクラウド版とWindowsアプリ版がある。このサイトを活用して，学習15でも扱った手書き数字データセットMNISTをニューラルネットワークで学習させてみよう。



図表 14　学習とバックプロパゲーション

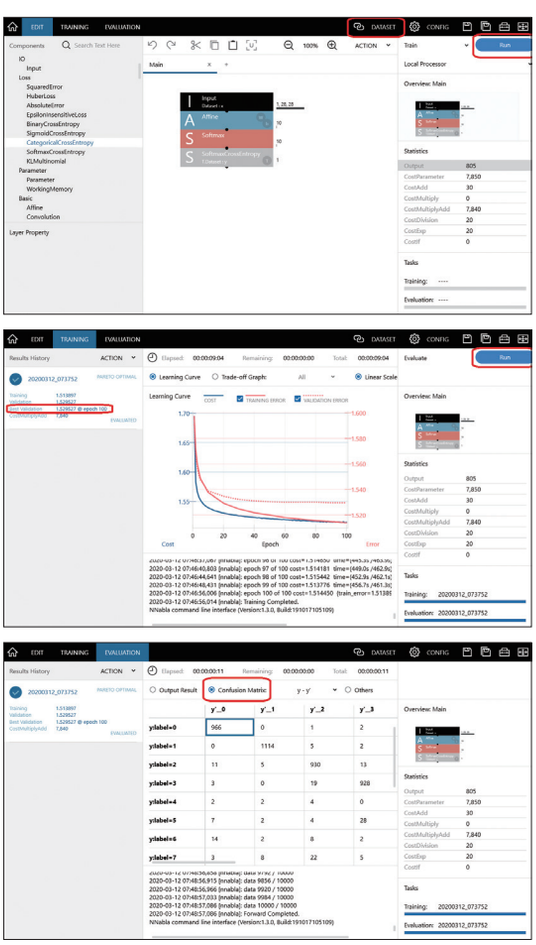
演習 4

EXERCISE

MNIST(4と9の手書き文字)の認識(2値分類)をニューラルネットワークで学習してみましょう。ここでは，Neural Network Console（https://dl.sony.com/ja/　アカウント取得が必要）を使って演習します。

最初に，使い方を学ぶためにサンプルを動かしてみよう。HOMEから01_logistic_regressionを選ぶ。これは，MNISTデータの4と9だけからなるデータから，2数の判別をロジスティック回帰と呼ばれる分類法で分類する例である。1層のニューラルネットワークで，INPUTは入力，AFFINは重みwを掛けbを加えるアフィン変換を表す。活性化関数はシグモイド関数，損失関数は交差エントロピー誤差を使用したニューラルネットワークが用意されている。

右上のRunのボタンを押すとトレーニング(訓練，学習)が始まる。Epochは最適化の繰り返しの世代数，すなわち勾配降下法によりバックプロパゲーションしながらwやb等のパラメータの修整を行っている。Costは損失関数の出力である。次第にCostが下がってきているので，最適化がうまくいっているようである。トレーニング終了後再びRunを押すとテストデータを用いた評価(検証)である。Confusion Matrixを選択すると，予測と正解の関係の行列が表示される。Accuracyは正答率である。95.2%の正答率なので，単純なニューラルネットワークであるのによい結果である。



図表 15　(上) ネットワークの編集　(中) トレーニング
(下) 評価 (混同行列，Confusion Matrix)

演習5

EXERCISE

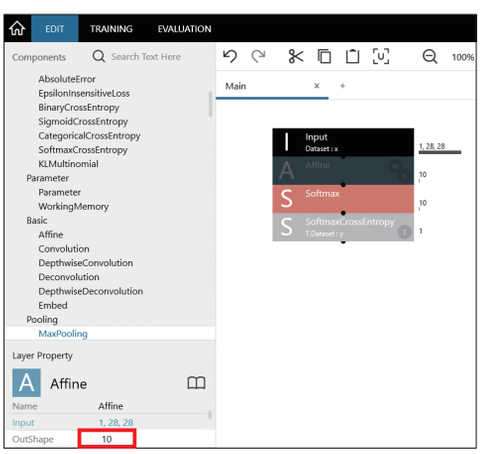
自分で10種類の文字を判別するニューラルネットワークを作成してみましょう。多層のニューラルネットワークでなければ精度の高い判別は難しいが，練習のために単純なニューラルネットワークを構築してみましょう。

HOMEからNew Projectを選択し，EDITタブに移動する。次に上のバーにあるDATASETをクリックし，左のTrainingを選択し，URI欄をクリックし，mnist_training.csv（60,000件）を選択する。同様にValidationにmnist_test.csv（10,000件）を選択して，使用するデータを設定する。次にEDITタブから，左のペインにあるINPUT，AFFINE，Softmax，SoftmaxCrossEntropyを中央のEDITペインにドラッグし，接続する。今回はMNISTの10種類の文字の分類なので，AFFINEをクリックし，画面左下のOutShapeを10に設定する。これで隠れ層1層と出力層1層の2層のニューラルネットワークが完成する図表16。

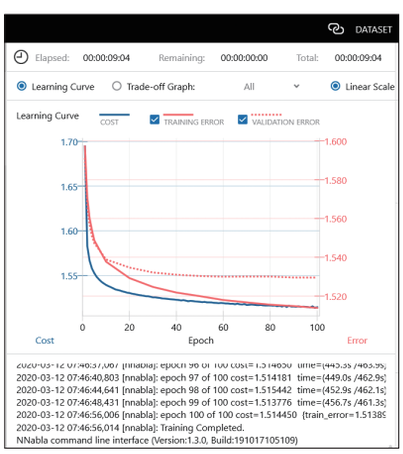
ここで右のペインのTrainのRunボタンをクリックするとTraining（学習）が始まる。Validation Errorを表すグラフが順調に減少していれば，学習は成功している。もし途中で上昇するならば，最適化の回数を調整する必要がある図表17。

学習が終了したら，右ペインのEvaluateをRunすると，学習結果をもとに評価が始まる。評価が終わったら，Confusion Matrixを選択すると，10×10の混同行列が表示される。Accuracyはここでは，0.9338である。つまり，10,000件のテストデータで約93%の正答率だったということである図表18。

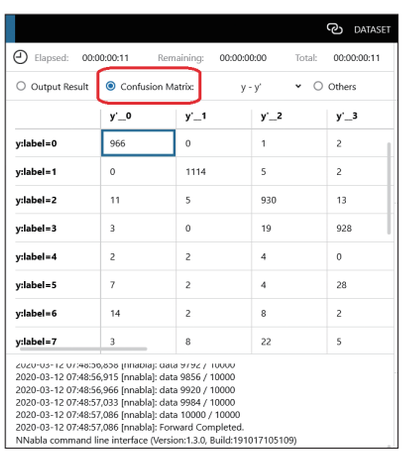
今回は隠れ層が単層のニューラルネットワークで試してみたが，これを多層にすることによりディープラーニングを作ることができ，より正答率の高い学習が可能である。作成したネットワークの活性化関数や損失関数を変えて学習させてみよう。2値分類か多値分類かで活性化関数や損失関数の選び方が変わることに注意しよう。



図表 16　ネットワークの編集



図表 17　トレーニング



図表 18　評価（混同行列）

【参考文献・参考サイト】

●「ゼロから作る　Deep Learning」斎藤康毅 著　オライリージャパン（2016）
●「新版　数理計画入門」福島雅夫 著　朝倉書店（2011）
●「高校数学でわかるディープラーニングのしくみ」涌井貞美 著　ペレ出版（2019）
●「直感　Deep Learning」Antonio Gulli，Sujit Pal 著　大串正矢，久保隆宏，中山光樹 訳　オライリージャパン（2018）
●「Neural Networks:A Visual Introduction for Beginners」Michael Taylor 著　Blue Windmill Media（2017）
●「Deep Learning（Adaptive Computation and Machine Learning series）」Ian Goodfellow, Yoshua Begio, Aaron Courville 著　The MIT Press（2016）
●「Neural Network Console」https://dl.sony.com/ja/
●「TensorFlow」http://playground.tensorflow.org
●「HUMAN OR AI」http://christinemcleavey.com/human-or-ai/
●「Experiments　with Google」https://experiments.withgoogle.com/collection/ai
●「MIT Computer Science & Artificial Intelligence Lab」https://www.csail.mit.edu/

学習活動と展開

学習活動の目的

- AI を理解する。
- AI の仕組みの基本の一つとしてニューラルネットワークを理解した上で AI の活用を考える。

学習活動とそれを促す問い

	問　い	学習活動
展開 1	AI はどのようなことに使われているかを調べ，試して，AI とは何か考えてみよう。	研究事例や実用化されている AI 技術を体験し理解を深める。その利用方法や課題を考える。
展開 2	ニューラルネットワークが何かを知ろう。	ニューラルネットワークの概念と仕組み，それを支える技術や手法を理解する。
展開 3	ニューラルネットワークを活用しよう。	実際にニューラルネットワークのプログラムを実行し，その仕組みの理解を深める。

展開 1

問　い

AI はどのようなことに使われているかを調べ，試して，AI とは何か考えてみよう。

学習活動

- ペアでどのようなものを AI と捉えているか確認後「自律性」「適応性」「学習」の用語を学ぶ。
- 活用事例や研究事例を複数調べて実際に試し，AI の得意なこと，課題，将来性をレポートにまとめる。
- 情報の特性を踏まえた上で，情報をどのように扱うかについて，自分の考えをまとめる。

指導上の留意点

- AI の様々な活用を知った後で, AI の得意なこと, 課題, 将来性について「自立性」「適応性」「学習」をキーワードとして示すようにする。
- AI についての考えができた後に，AI が実用的に認められるまでの課題がなぜ解消されたかを認識させる。
- 現在の AI の実用性と将来性を技術面，倫理面の両面から考えるよう生徒に促す。

展開 2

問い

ニューラルネットワークの概念，AI とニューラルネットワークの関連性を知り，ニューラルネットワークの仕組みを理解しよう。

学習活動

- 人間のニューロンの仕組みとニューラルネットワークの概念を確認する。
- AI，機械学習，深層学習，ニューラルネットワークの違いと関連を知る。
- 簡単な計算やプログラムを使った演習により,ニューラルネットワークの仕組みを理解する。

指導上の留意点

- ニューラルネットワークの AI との関連性を認識させる。
- AI，機械学習，深層学習の用語の混同や誤解に注意する。
- ニューラルネットワークの各層の造りや損失関数による学習の判定は簡単な演習により体験的に理解を促す。
- バックプロパゲーションについては概念的に扱い，数式やプログラムを深追いさせない。

展開 3

問い

ブロックプログラミングで手書き文字認識をするニューラルネットワークを動かしてみよう。

学習活動

ブロックプログラミングで実際にニューラルネットワークによる手書き文字認識を体験し，層の概念への理解を深めニューラルネットワークの威力を感じる。

指導上の留意点

- 生徒が実際にニューラルネットワークを動かして体験的に理解を促すことを重視する。
- 余裕がある生徒には自分でデータを用意させ，サンプルプログラムに手を加えさせる。

まとめ

まとめ

- AI が何かを説明させ，授業実施前より正しく理解しているか振り返る。
- ニューラルネットワークの概念と仕組みを振り返る。
- これからの AI の課題（技術面，倫理面）と活用をこれまでの学習を基にレポートにまとめさせる。

学習

18 テキストマイニングと画像認識

研修内容

研修の目的

- 機械学習やニューラルネットワークの応用技術について理解させる授業ができるようになる。
- 物体検出や画像認識技術に関して，アプリケーションを作成し，活用できる授業ができるようになる。
- テキストマイニングに関して，感情分析や類似度などの既存のリソースや技術を用いて分析できる授業ができるようになる。

この学習項目で使用するプログラミング言語はＲです。

1 データサイエンスの活用

これまで，いくつかの機械学習の演習を紹介してきたが，訓練データを集めて，学習させ，予測に使っていく過程は理解できたかと考える。実際に多くのデータで学習をあらかじめ行っておき，その結果を利用できるようにする仕組みも多くある。

例えば，機械学習に対応したソフトウェアによる学習済モデルの活用である図表１。ウェブカメラで写真を撮り，あらかじめ学習されている結果を使い，写真と似ている著名人との類似度を表示させたものである。学習済モデルを利用することにより，1，2行でこのようなアプリを作成することもできる。

図表1　画像認識の例

このような学習結果を蓄積するために様々な試みが行われている。reCAPTCHAという仕組みは，インターネット上の認証が，プログラムやボットなどで突破されないように，人間であることを確認するための手段として用いられている図表２。その一方で，この人間が入力した結果はOCR（Optical Character Reader）の読み取り結果と照合され，より正確に様々な文書を読み取れるようにOCRの精度を上げることに利用している。例えば，New York Timesの過去の記事やGoogle Booksの画像データの読み取りもこの結果が利用されている。機械学習やニューラルネットワークで訓練されたデータは世界中で日々更新され，ユーザーはリポジトリとして活用することができる。

図表2　reCAPTCHAの例

本学習では，利用環境を整えるための手順が多く，研修内容としてやや高度なものもあるが，データサイエンスの活用として扱っていただければ幸いである。

演習 1

EXERCISE

Amazon Mechanical Turk (https://aws.amazon.com/jp/mturk/faqs/) のようにコンピュータだけではできないような仕事をプログラム（アプリ）と人間の仕事で補っている例を考えてみましょう。そのような中で，人間が目的を知って意識的に行っている仕事には何があるかを調べるとともに，無意識でやらされている仕事には何があるかについても考えましょう。

2 MeCabを利用したテキストマイニング

「情報Ⅰ」の教員研修用教材と同様に，青空文庫の作品を題材にテキストマイニングの演習を行う。ここでは，Word2vecを用いて，単語の類似度を探る。また，感情分析に関しても少し触れていく。Word2vecとは，テキスト処理のためのニューラルネットワークの応用技術である。テキストを入力することにより，その類似度を学習し，ベクトルとして返す。これにより，単語の類似度を計算し，様々な分析に用いることができる。この実習では，形態素解析を行うため，あらかじめMeCabのインストールが必要である。

演習 2

EXERCISE

青空文庫の夏目漱石作「坊っちゃん」を題材に，Word2vec を用いて，分析しましょう。

準備 この演習ではRの32ビット版の使用を推奨(他の環境では，RMeCabパッケージでエラーが出る可能性がある)。

最初にWord2vecパッケージをgithubからソースをダウンロードし，ビルドする。

```
install.packages("devtools")
devtools::install_github("bmschmidt/wordVectors")
```

後で行う感情分析のための感情辞書をダウンロードしておく。

```
dic <- read.table(
  "http://www.lr.pi.titech.ac.jp/~takamura/pubs/pn_ja.dic",
  sep = ":", stringsAsFactors = FALSE, fileEncoding = "CP932")
```

感情辞書は，外国語に関しては複数有名なものがあるが，今回は，日本語辞書として，東京工業大学のPN Tableを使用する。

実習

RMeCabを読み込む。最初の行がインストール，2行目が読み込みを表す。

```
install.packages("RMeCab", repos = "http://rmecab.jp/R")
library(RMeCab)
```

RMeCabの作者でもある徳島大学の石田基広教授のAozoraパッケージを読み込み，夏目漱石の「坊っちゃん」を読み込む。読み込み方は，「情報Ⅰ」教員研修用教材の196ページを参考にするとよい。

```
source("http://rmecab.jp/R/Aozora.R")
bochan<-Aozora(
  url="https://www.aozora.gr.jp/cards/000148/files/752_ruby_2438.zip")
```

次に，データの整形等に必要なパッケージを読み込む。これらのパッケージがまだインストールされていない場合は，install.packages("dplyr") のようにインストールしておく。

```
library(dplyr)
library(purrr)
library(magrittr)
library(wordVectors)
```

それでは，データの加工をしていこう。

```
tf <- tempfile()
RMeCabText(bochan) %>% map(function(x)
  ifelse((x[[2]] %in% c("名詞", "形容詞", "動詞")) &&
           (!x[[3]] %in% c("数", "非自立", "代名詞", "接尾")) &&
           (x[[8]] != "*"), x[[8]],  "")) %>%
  paste(" ", collapse = "") %>%
  write(file = tf, append = TRUE)
```

ここまでで，「坊っちゃん」本文から，名詞と形容詞，動詞を取り出し，数や非自立語等を取り除き，「分かち書き」をしている。分かち書きした文章は，tfというファイル名で保存してあるので，file.show(tf)とすると，見ることができる。

次に，文章をWord2vecで学習し，モデルを構築する。少し処理に時間がかかる。

```
model_bochan <- train_word2vec(tf,"tf.bin", min_count = 2, force=T)
model_bochan
```

単語の類似度を調べてみよう。

```
model_bochan %>% closest_to("赤")
```

出力結果

```
      word similarity to "赤"
1     赤                1.0000000
2     パイプ            0.8395877
3     優しい            0.8393746
4     馴染              0.8331806
5     シャツ            0.8311088
6     文学              0.8293475
7     迷惑              0.8254646
8     ターナー          0.8189216
9     見せびらかす      0.8150383
10    雑誌              0.8144845
```

登場人物の赤シャツや山嵐の性格や特徴が見える。closest_to("山嵐")とすると，山嵐の様子が見えるだろう。モデルの要素は足し算や引き算ができる。これに関しては，参考文献の「Word2Vecで『おじさん』と『お兄さん』を比較してみた」が分かりやすい。

```
model_bochan %>%
nearest_to(model_bochan[["マドンナ"]] - model_bochan[["シャツ"]])
```

出力結果

```
      先生      バッタ        親切          違う      イナゴ        居る      マドンナ
0.4540880 0.4784261 0.5209302 0.5348532 0.5349468 0.5463454 0.5527229
        女        上品          声
0.5556836 0.5598718 0.5713779
```

次に簡単な感情分析をやってみよう。Word2vecでも感情の主な用語との距離を求めることにより，感情分析が可能であるが，ここでは，既にある感情辞書で分析してみよう。
辞書のV1とV4の列だけにする。V1をTERMという名前に変更する。

```
01  dic2 <- dic %>% select(V1, V4) %>% rename(TERM = V1)
02  dic2 %<>% distinct(TERM, .keep_all = TRUE)
03  head(dic2)
```

出力結果

```
     TERM       V4
1    優れる     1.000000
2    良い       0.999995
3    喜ぶ       0.999979
4    褒める     0.999979
5    めでたい   0.999645
6    賢い       0.999486
```

ポジティブな単語は1に近く，ネガティブな単語は-1に近く設定されている。

次に「坊っちゃん」から名詞と形容詞を取り出し，数や接尾語を取り除く。bochan2は単語の頻度表になっている。これに辞書にある単語について，V4のスコアを加える。

```
01  bochan2 <- docDF(bochan, pos = c("名詞","形容詞"),type = 1)
02  bochan2 %<>% filter(!POS2 %in% c("数","接尾","非自立"))
03  bochan3 <-bochan2%>%left_join(dic2)
```

ネガティブな言葉やネガティブな用語については，次のようにすると表示することができる。

```
01  bochan3 %>% select(TERM, V4) %>% arrange(V4) %>% head(10)
02  bochan3 %>% select(TERM, V4) %>% arrange(desc(V4)) %>% head(10)
```

次に，ポジティブな単語とネガティブな単語の種類を合計する。ここでは出現頻度は計算しない。また，少し工夫すると，出現頻度も含めた合計も計算できる。

```
01  bochan3 %>% summarize( sum (V4 > 0, na.rm = T))
```

出力結果

```
  sum(V4 > 0, na.rm = T)
1                    257
```

```
01  bochan3 %>% summarize( sum (V4 < 0, na.rm = T))
```

出力結果

```
  sum(V4 < 0, na.rm = T)
1                   2015
```

演習 3

EXERCISE

感情分析の結果，ネガティブな用語が多い理由について，この結果を基に話し合ってみましょう。

3 TinyYOLOを利用した物体検出

「YOLO（You Only Look Once）」とは，2016年にJoseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadiの論文"You Only Look Once: Unified, Real-Time Object Detection"によって発表されたリアルタイム物体検出・認識アルゴリズムである図表3。ニューラルネットワークや分類などの考え方を複合させ，従来は画像をトレース（走査）して物体を検出していたものを，名前の通り，"1回見ただけ"で検出するようにしたアルゴリズムである。現在では，医療画像分析などにも応用されているアルゴリズムである。アルゴリズムの詳細は，ここでは説明しないが，これを利用して，写真上の物体の検出を行ってみたい。

図表3 「YOLO」https://pjreddie.com/darknet/yolo/

演習4 EXERCISE

それぞれが用意した写真について，Tiny YOLO アルゴリズムを用いて，写真上にある物体を検出・識別するプログラムを作成してみましょう。

準備

最初にYOLOアルゴリズムを実行するためのライブラリimage.darknetをインストールする。
このライブラリは，標準のCRAN（The Comprehensive R Archive Network）からインストールするのではなく，githubからソースをダウンロードし，ビルド（コンパイル）するので，手元の環境にCなどの開発環境が既に導入されていなければならない。また，Rの中からビルドするため，devtoolsライブラリも必要である。

```
01 install.packages("devtools")
02 devtools::install_github("bnosac/image",
     subdir = "image.darknet", build_vignettes = TRUE)
```

上記を入力し，しばらく待つ。途中でエラーが出たときは，解消して再実行する必要がある。ネットワーク等の関係で，githubへのアクセスができない場合は，https://github.com/bnosac/image からzipファイルとしてダウンロードしたファイルからビルドする。

```
01 devtools::install_local("c:/Users/shinya/Desktop/image-master.zip",
     subdir = "image.darknet", build_vignettes = TRUE)
```

物体検出

長い準備が終わったら，いよいよ物体検出である。image.darknetライブラリを読み込む。

```
01 library(image.darknet)
```

次に，少し長い命令を入力する。本教材がPDFで提供されていれば，下記の命令をそのままコピー＆ペーストしてもよい。

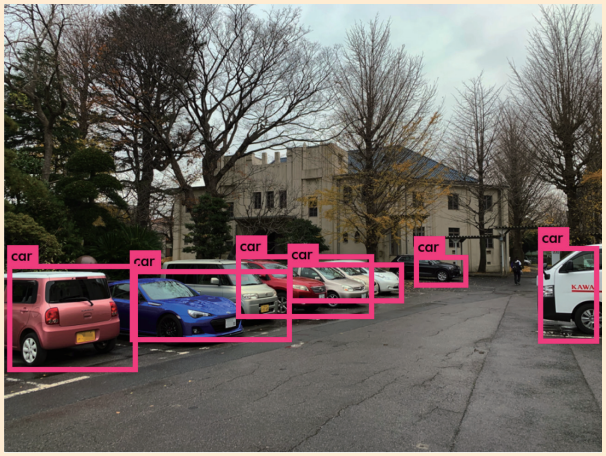
```
01 yolo_tiny_voc <- image_darknet_model(type = 'detect',
   model = "tiny-yolo-voc.cfg",
   weights = system.file(package="image.darknet","models",
   "tiny-yolo-voc.weights"),
   labels=system.file(package="image.darknet","include","darknet",
   "data","voc.names"))
```

ここでは，Tiny YOLOの設定をしている。それでは，対象となる画像を読み込もう。今回読み込む写真は，自分で用意してみよう。ここでは，次の写真をもとに実行する 図表4。ファイル名は，school.pngとしてある。どこまで，物体検出，認識ができるだろうか。（出力は省略）

図表4　物体検出，認識に利用する写真例

```
01 x <- image_darknet_detect(
   file = "c:/Users/shinya/Desktop/school.png",
   object=yolo_tiny_voc,threshold = 0.20)
```

六つの物体を検出したようだ。全て自動車である。threshold（閾値）を変更すると，検出する物体の数も変わってくる。試してみよう。結果は，Rのワーキングディレクトリにpredictions.pngという名前で保存される。作成された結果を示す 図表5。



図表5　物体検出結果

［参考文献・参考サイト］

- 「R+RMeCab で感情分析」 https://qiita.com/rmecab/items/b1a55a0e3a0a8637a461
- 「R で日本語テキストに word2vec」 https://qiita.com/rmecab/items/c165a67a2f02e76e8390
- 「Word2Vec で「おじさん」と「お兄さん」を比較してみた」 https://www.pc-koubou.jp/magazine/9905
- 「Object detection in just 3 lines of R code using Tiny YOLO」 https://heartbeat.fritz.ai/object-detection-in-just-3-lines-of-r-code-using-tiny-yolo-b5a16e50e8a0?gi=4e477020ed3f
- 「You Only Look Once: Unified, Real-Time Object Detection」 Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi 著（2016） https://arxiv.org/abs/1506.02640

学習活動と展開

学習活動の目的

- 機械学習やディープラーニングで訓練された結果の資源はどんなところにあるのか理解できる。また，そのような資源を活用したサービスやアプリにはどのようなものがあるか理解できる。
- テキストマイニングに関して，自分で選んだ文章に関して感情分析を行うことができる。
- 画像認識に関して，YOLO 等の方法を使って，物体検出や認識を行うことができる。
- データ分析を行った結果を基にして，様々なサービスを考えることができる。

学習活動とそれを促す問い

	問　い	学習活動
展開 1	機械学習やディープラーニングの応用技術には何があるだろうか。	機械学習やディープラーニングの学習結果や応用技術について調べる。
展開 2	テキストマイニングを行うことで何が分かるか。	テキストマイニングについて，感情分析などを含め，文書の内容を分析し，解釈する。
展開 3	画像認識を行うことで何ができるだろうか。	画像認識を行うためにどのような技術があり，どのように活用されているか調べる。

展開 1	
問　い	機械学習やディープラーニングの応用技術には何があるだろうか。
▼	
学習活動	機械学習やディープラーニングの学習結果や応用技術について調べる。
▼	
指導上の留意点	活動につまずいている生徒に，必要に応じてヒントを与え，生徒自らが機械学習やその結果の活用について理解できるよう促す。

展開 2

問　い	テキストマイニングを行うことで何が分かるか。
学習活動	テキストマイニングについて，感情分析などを含め，文書の内容を分析し，解釈する。
指導上の留意点	テキストマイニングの手法を理解するだけでなく，その結果から何が読み取れるのか解釈できるように促す。

展開 3

問　い	画像認識を行うことで何ができるだろうか。
学習活動	画像認識を行うためにどのような技術があり，どのように活用されているか調べる。
指導上の留意点	画像認識がどのような場面で用いられているのか，自分の身近な生活の場だけでなく，医療，工業，その他の専門分野への応用を積極的に考えられるように促す。

まとめ

まとめ	機械学習やディープラーニングが社会にどのように活用され，今後どのような応用が考えられるか，生徒自らが積極的に考えられるようになり，データ分析の有用性を理解できるようになる。

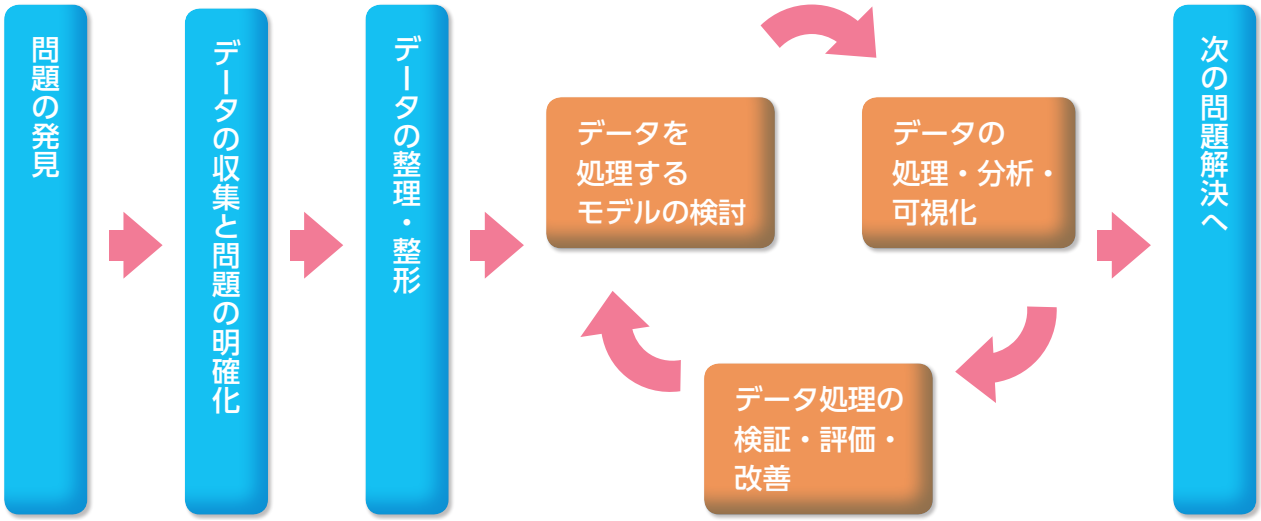
全体を通じた学習活動の進め方

全体を通じた学習活動の目的

●データの分析のための様々な手法を通して，目的に応じてデータを収集，整理，整形し，データに基づく現象のモデル化やデータの処理，解釈・表現を行い，モデルを評価する力を育成する。

全体を通じた学習活動の流れ

「情報Ⅰ」の (4) 情報通信ネットワークとデータの活用と同様に，自然現象や社会現象についてデータを用いた解決について検討し，データの収集，整理・整形，モデル化，処理，分析，評価，改善という一連の学習活動を行う。



情報とデータサイエンスについての授業を計画するにあたり，

❶データの収集
❷データの整理・整形
❸データのモデル化・処理・分析・可視化
❹データ処理の検証・評価・改善
❺発展的な学習

について，以下の項目と内容に留意して行うことが望ましい。

データの収集 1	●データの信憑性や信頼性の検討が必要であることの理解を深める。 ●問題解決に必要なデータの収集方法について理解できるようにするとともに，オープンデータや Web 上のデータの収集を体験できる活動を行う。 ●収集したデータを蓄積・管理するための仕組みについて理解を深め，関係データベースや NoSQL の利用を体験できる活動を行う。
データの整理・整形 2	●データの形式の違いを理解して，データの整理・整形を体験できる活動を行う。データの整理に関しては，欠損値や異常値の扱い，ワイドフォーマットやロングフォーマットの特性を理解する。 ●データを目的に応じた処理しやすい形式にするために，データクリーニングや整理・整形の手法を体験的に理解させる。
データのモデル化・処理・分析・可視化 3	●データ分析の目的に応じて，重回帰分析・主成分分析・分類・クラスタリングなどからモデルを検討し，修正していくことを体験する。 ●大量のデータを用いて，実際にデータを処理・分析する活動を行う。
データ処理の検証・評価・改善 4	●複数のモデルを用いて，その正答率 (正確度) を比較するなど，適切なモデルを選択する活動を行う。 ●テストデータを用いて正答率を求めることなどにより評価し，適合不足や過剰適合 (過学習) が生じることを体験的に理解できる活動を行う。
発展的な学習 5	●手書き文字の認識や文章の感情分析，画像認識するなどの処理について考え，実際に体験するといった学習活動が考えられる。 ●データサイエンスの技術や考え方が，これからの社会にどのように活かすことができるか考察する学習を行う。

全体を通じた学習活動を行う上での注意点

授業時の環境に応じて，利用できるWebサービス，ソフトウェアやプログラミング言語を適切に選択し，体験的に学習できるよう留意する。このとき，必要に応じてクラウドサービスを利用したり，プログラミング言語で実行するのに必要なライブラリやパッケージをインストールしたりして，実際にデータを収集，整理・整形，処理することを体験的に学習できるよう留意する。

クラウドサービスの利用やライブラリやパッケージのインストールが環境的に難しい場合は，それぞれのデータを扱うための考え方を理解し，どのような処理を行えばよいかを考えられるようにする。

データを単に収集，整理・整形，処理するだけでなく，問題解決への活用やデータサイエンスが身近な生活や社会に及ぼす影響や役割についても考えられるよう留意する。

MEMO