数理生物学演習

第14回　画像解析・機械学習・ニューラルネット

村田英和
murata@morphometrics.jp
九州大学大学院システム生命科学府
数理生物学研究室

第１３回：機械学習による画像処理

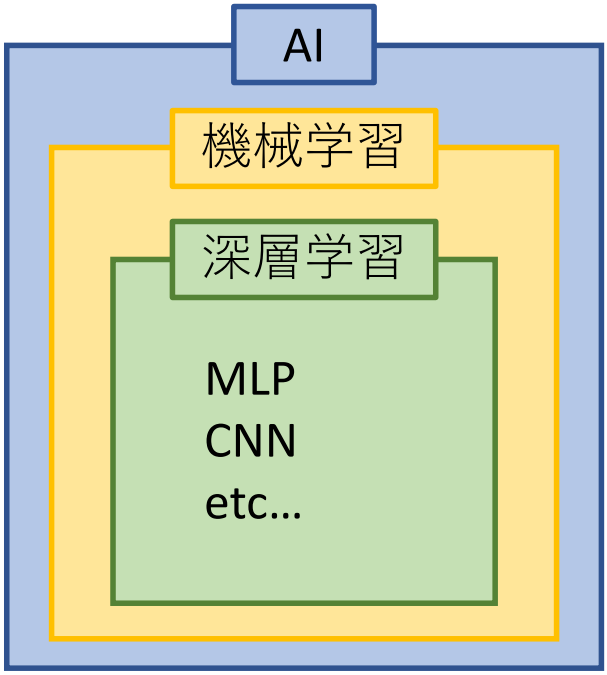
本日の目標

- ニューラルネットワーク
- 深層学習
- 画像解析

やらないこと

- NN,　DL以外の機械学習
- 学習
- CNN, RNN, Transformer
- ライブラリの使い方

AI，機械学習，深層学習？



- 人工知能（AI：artificial intelligence）は、「計算」という概念と「コンピュータ」という道具を用いて「知能」を研究する計算機科学の一分野（佐藤 2018）
- 機械学習は、経験を通じて自動的に改善されるコンピューターアルゴリズムの研究（T. Mitchell, M. Hill,1997）

今回は主に深層学習について扱います

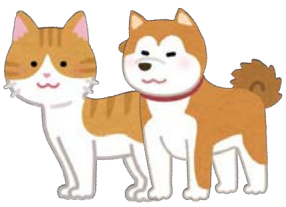
機械学習の種類

- 教師あり学習（Supervised learning）
  - 学習に教師データ（答え）が必要
  - 深層学習は主にこっち


イヌ

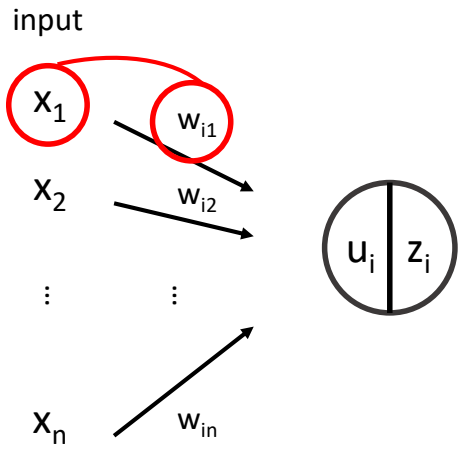

ネコ

- 教師無し学習（Unsupervised (Self-supervised) Learning）
  - 学習に教師データ（答え）が要らない
  - クラスタリングや次元圧縮など
  - 深層学習でこれを扱うものもある



ニューラルネットワーク

ユニットの構造



xからuの計算

$$u_i = \sum_{j=1}^{n} w_{ij} x_j$$

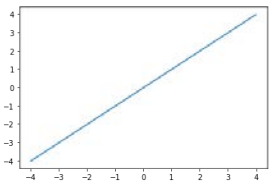
uからzの計算

$$z_i = f(u_i)$$

$f(\cdot)$：活性化関数 activation function

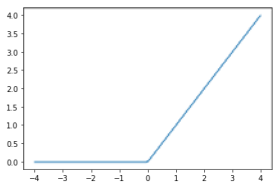
恒等写像

$$f(u) = u$$



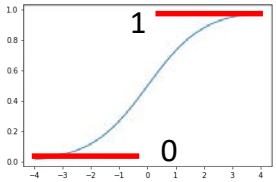
ReLU

$$f(u) = \max(0, u)$$



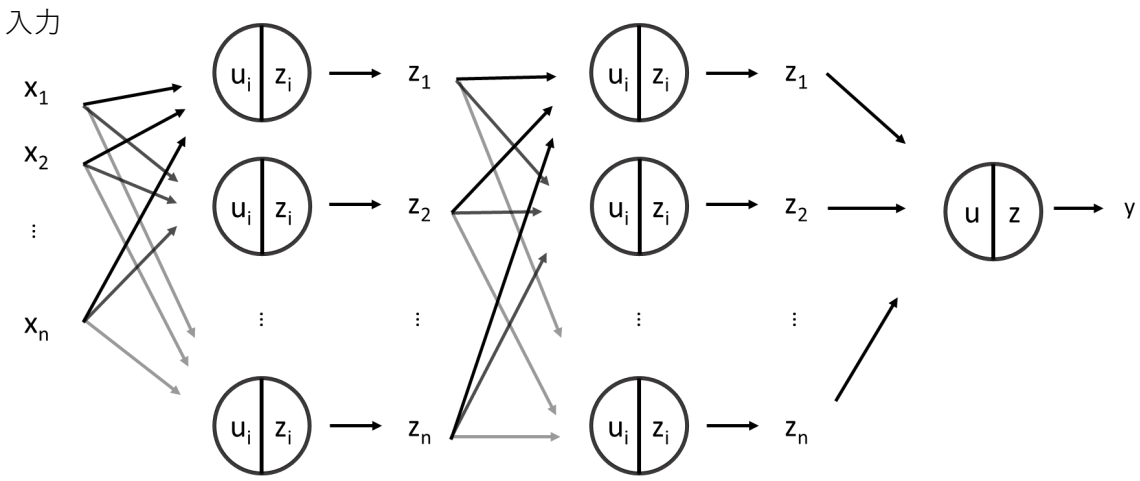
シグモイド

$$f(u) = \frac{1}{1 + e^{-u}}$$

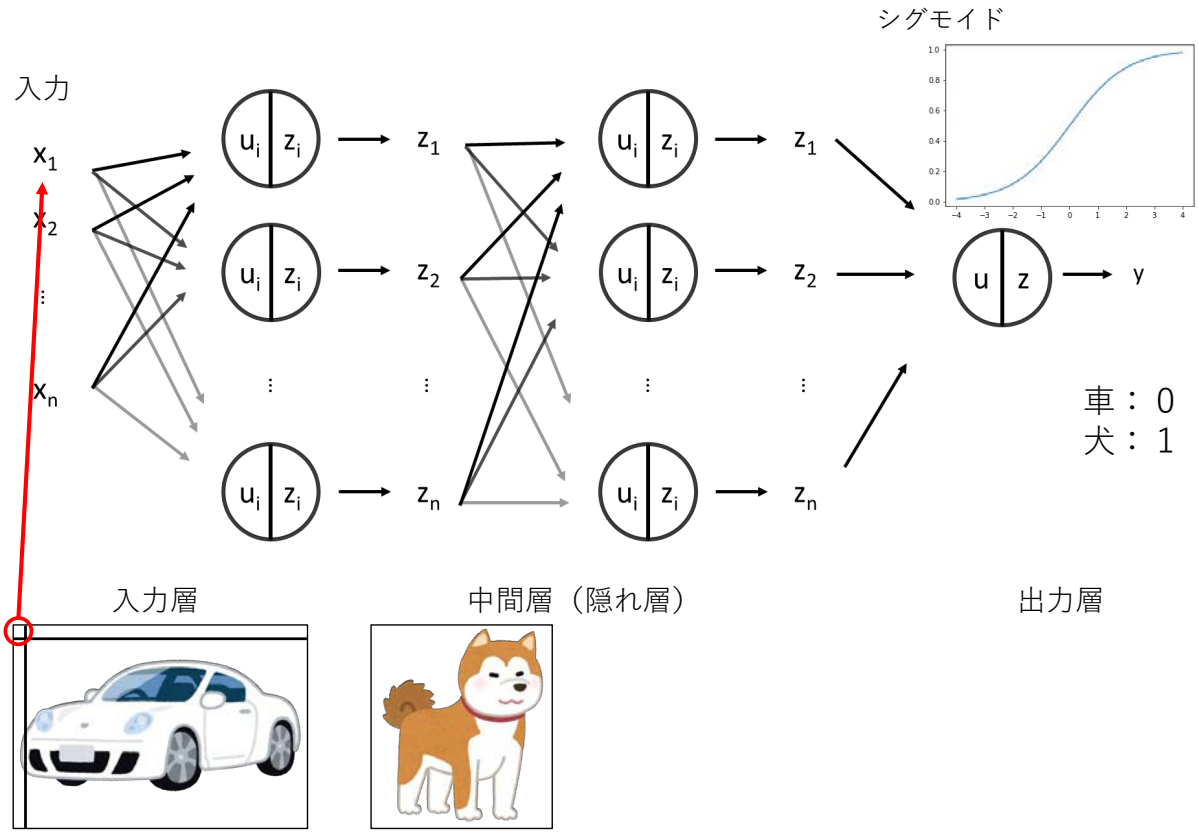


状況により使い分ける，他にもある

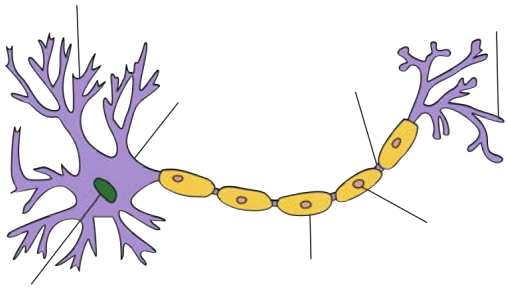
多層パーセプトロン（MLP）



多層パーセプトロン（MLP）



Ex. 人工ニューラルネットワーク？？



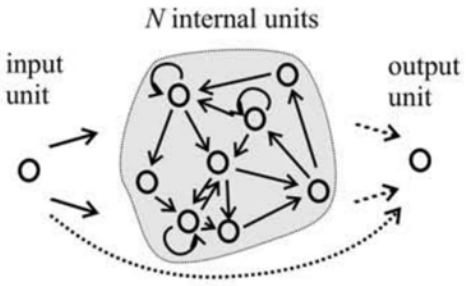
https://ja.wikipedia.org/wiki/%E7%A5%9E%E7%B5%8C%E7%B4%B0%E8%83%9E#/media/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB:Neuron_Hand-tuned.svg

神経細胞の様子を数理モデルで
表現したいわけではない

計算論的神経科学
(Computational Neuroscience)

ex.) FitzHugh-Nagumoモデル
Hodgkin-Huxleyモデル

計算論的神経科学から
情報科学に応用された例



(Jaeger, H. 2003)

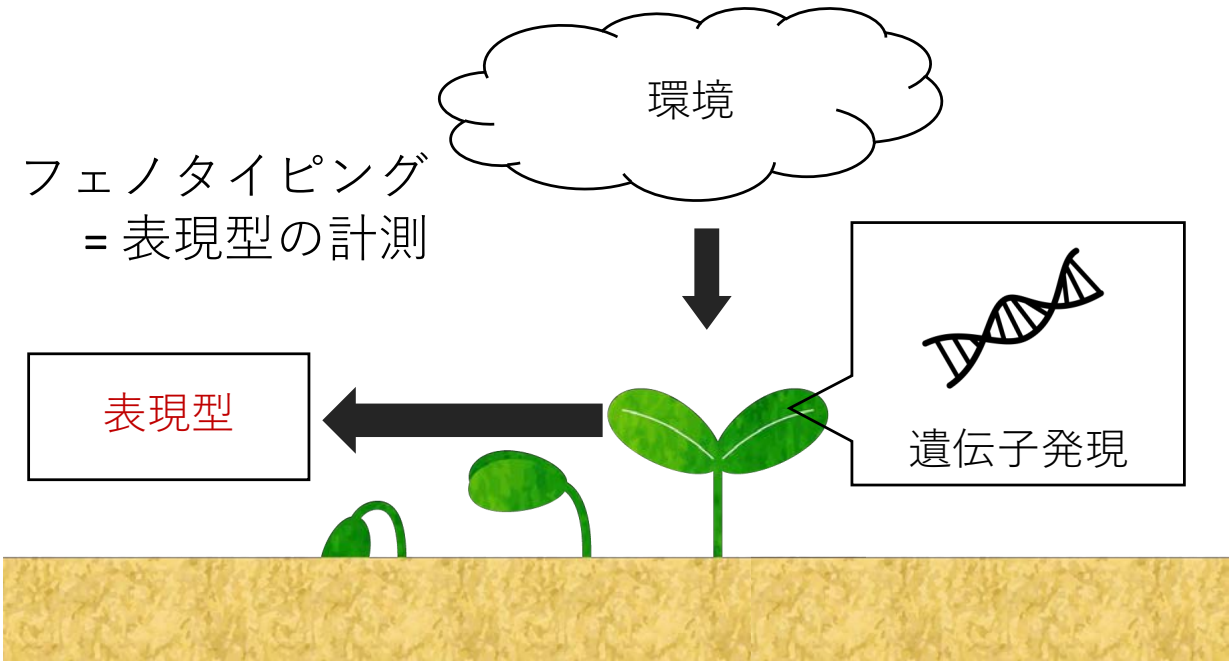
Reservoir Computing

生物学における画像解析の例

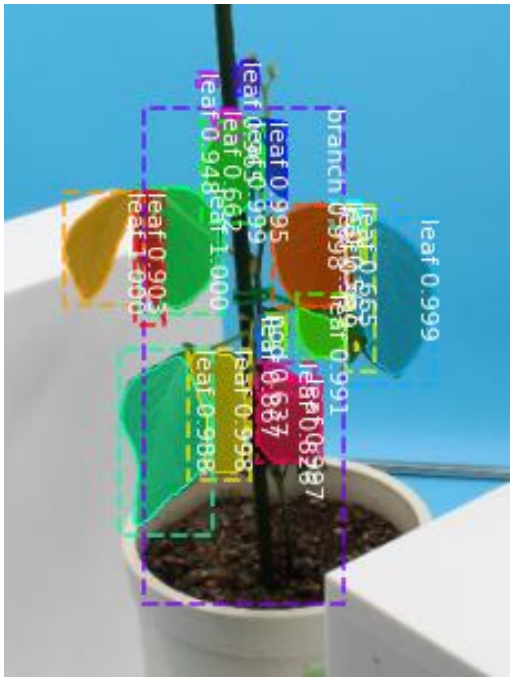
- Tracking
  - DeepLabCut (http://www.mackenziemathislab.org/deeplabcut)

- 生き物の分類（昨年の資料も参照）
  - iNaturalist (https://www.inaturalist.org/)
  - Biome (https://biome.co.jp/app-biome/)

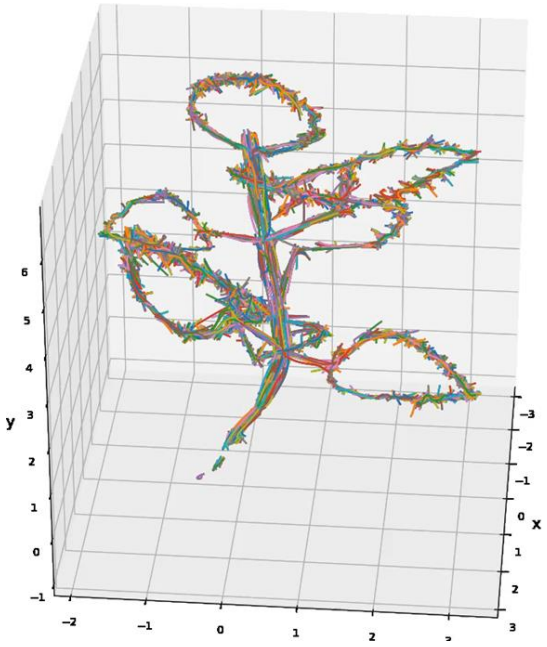
背景



私の研究

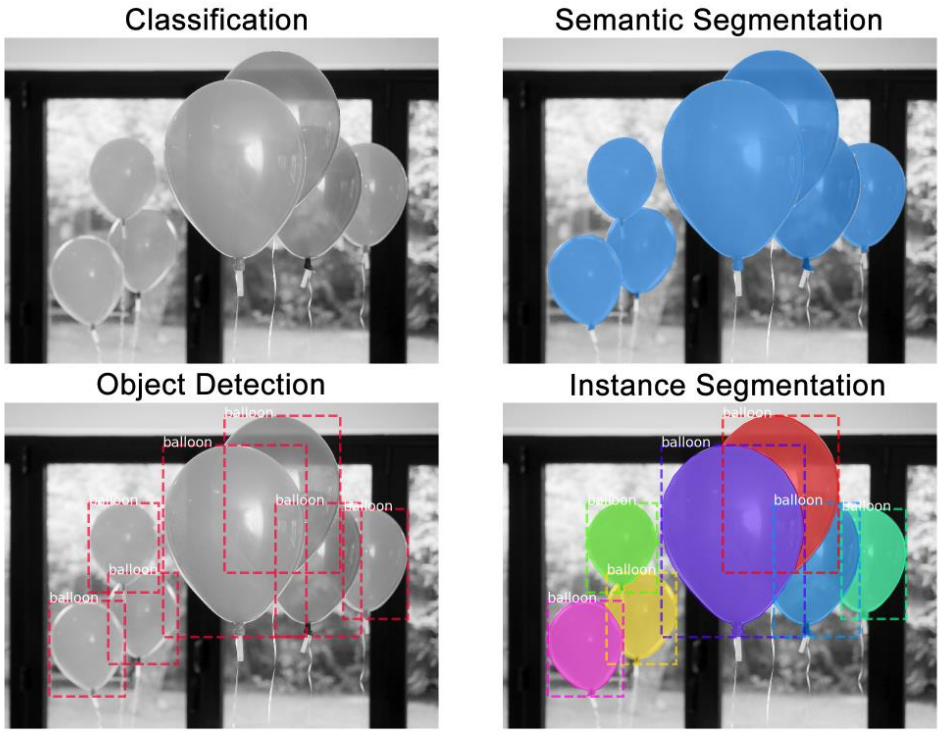


葉ごとのインスタンスセグメンテーション



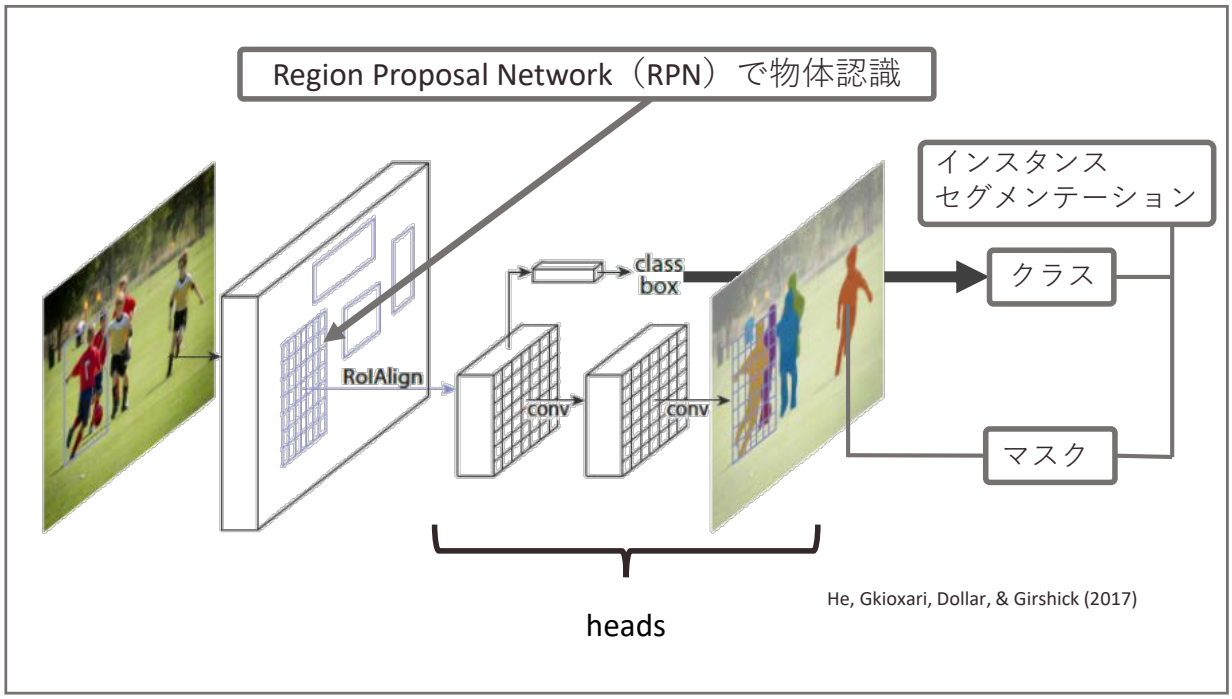
2D輪郭からの再構築

領域分割 (Segmentation)



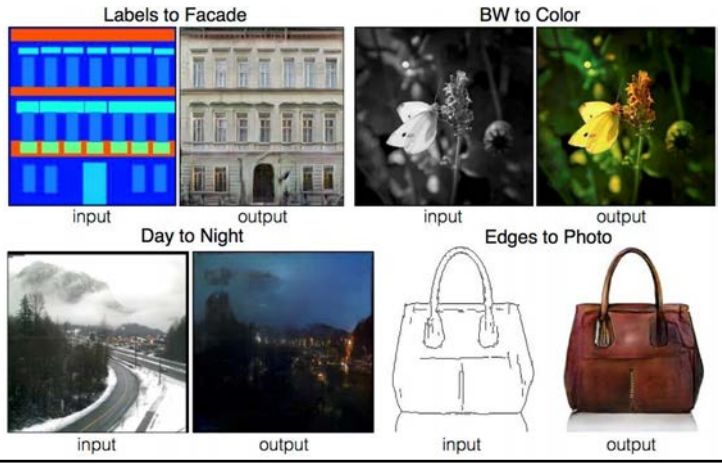
https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46

Mask R-CNN（He, Gkioxari, Dollar & Girshick 2017）



Ex. 生成モデル

Generative adversarial networks (GAN)



Phillip *et al.* 2016

Wasserstein GAN

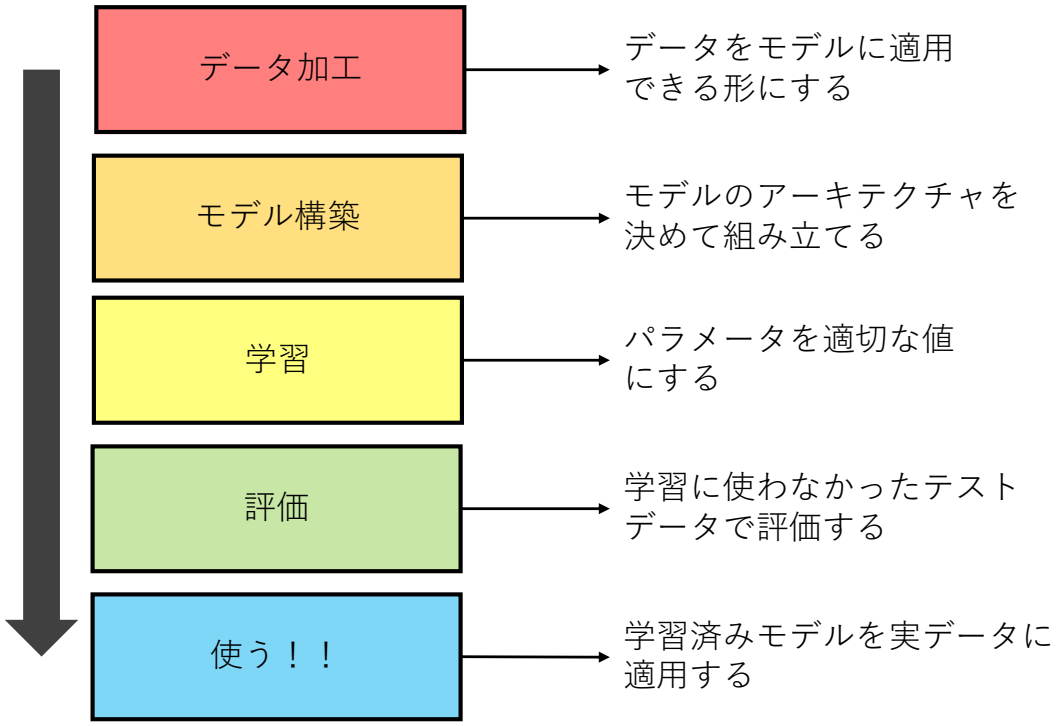
Wasserstein距離を学習に使用する.
Wasserstein距離：Earth Mover's Distance(EDM)の一般化. 分布間の距離を測る
確率分布を扱う ⇒ ベイズの知識が必要

実際にやってみよう！

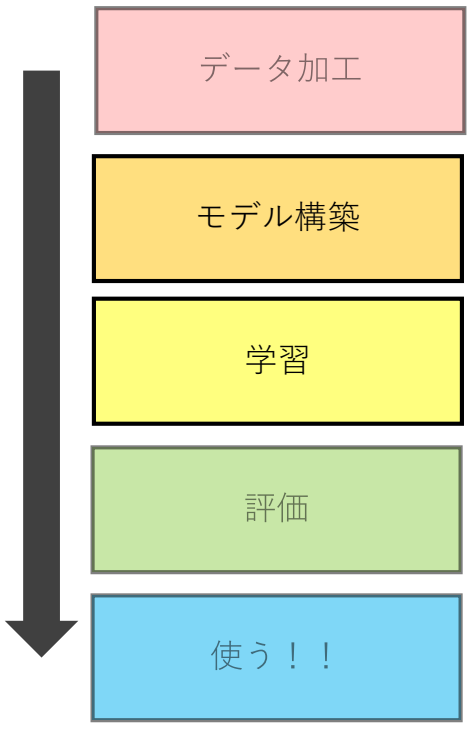
Deep Learningの準備

- 課題設定（何がしたい？）
  - ex.) 画像の分類，コロニーやセルのカウント，トラッキング
- データ収集
- アノテーション（正解ラベルを付ける）

Deep Learningの流れ



Deep Learningの流れ

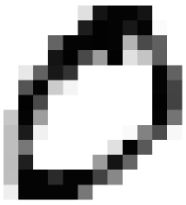


ライブラリの種類

Keras

MNISTデータセット

http://yann.lecun.com/exdb/mnist/ Lecun et al. (1998)



画像

ラベル：0

- 手書きの数字に対してラベルが降られている
- 28×28 pixel
- 訓練データ：60000枚
- テストデータ：10000枚
- 0〜9の10クラス

問題設定：手書きの数字を認識する

このデータを用いて多クラス分類を行う

ノートブックを開く

1. GitHubのタブを選択

2. 対象のGitHubレポジトリのURLを入力
https://github.com/murata721/Compbio2021

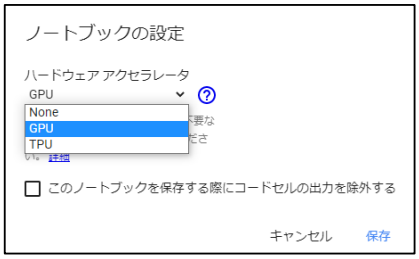
3. 開きたいノートブックを選択

4. ドライブにコピーでColab Notebooks内に保存される

ランタイムのタイプ

1. ランタイムのタイプを変更

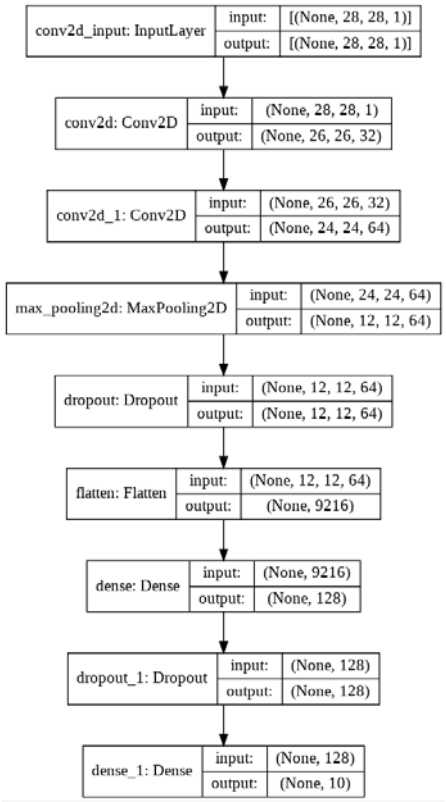
2. GPUを選択



データの加工

- ダウンロード
- train, validation, testに分割する
  - train：学習用
  - validation：ハイパーパラメータ調節用
  - test：評価用
- データのシャッフル
- shapeの変更
- 画像の標準化
- ラベルのone-hot vector化

モデル構築



- Kerasを用いることでレゴブロックを組み立てるように構築できる

  https://qiita.com/fukuit/items/b3fa460577a0ea139c88

学習・評価・使用

学習

```
from tensorflow.keras.optimizers import RMSprop

model.compile(loss='categorical_crossentropy',
              optimizer=RMSprop(),
              metrics=['accuracy'])

history = model.fit(x_train, y_train,
                    batch_size=128,
                    epochs=10,
                    verbose=1,
                    validation_data=(x_valid, y_valid))
```

評価

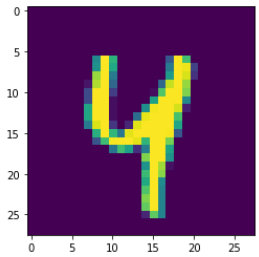
```
score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
Test loss: 0.0385587140917778
Test accuracy: 0.9882000088691711
```

使用

```
# テストデータを取り出して適用してみる
plt.imshow(x_t)

# trainの一番最初のラベル
print("ラベル：",y_t)
```

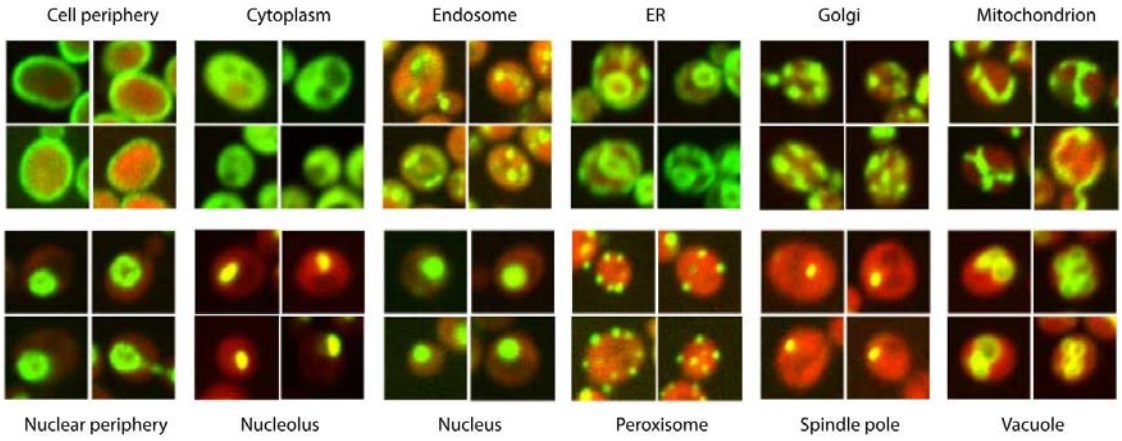
```
ラベル： 4
```



```
print("推測したラベル：" ,np.argmax(model.predict(x_test[42].reshape(1,28
                                  batch_size=None, verbose=0, steps=None
```

```
推測したラベル： 4
```

Yeast GFP protein localization classification



Reference
Accurate Classification of Protein Subcellular Localization from High-Throughput Microscopy Images Using Deep Learning Tanel Pärnamaa and Leopold Parts G3: GENES, GENOMES, GENETICS May 1, 2017 vol. 7 no. 5 1385-1392; https://doi.org/10.1534/g3.116.033654

問題設定：画像を局在部位ごとに分けたい

https://github.com/totti0223/deep_learning_for_biologists_with_keras

本日の課題

Easy

A

酵母菌のタンパク質局在の画像をいくつか取得した．これを自動で分類したい．

授業で扱った学習済みモデルを用いるために何をすればよいか説明しなさい．

or

B

生物学の研究にDeep Learningを用いた研究を1つ挙げ，そこでの課題設定と解決のためにどのようなデータを用いたかを説明しなさい．

（参考文献を記載すること）

and

C

質問，意見，要望等をどうぞ．

本日の課題

Easy

A

酵母菌のタンパク質局在の画像をいくつか取得した．これを自動で分類したい．

授業で扱った学習済みモデルを用いるために何をすればよいか説明しなさい．

or

B

生物学の研究にDeep Learningを用いた研究を1つ挙げ，そこでの課題設定と解決のためにどのようなデータを用いたかを説明しなさい．

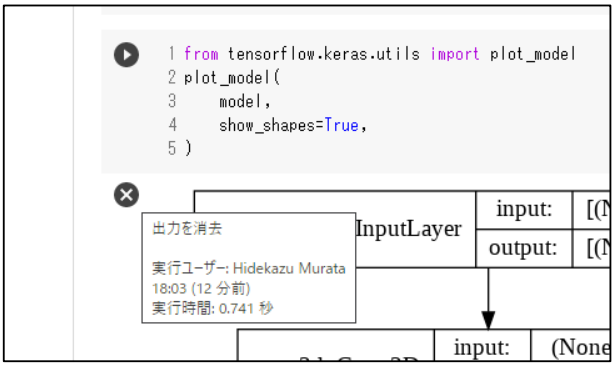
（参考文献を記載すること）

本日の課題

Hard

自分で書いた数字（デジタル可）を学習したMNISTの分類モデルにかけ分類できることを確認しなさい．これには演習で用いたノートブックを使用してよい．

ただし，使用した画像を表示すること



Tips：
提出するノートブックの容量が大きい場合，不要な出力を消去することで軽くなります

次回予告

第15回：数理生物学でのプログラミング
7月26日

Hard課題補足

例では画像処理にPillowを用います.
（画像処理ライブラリは他にOpenCV等がある）

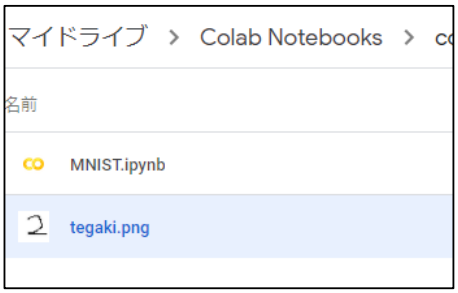
```
from PIL import Image
import numpy as np

img_path = “自分で作った画像のパス"
img = Image.open(img_path) # 画像読み込み
gray_img = img.convert(‘L’)    # グレースケールに変換
np_gray_img = np.array(gray_img) # ndarrayにキャスト

# np_grayとMNISTのデータを比べてモデルに適用できるように変換してください

print(“推測したラベル：”,
            np.argmax(model.predict(“自分で加工したデータ”,
                        batch_size=None, verbose=0, steps=None)))
```

Hard課題補足

画像の取り込み方

1. 適当な画像を作る
（この時点で28×28 pixelならリサイズしなくてよい，pngかjpegがよい）

2. Googleドライブに作った画像をアップロード（ノートブックと同じ階層を推奨）

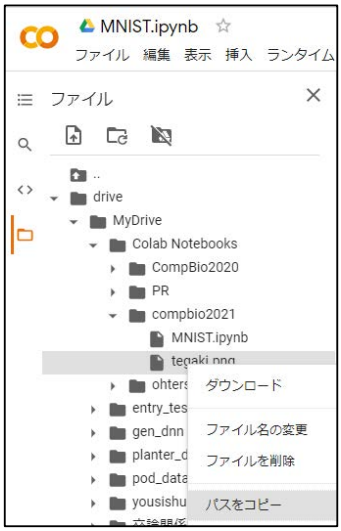


Hard課題補足

画像の取り込み方

3. Colabの左側のメニューからドライブをマウントする

4. drive/My Drive　以下の
自分が保存したフォルダから画像を
右クリック→パスをコピーを選択



Hard課題補足

画像の取り込み方

5. Ctrl + v で任意の場所に貼り付け