

# 数理生物学演習

第13回 ニューラルネットワーク

野下 浩司 (Noshita, Koji)

✉ noshita@morphometrics.jp

🏠 <https://koji.noshita.net>

理学研究院 数理生物学研究室

## 第13回：ニューラルネットワーク

### 本日の目標

- パーセプトロン
- ニューラルネットワーク
- 画像認識

# 生物学における画像解析の例

画像認識：画像にラベルをつける

物体認識：カウントする

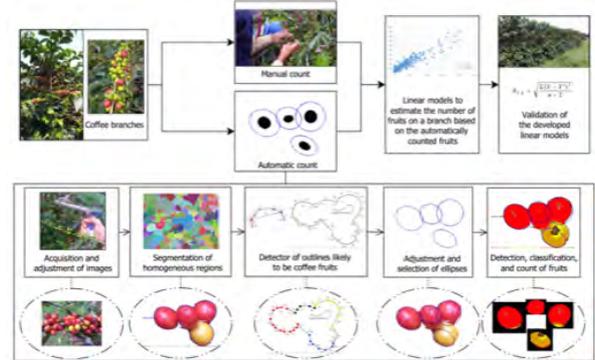
生物の分類



iNaturalist

<https://www.inaturalist.org/>

コーヒーの実

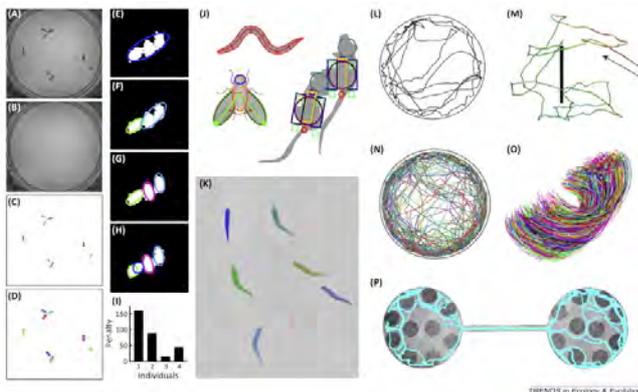
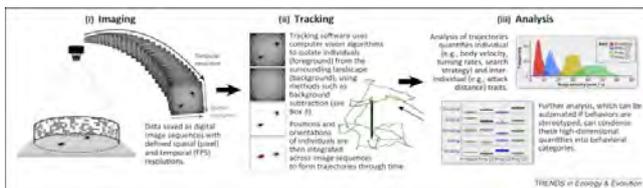


Ramos et al. (2017)

基本的なタスク（画像認識，物体認識）も面白い応用先を考えると生物学的に意味のある情報を取る重要なツールになりうる

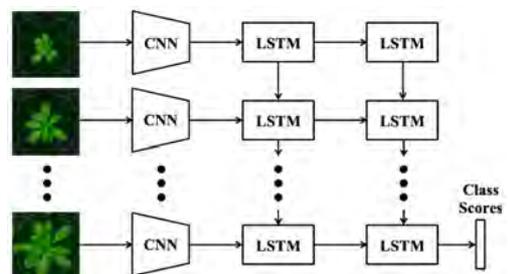
# 生物学における画像解析の例

物体認識+時系列データ：トラッキング



Dell et al. 2015

画像認識+時系列データ：時系列画像からの系統（accession）の分類  
Namin et al. (2018)



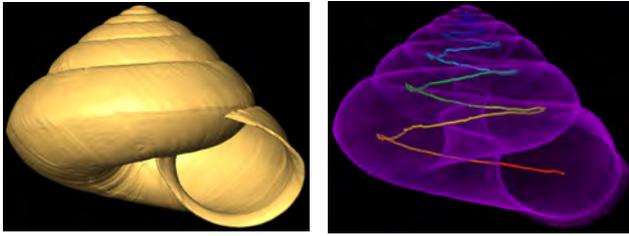
大量のデータを自動で処理する



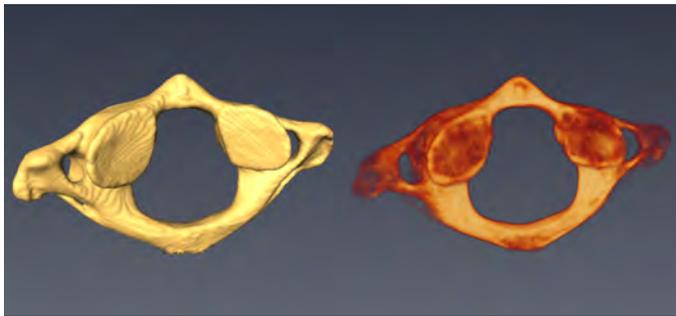
成長パタンの推定などにも応用可能

# データの種類

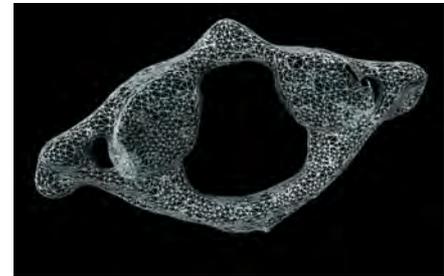
ボリューム



点群

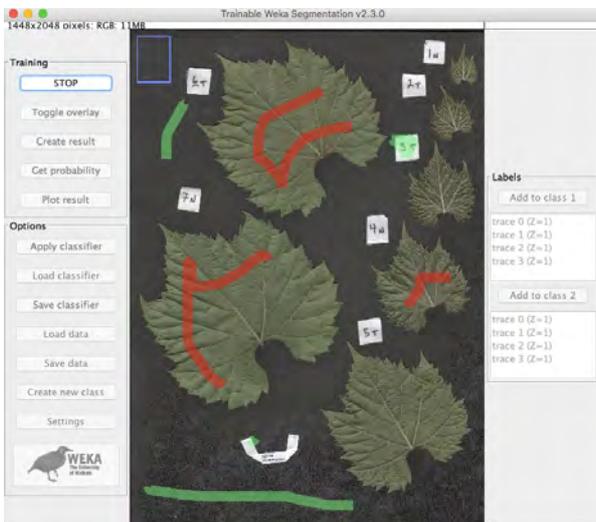


メッシュ



## 生物学における画像解析の例

輪郭抽出

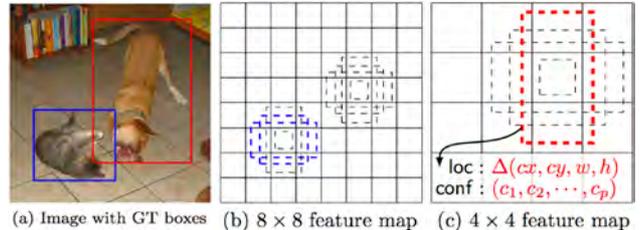


# 深層学習を利用した画像解析

## 物体認識



## 物体検出



## 領域分割

(semantic segmentation)



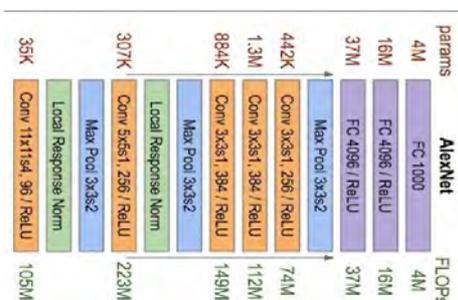
## 画像生成



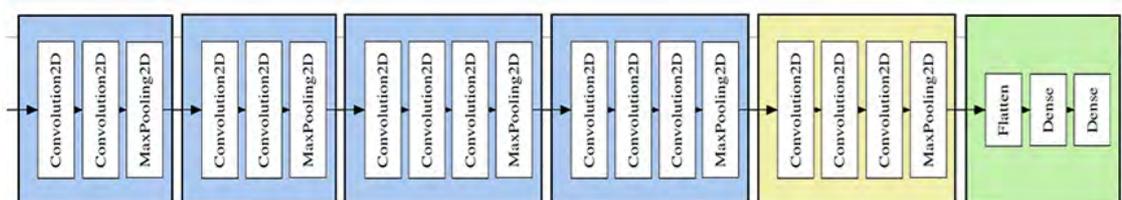
# 深層学習 deep learning ?

- 多層のニューラルネットワークを用いた機械学習手法
- 畳み込みニューラルネットのブレイク

AlexNet  
Krizhevskyyetal. 2017



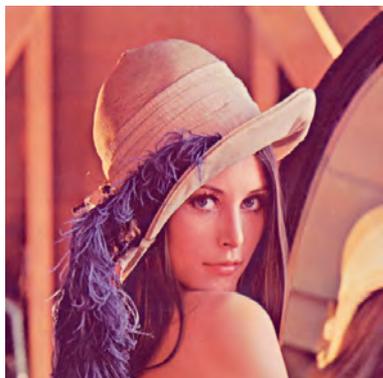
VGG16



# 物体認識

写っているものを認識

(どこに写っているかは不問)



特徴名:	値
説明	<pre>{ "tags": [ "person", "woman", "clothing", "hat", "hair", "girl", "sitting", "looking", "young", "holding", "wearing", "table", "standing", "black", "head", "white", "cake", "phone", "red", "umbrella", "people", "man" ], "captions": [ { "text": "a woman wearing a hat", "confidence": 0.9403839 } ] }</pre>
タグ	<pre>[ { "name": "person", "confidence": 0.9961416 }, { "name": "woman", "confidence": 0.9500116 }, { "name": "hair", "confidence": 0.4392622 } ]</pre>

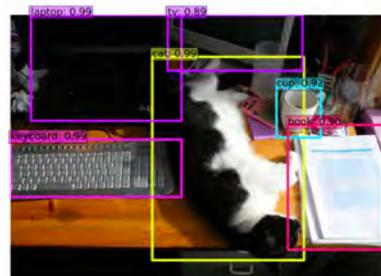
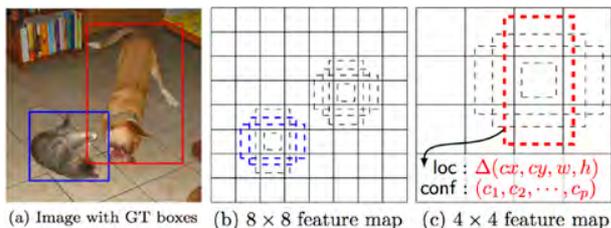
- Google Cloud Vision API
- Microsoft Computer Vision API <https://azure.microsoft.com/ja-jp/services/cognitive-services/computer-vision/>

# 物体検出

どこに何があるかを答える

(ある程度の範囲を指定できればOK)

SSD (Single Shot Multibox Detector)



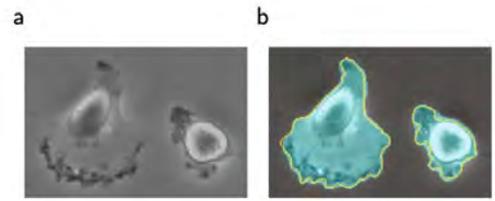
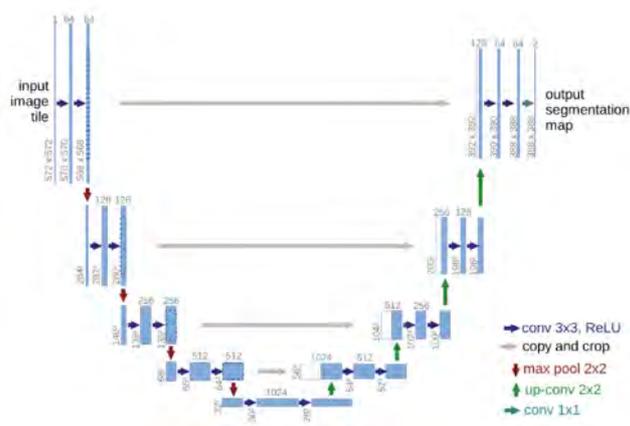
Liu et al. (2016)



# 領域分割 Semantic Segmentation

## ピクセルレベルでどの物体か認識

U-Net Ronneberger et al. (2015)



# 領域分割 Instance Segmentation

Mask R-CNN

He et al. (2018)



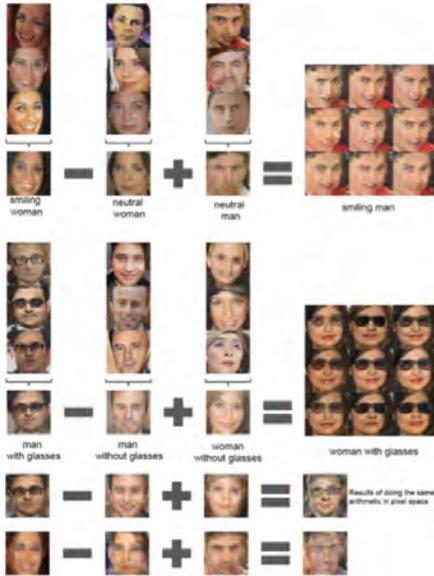
Figure 2. Mask R-CNN results on the COCO test set. These results are based on ResNet-101 [19], achieving a mask AP of 35.7 and running at 5 fps. Masks are shown in color, and bounding box, category, and confidences are also shown.

- 矩形検出, セグメンテーション, 姿勢推定を同時に可能



# 画像生成

DCGAN Radford et al. (2016)



## 仮想的ベッドルーム



## 仮想的CDジャケット



実際にやってみよう！

# ノートブックを開く

## GitHubから開く



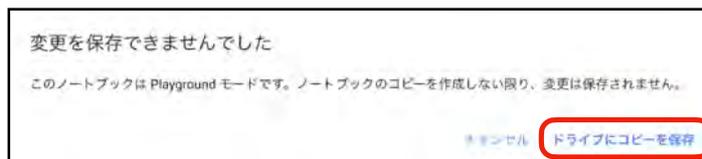
1. GitHubのタブを選択

2. 対象となるGitHubレポジトリのURLを入力

<https://github.com/noshita/Compbio2019-notebook>

3. 開きたいノートブックを選択

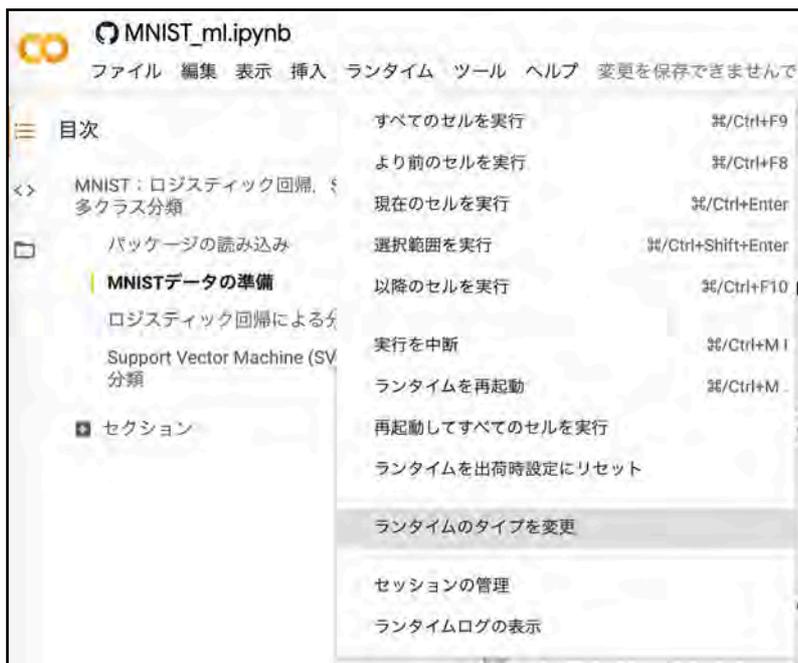
## Google Driveへ保存



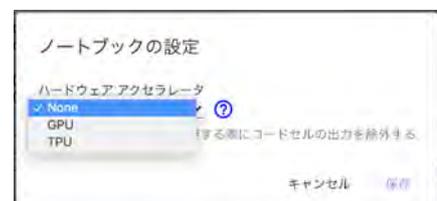
自分のGoogle Driveへも保存可能

# ランタイムのタイプを変更

## 1. ランタイムのタイプを変更



## 2. 今回はGPUを選択



画像認識：ある画像に何が写っているか自動で検出したい

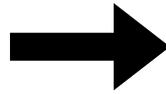
生物学的な課題の例：生物種の分類，細胞の種類認識，…



この演習では問題を以下のように読み替えて取り組んでみよう

問題設定：手書き数字をコンピュータに認識させたい

5  
0  
4  
1  
9  
2  
1  
3  
1  
4



5  
0  
4  
1  
9  
2  
1  
3  
1  
4

MNIST (Modified National Institute of Standards and Technology)

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

- 28x28ピクセルの手書き数字画像のデータセット
  - 訓練データ：60000枚
  - テストデータ：10000枚
- 0-9までの10クラス
- 画像処理系のベンチマークなどによく用いられる
  - Kerasのデータセットにも含まれている

5	0	4	1	9	2	1	3	1	4
3	5	3	6	1	7	2	8	6	9
4	0	9	1	1	2	4	3	2	7
3	8	6	9	0	5	6	0	7	6
1	8	7	9	3	9	8	5	9	3
3	0	7	4	9	8	0	9	4	1
4	4	6	0	4	5	6	1	0	0
1	7	1	6	3	0	2	1	1	7
8	0	2	6	7	8	3	9	0	4
6	7	4	6	8	0	7	8	3	1



# ロジスティック回帰, SVM

ロジスティック回帰による分類：正答率 ~ 90%

SVMによる分類：正答率 ~ 94%

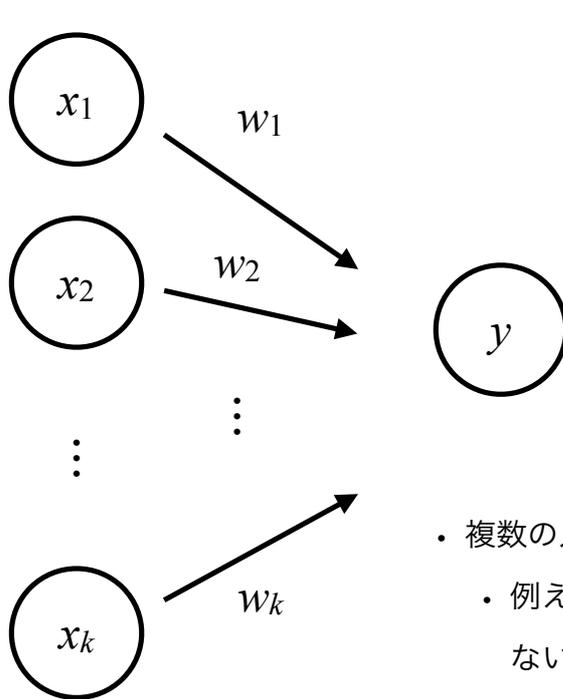
実際にやってみよう！

サンプルノートブック 01\_MNIST\_ml.ipynb

# パーセプトロン perceptron

Rosenblatt (1958)

## 単純パーセプトロン



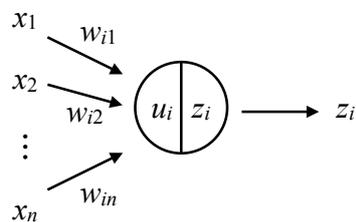
$$y = \text{sgn} \left( \sum_{i=1}^k w_i x_i \right)$$

$x_i$ : i番目の入力  
 $w_i$ : i番目の重み  
 $y$ : 出力

- 複数の入力を受けて±1を返す（二値分類に利用できる）
- 例えば、ある画像に動物Aが写っているか (+1) いないか (-1) , などが考えられる

# 順伝播型ニューラルネットワーク Feedforward Neural Network

## ユニットの構造

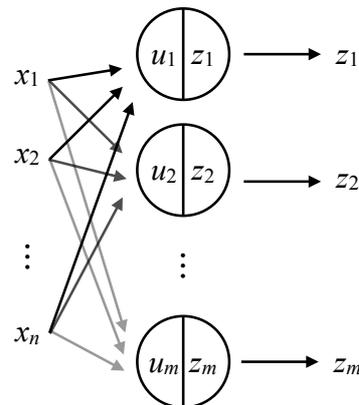


$$u_i = \sum_{j=1}^n w_{ij} x_j$$

活性化関数 activation function

$$z_i = f(u_i)$$

## 2層

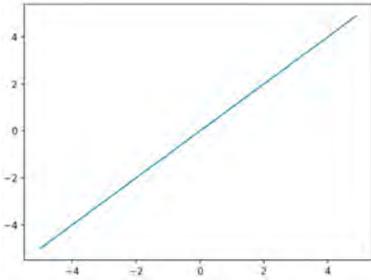


$$\mathbf{z} = \mathbf{f}(\mathbf{u})$$
$$\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{pmatrix}$$

# 活性化関数 activation function

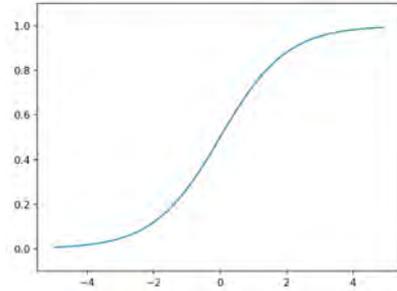
恒等写像

$$f(u) = u$$



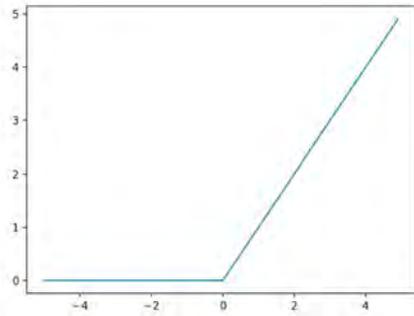
シグモイド関数

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



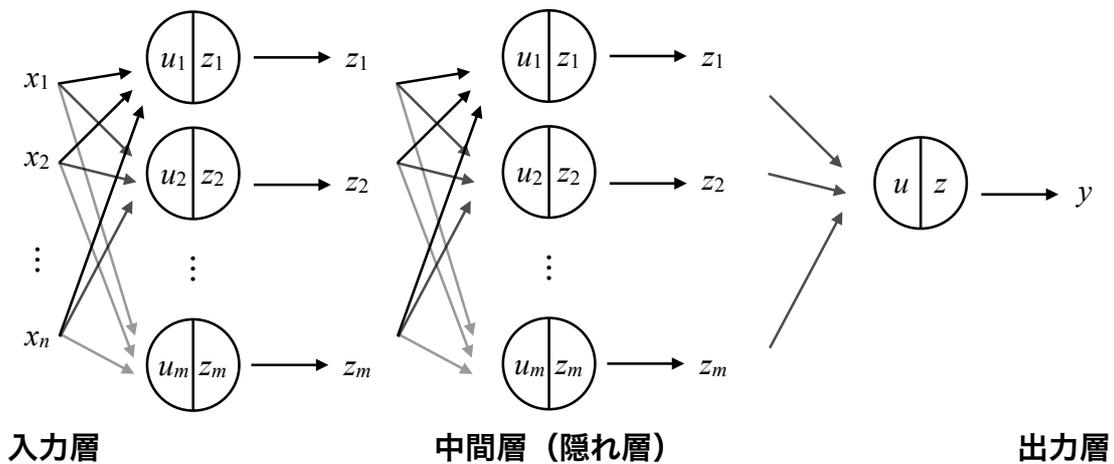
Rectified Linear Unit (ReLU)

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



様々な活性化関数があるが、  
目的に応じて使い分ける

## 多層ネットワーク



# 確率的勾配降下法

## stochastic gradient descent (SGD)

極小値を探索する

勾配降下法

$$\text{勾配 } \nabla E = \frac{\partial E}{\partial \mathbf{w}} = \begin{pmatrix} \frac{\partial E}{\partial w_1} \\ \frac{\partial E}{\partial w_2} \\ \vdots \\ \frac{\partial E}{\partial w_m} \end{pmatrix}$$

パラメータの更新

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \varepsilon \nabla E$$

$\varepsilon$  : 学習係数 (learning rate)

確率的勾配降下法

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \varepsilon \nabla E_n$$

1つあるいは少数のサンプルについて損失関数を最小化する

バッチサイズ Batch size : 一度の学習に使うデータサイズ

エポック epoch : 学習の際にデータ全体を一周すること

# ニュートン法 Newton's method

復習 第4回

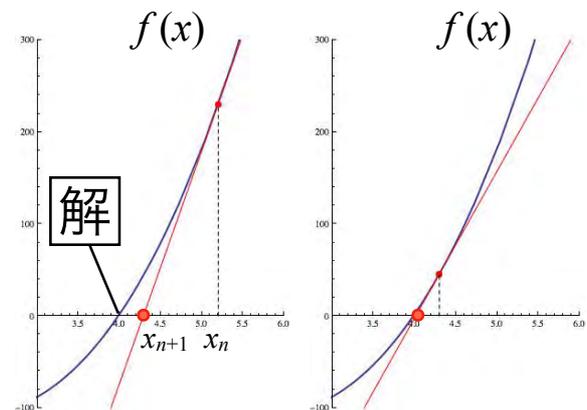
方程式を解くためのアルゴリズム

解を求めたい方程式を  $f(x)=0$  とすれば,

解は  $f(x)$  と  $x$  軸との交点の  $x$  座標になる。

では, どうやって「数值的に」求めるか?

1. 解の近似値を  $x_n$  とし, 適当なその初期値  $x_0$  を決める
  2. 解の近似値  $x_n$  での接線を求める
  3. この接線と  $x$  軸との交点を求める
  4. 交点の  $x$  座標を新たに近似値  $x_{n+1}$  として採用する
- 以後, 近似値が収束するまで  
2.~4. を繰り返す.



$x_n$  の漸化式を求めてみてください

# 誤差逆伝播法 back propagation

損失関数の微分を効率的に計算する

多層順伝播型ネットワーク

確率的勾配降下法で解くことを考える

勾配 
$$\frac{\partial E_n}{\partial w_{ij}^{(l)}} = \frac{\partial E_n}{\partial u_i^{(l)}} \frac{\partial u_i^{(l)}}{\partial w_{ij}^{(l)}} \dots \text{(A)}$$

右辺  
第一項 
$$\frac{\partial E_n}{\partial u_i^{(l)}} = \frac{\partial E_n}{\partial u_k^{(l+1)}} \frac{\partial u_k^{(l+1)}}{\partial u_i^{(l)}}$$

$$\delta_i^{(l)} = \frac{\partial E_n}{\partial u_i^{(l)}} \quad \text{とすると}$$

$$\delta_i^{(l)} = \sum_k \delta_k^{(l+1)} \left( w_{ki}^{(l+1)} f'(u_i^{(l)}) \right) \dots \text{(B)}$$

$l$ 層の $\delta_i$ は $l+1$ 層の $\delta_k$  ( $k=1,2, \dots$ ) から計算できる → 逆伝播

出力層を考えると

$$\delta_i^{(L)} = \frac{\partial E_n}{\partial u_i^{(L)}}$$

で、損失関数が定めれば計算できる

右辺  
第二項 
$$\frac{\partial u_i^{(l)}}{\partial w_{ij}^{(l)}} = \frac{\partial}{\partial w_{ij}^{(l)}} \left( w_{km}^{(l)} z_m^{(l-1)} \right) = z_j^{(l-1)}$$

1. 訓練サンプルを入力にして、各層 $l$ の $u$ と $z$ を順に計算
2. 出力層での $\delta$ を求める
3. 中間層での $\delta$ を式Bに従い、計算する (逆伝播)
4. 各層 $l$ のパラメータ $w$ に関する微分を式Aに従って計算する

実際にやってみよう！

サンプルノートブック 02\_MNIST\_nn.ipynb

# CNNは何がよいの？

順伝播型ニューラルネットワーク (fully-connected)

- 一次元のベクトル表現  
→ 二次元平面上の位置関係の情報は失われる

畳み込みニューラルネットワーク

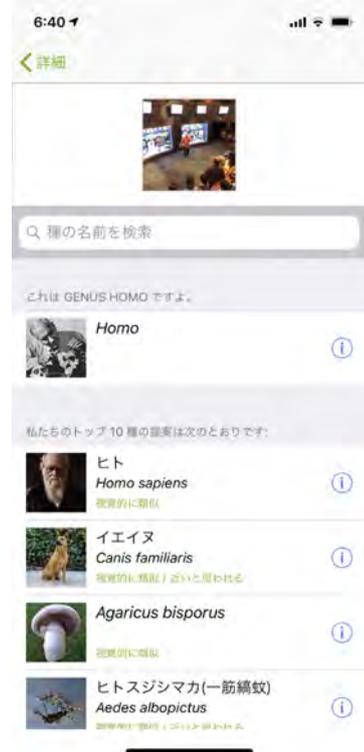
- 画像内の局所的な特徴を要約
- 同じ層の枚数を増やして多チャンネルも表現可能  
例：RGB画像→3チャンネル
- 特徴同士の（相対的な）位置関係を反映
- パラメータ数が（全結合に比べて）少ない

実際にやってみよう！

サンプルノートブック `03_MNIST_cnn.ipynb`

# 動物種の認識

iNaturalist



<https://www.inaturalist.org/>

# 生物学における画像解析の例

画像認識：画像にラベルをつける

物体認識：カウントする

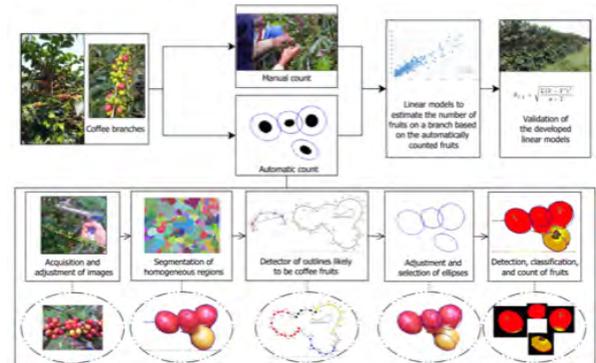
生物の分類



iNaturalist

<https://www.inaturalist.org/>

コーヒーの実

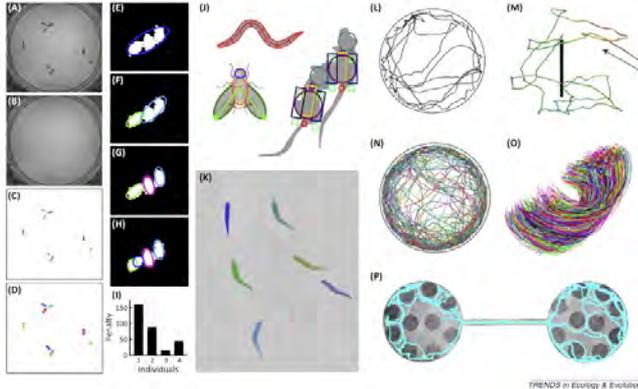
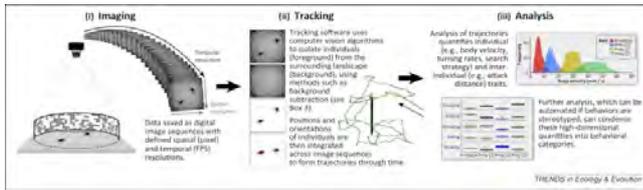


Ramos et al. (2017)

基本的なタスク（画像認識，物体認識）も面白い応用先を考えると生物学的に意味のある情報を取る重要なツールになりうる

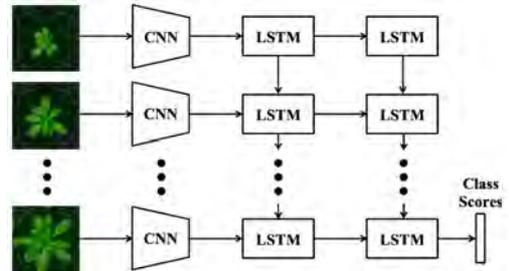
# 生物学における画像解析の例

## 物体認識+時系列データ：トラッキング



Dell et al. 2015

## 画像認識+時系列データ： 時系列画像からの系統 (accession) の分類 Namin et al. (2018)



大量のデータを自動で処理する



成長パタンの推定などにも応用可能

実際にやってみよう！

[https://github.com/totti0223/deep\\_learning\\_for\\_biotologists\\_with\\_keras](https://github.com/totti0223/deep_learning_for_biotologists_with_keras)

# 本日の課題

1. [質問] 前回の課題2（最終課題発表会についての質問・意見など）を受講者内で共有しても良いか？
2. 質問, 意見, 要望等をどうぞ.

基本的には特になし. 出席を兼ねるので2を提出はしてね.

Moodleのオンラインフォームから提出

## 次回予告

第14回：数理生物学でのプログラミング

8月10日（月）

## 復習推奨

- 特になし