

# そうだったのか！ 可視化して理解するディープラーニング

MathWorks Japan

アプリケーションエンジニアリング部

草野 駿一

これまで考えられなかったような画像処理が可能に！

- 画像のスタイル変換



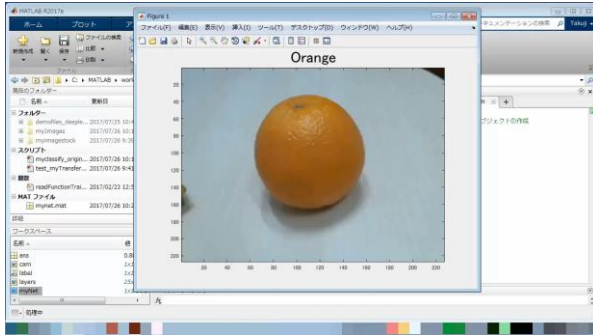
- Toonify



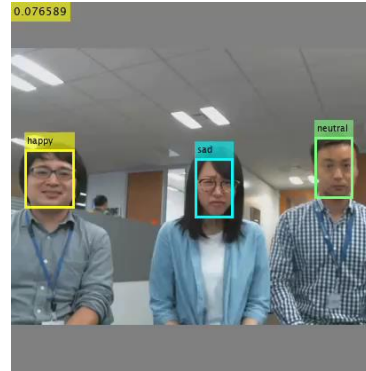


# ディープラーニングの画像への適用でできること

分類



顔検出



セマンティック  
セグメンテーション



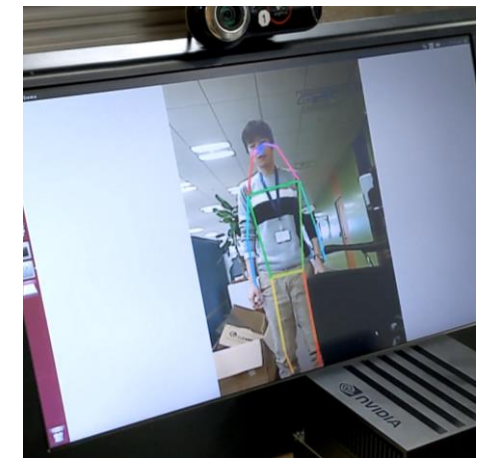
回帰 (数値推定)



画像生成



骨格検出



# 「AI」を全学部必修科目へ ディープラーニングを MATLAB で学習

## 金沢工業大学

金沢工業大学は MATLAB® による画像認識演習を含む「AI基礎」講座を開講します。この授業は全学部学科 1 年生1500名以上を対象に世界的に不足する AI 人材の育成を目指すものです。

MathWorks は、演習で使用する畳み込みニューラルネットワーク (CNN) による画像分類課題の作成に協力しました。授業中に学生が書いた文字をその場で学習データとして用い、学習データの拡張による精度向上の体験や、App Designer で作成した CNN 学習過程を可視化する GUI の利用により、AI を身近に感じながらより実践的な知識を習得できます。

金沢工業大学は今回の授業開発に先立ち、2015 年より Campus-Wide License を導入しています。これにより、全学生と教職員が MATLAB や Simulink® をいつでも、どこでも、自由に利用できるようになっています。

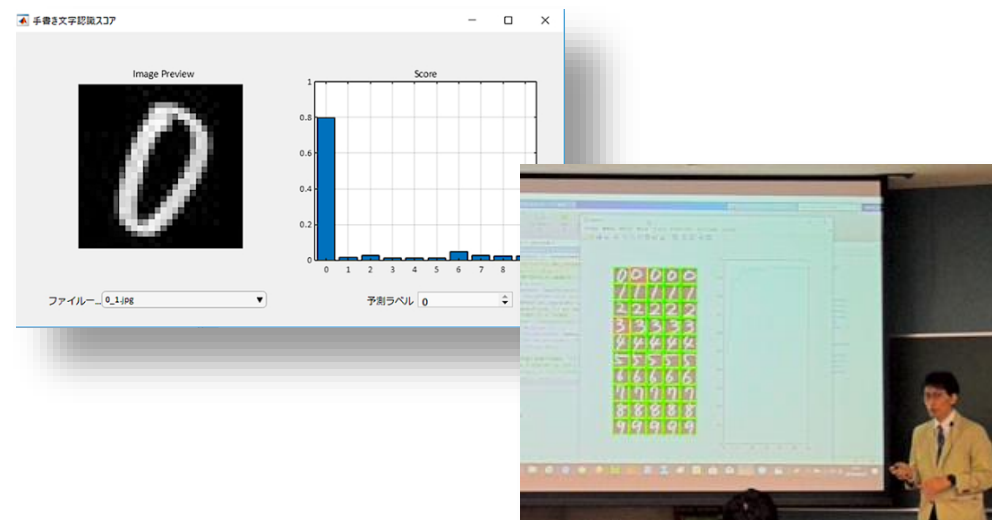
### MATLAB 利用のメリット:

- 毎年1500人規模の授業を可能にするバージョン管理された環境を構築
- 学習データ取り込みの自動化など授業のフロー全体を効率化
- AI に対する学生の直感的な理解を助ける GUI を構築可能
- Campus-Wide License とコード配布により自発的な応用学習が可能

“

MATLABの直感的なGUIと豊富なライブラリが魅力で「AI基礎」講座に採用しました。GUIを併用することで一年生でもAIの本質を理解でき、コードのカスタマイズも容易なため、学生の深い学びを実現することができます。

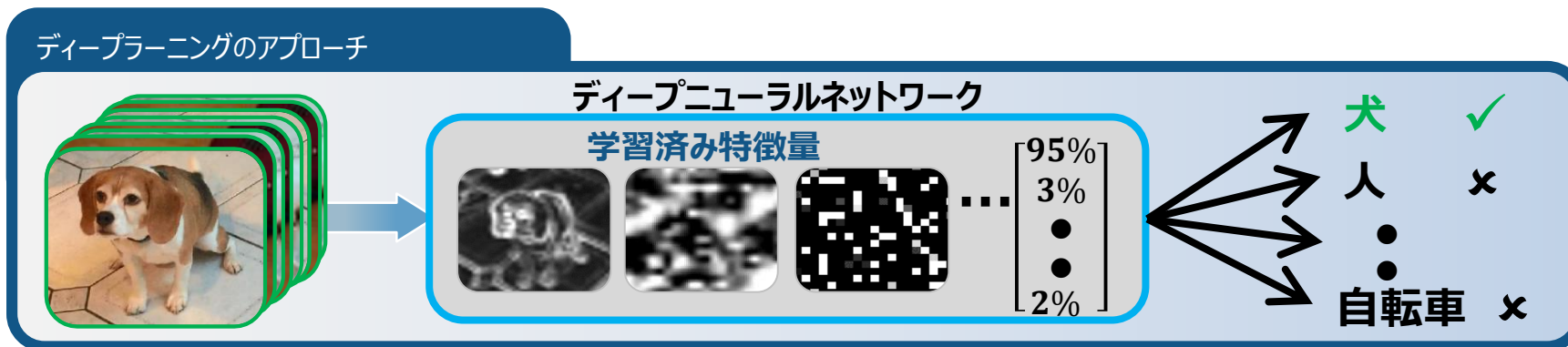
”



# ディープラーニング(深層学習)の特長



- 特徴量のデザインは人間が考え出す必要があった
- 様々な場合に対応できる特徴量を見つけ出すのが困難



特徴量のデザインすら、データから学習させることができる  
(ただし、パラメータチューニングの難しさや膨大な計算時間などの課題もある)

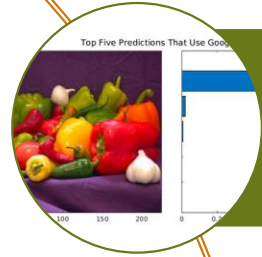
## 本セミナーで伝えたいこと

1. ディープラーニングの仕組み自体は、そんなに難しくない！
2. データを可視化して確認することが、理解への近道です！
3. 理解ができたなら、ツール（ライブラリ）を活用してください！

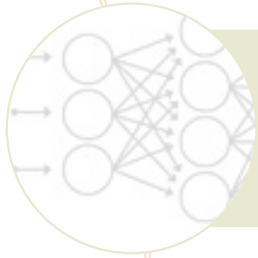
本日は下記の2点に絞ります

(画像の) ディープラーニングにおける特徴抽出の仕組み

学習の仕組み



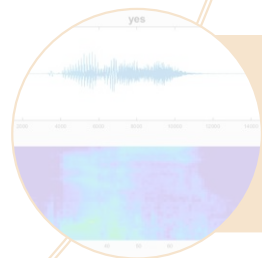
# 画像のディープラーニング



ニューラルネットワークとは？



実験を高速で繰り返すために



さらに学習したい人に



# 画像のディープラーニング

- 畳み込みニューラルネットワーク
- (**CNN: Convolutional Neural Network**)

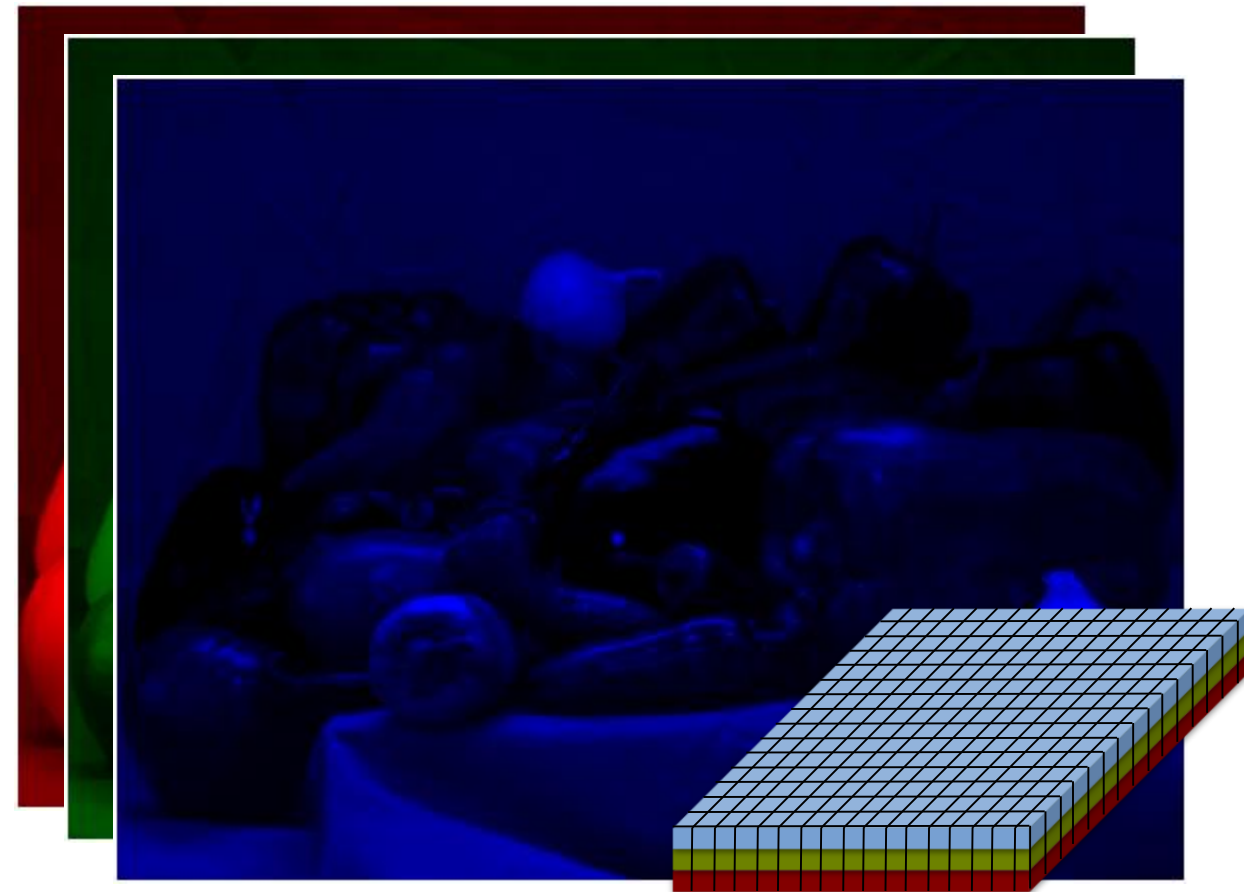
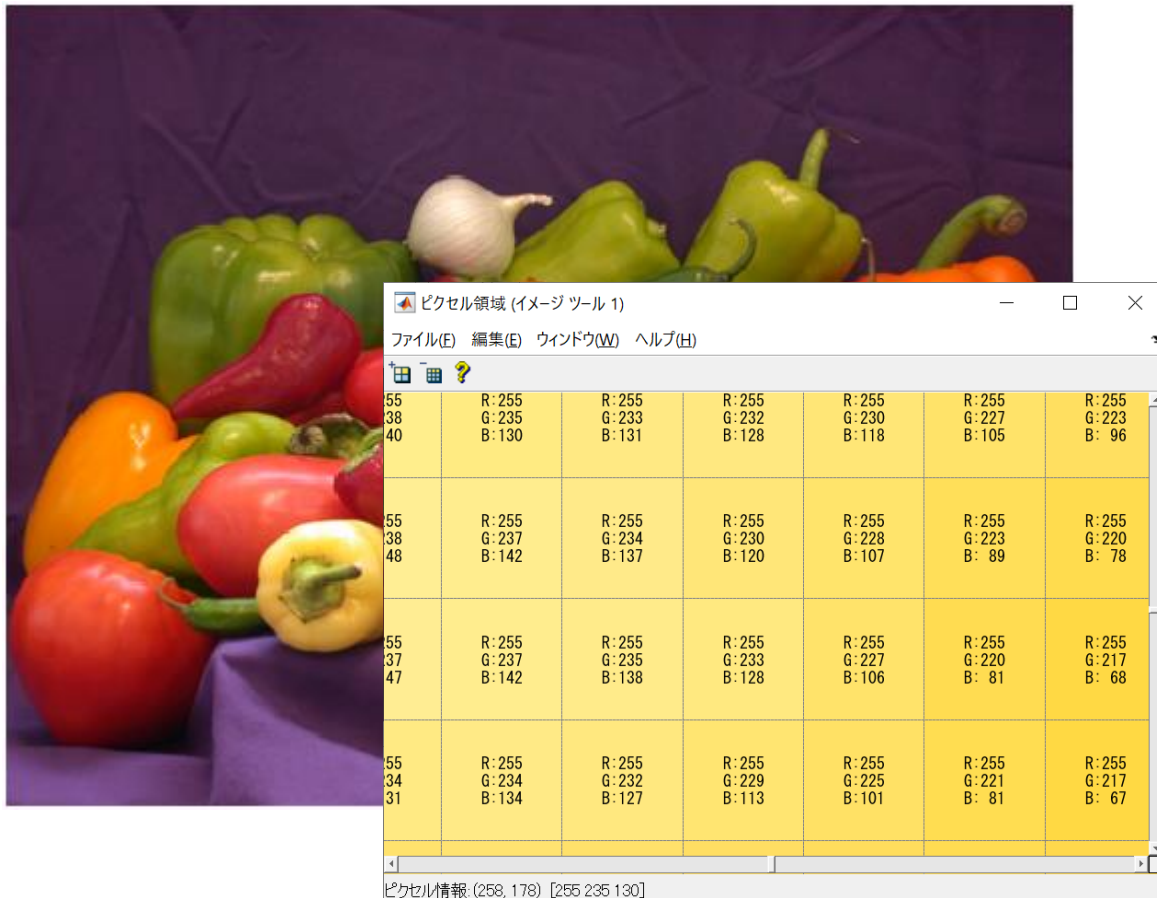


畳み込み？

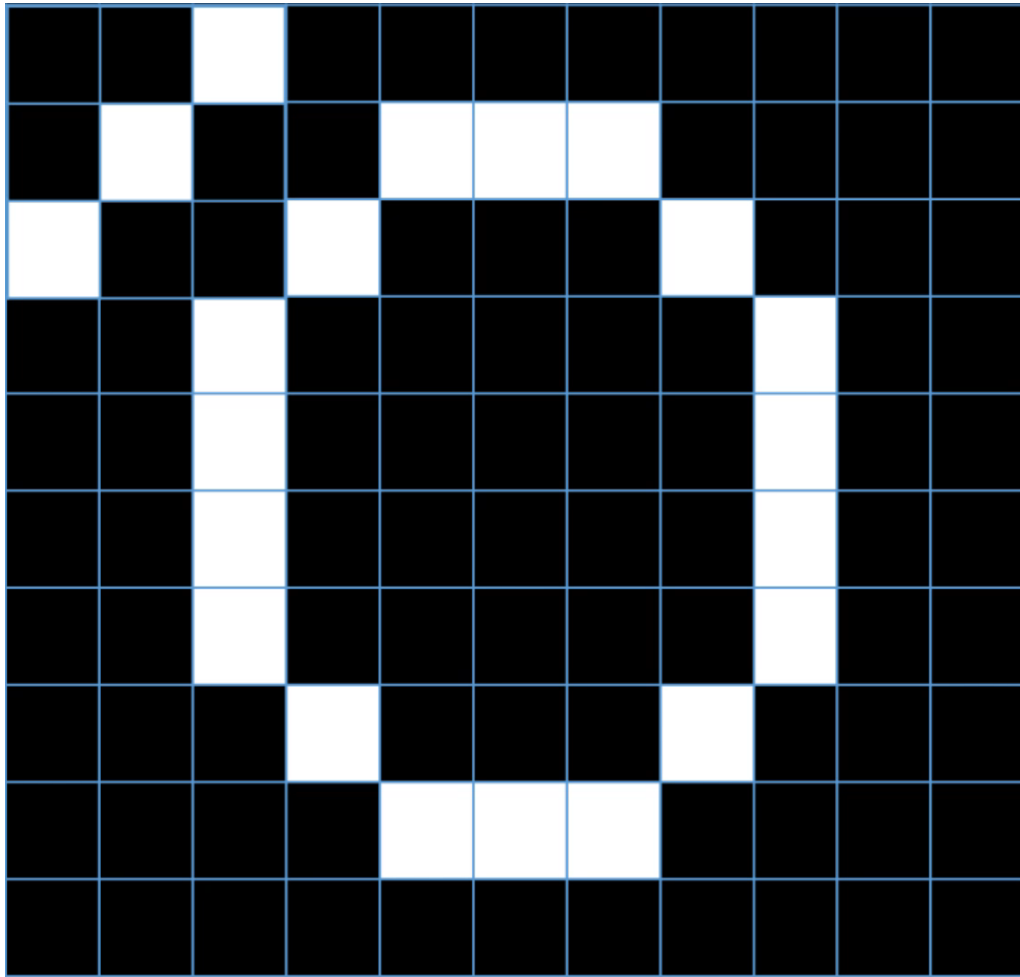
ニューラル  
ネットワーク？

# 画像データの構造

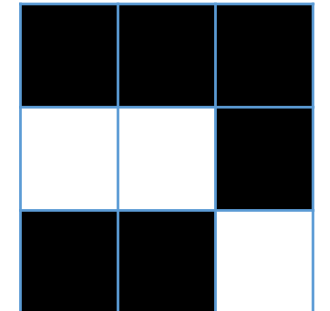
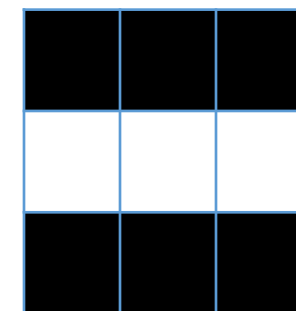
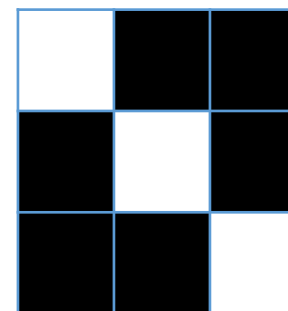
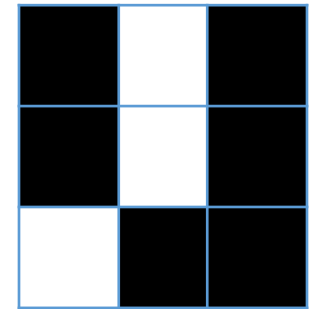
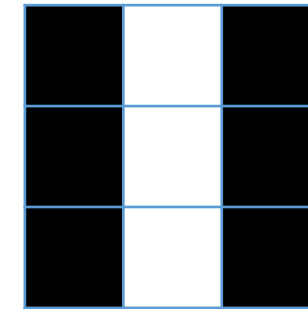
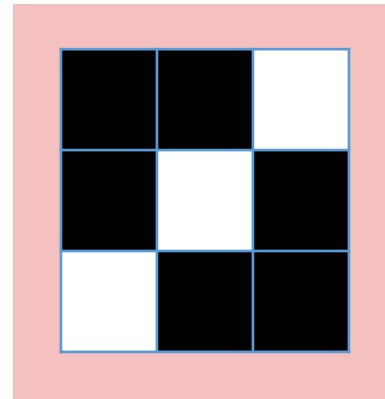
- 画像はRGBチャンネルごとの輝度配列（M行N列の数値データ）を重ねたもの



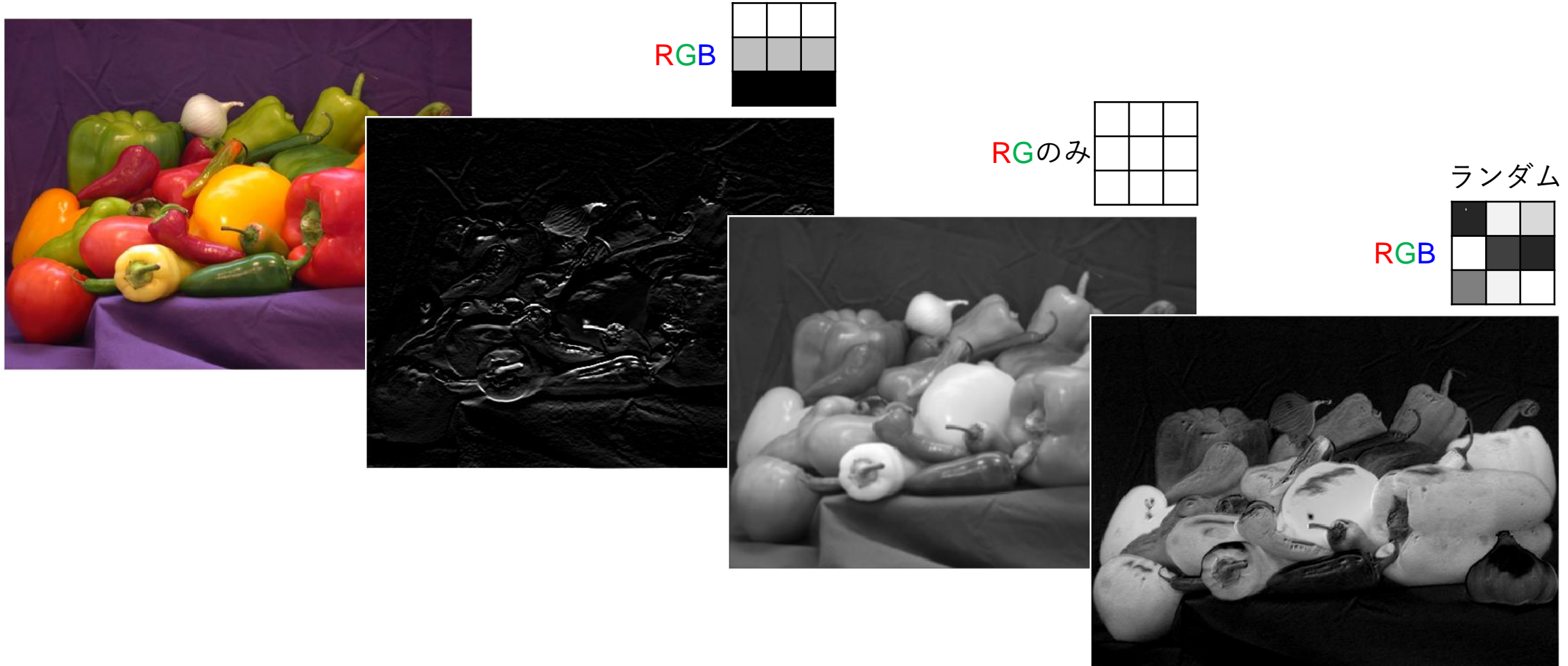
畳み込みとは、テンプレートとの類似度を場所ごとに計算すること



テンプレート画像 (カーネル)

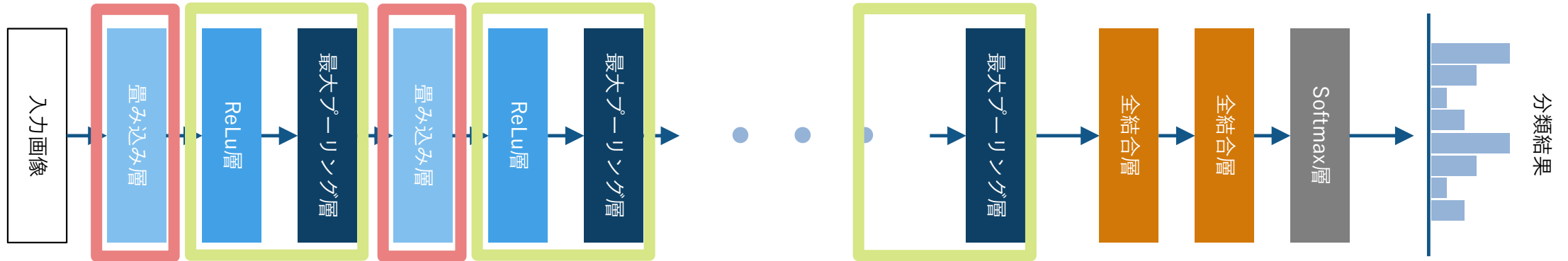


# 畳み込みによる特徴抽出例



カーネルを変えることで、様々な特徴を取り出すことができる

# 分類CNNのネットワーク基本構造



単純

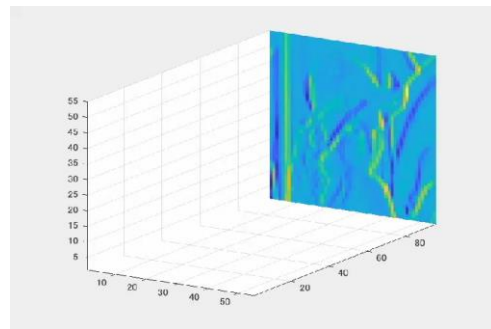
(理解できる)

複雑

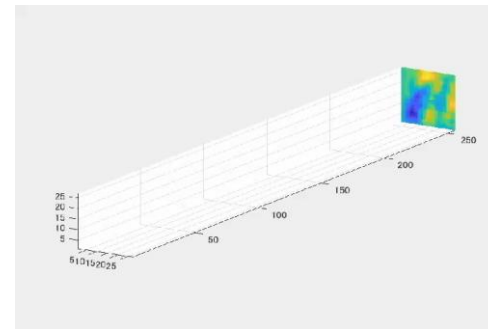
(解釈が難しい)



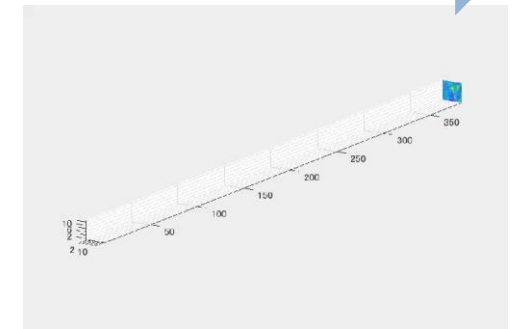
RGBの3チャンネル



1番目の畳み込み層の応答  
96チャンネル



2番目の畳み込み層の応答  
256チャンネル

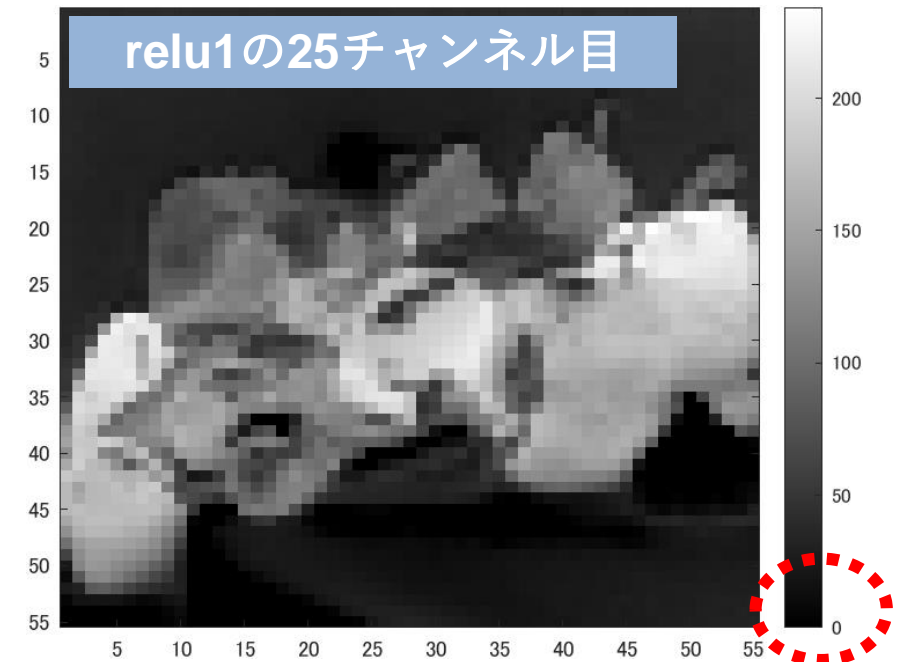
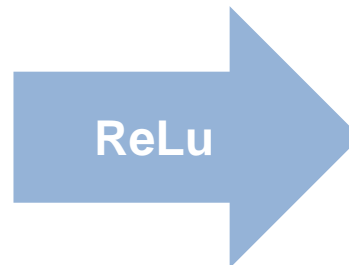
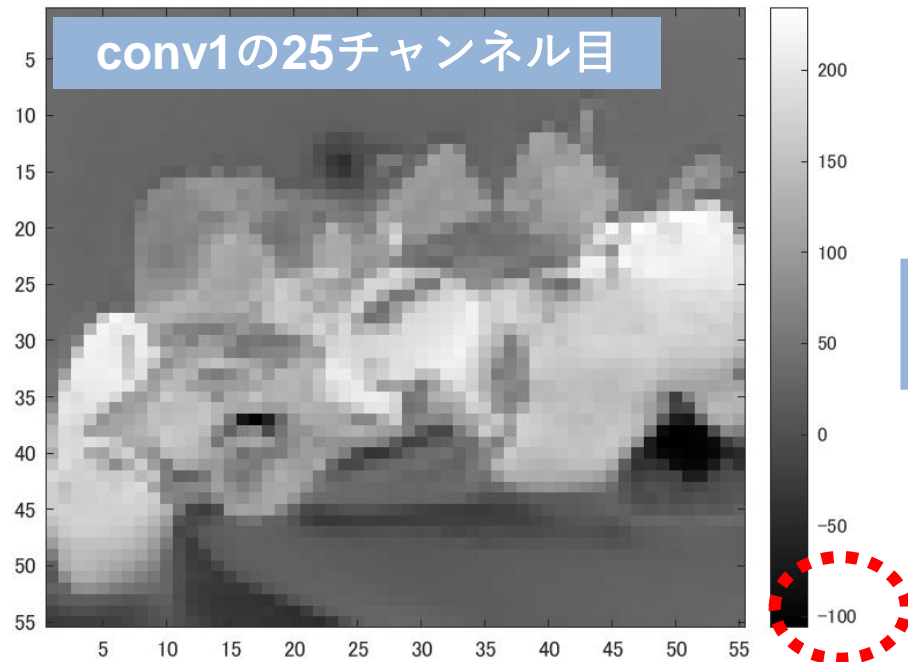
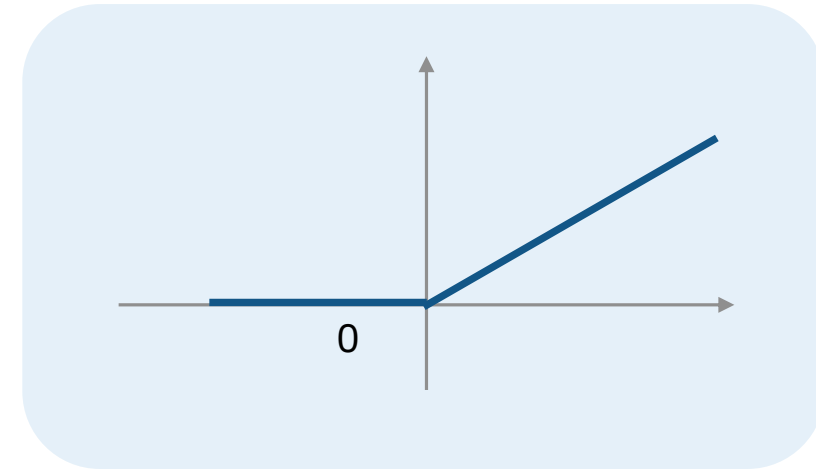


3番目の畳み込み層の応答  
384チャンネル



# ReLUとは

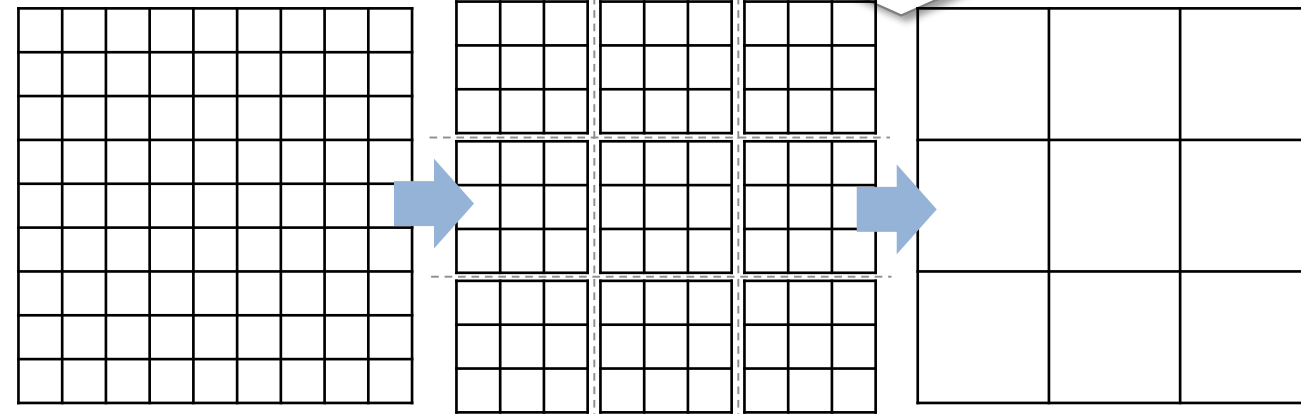
- ReLu: Rectified Linear Unit
- 活性化関数の一つ
  - 活性化 = 非線形
- 負の入力は通さない
- 正の入力そのまま通す
- 強い特徴量だけを取り出す効果



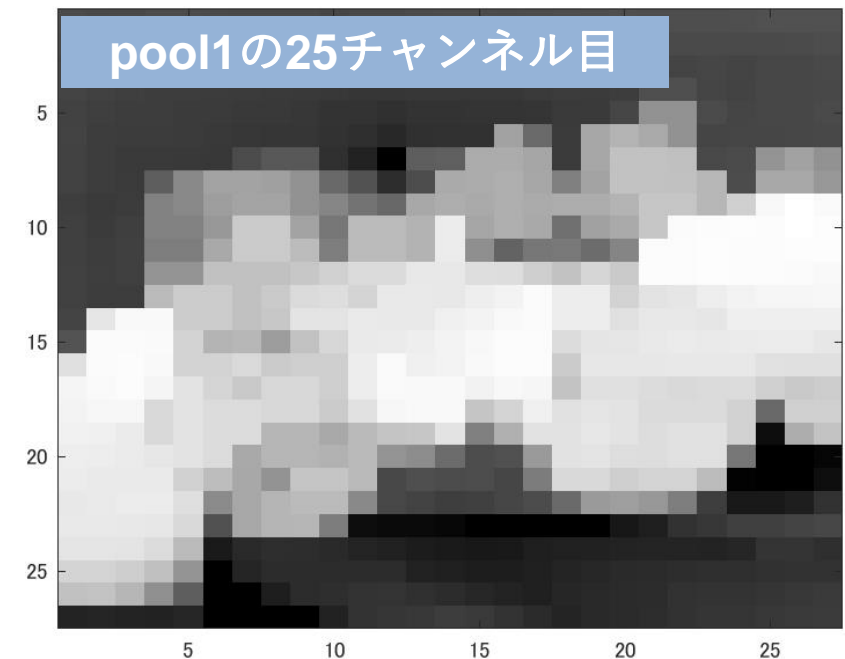
# Pooling Layer (プーリング層)

- 局所領域内での最大値または平均値を出力
- 平行移動等に対するロバスト性に関係
- 通常間引きを伴うことで、計算負荷を低減
- 大局的な特徴抽出に繋がる

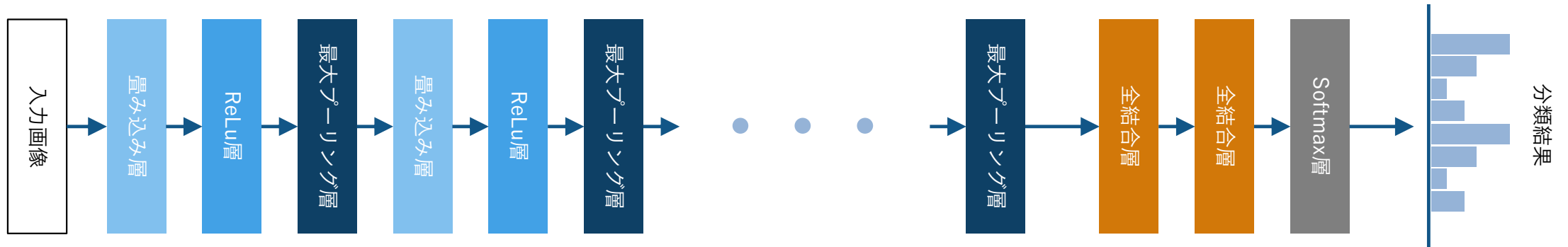
最大値を出力する場合 : Max Pooling  
平均値を出力する場合 : Average Pooling



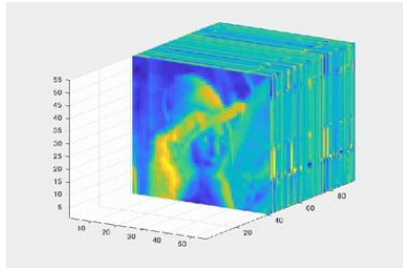
max pooling



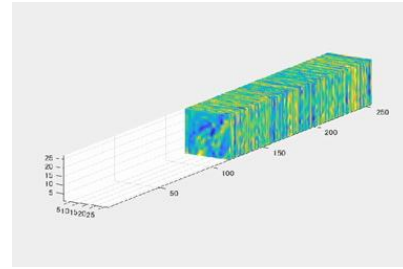
# CNNで得られる特徴



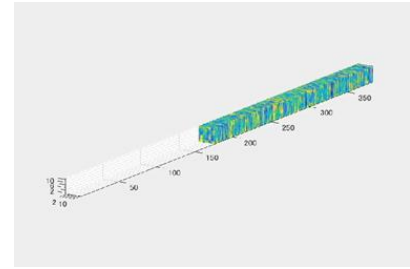
227 x 227 x 3 1番目の畳み込み層の応答  
55 x 55 x 96



2番目の畳み込み層の応答  
27 x 27 x 256



3番目の畳み込み層の応答  
13 x 13 x 384

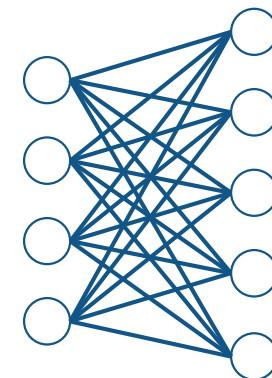


|           |     |     |
|-----------|-----|-----|
| 層の深さ      | 浅い  | 深い  |
| チャンネル数    | 少ない | 多い  |
| 画像サイズ     | 大きい | 小さい |
| 特徴の複雑さ    | 単純  | 複雑  |
| 特徴の空間スケール | 局所  | 大局  |
| 特徴の解釈     | 簡単  | 難しい |

# 全結合層

- 入力の配列・ベクトルの全要素に対する線形結合
- 成果物 = 特徴ベクトルと見なすことができる

$$y = \sum_{k=1}^n w_k \cdot x_k + b$$



画像（特徴マップ）を特徴ベクトルに変換する



全結合

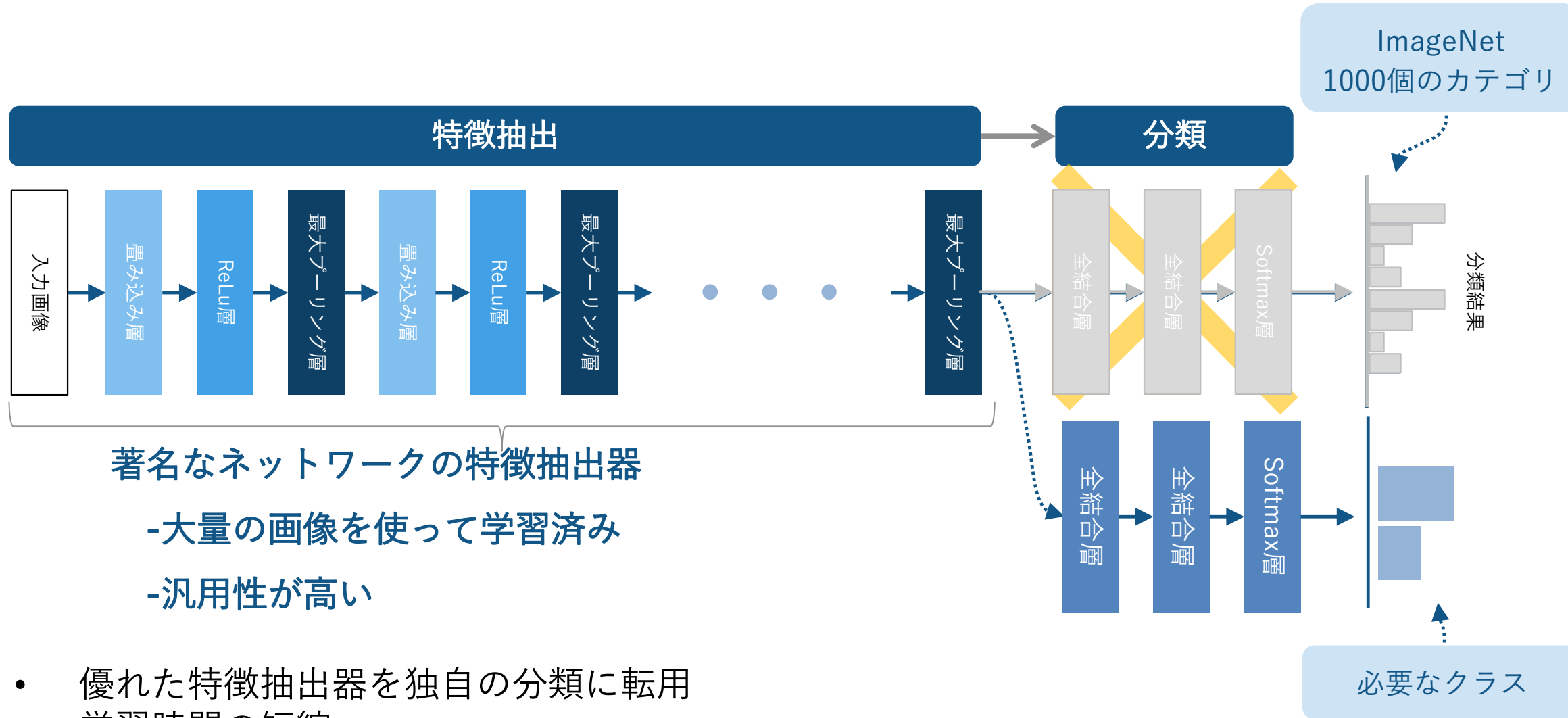
出力クラス数分の長さを持つベクトル

softmax層（正規化）



クラス分類

# 転移学習による分類器の学習



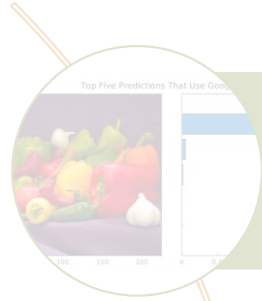
- 優れた特徴抽出器を独自の分類に転用
- 学習時間の短縮
- 少量の学習データで高精度が期待できる



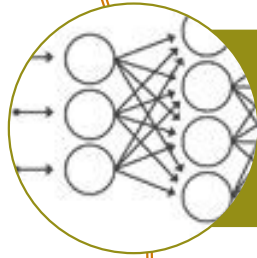
# 画像のディープラーニングのまとめ

## 特徴抽出の過程を可視化して確認

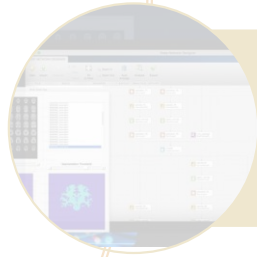
- 画像のディープラーニング = 畳み込みニューラルネットワーク (CNN)
- 畳み込み演算によって画像から特徴を抽出できる
- 異なる役割を持った層に画像データを通していくことで、複雑で大域的な特徴を抽出することができる。
  - 畳み込み
  - 活性化
  - プーリング
  - 全結合
- ネットワークは特徴抽出のための層と、特定のタスク（分類ネットワークなら分類）のための層に分けて考えることができる。
- 転移学習で手軽に高精度が期待できる。



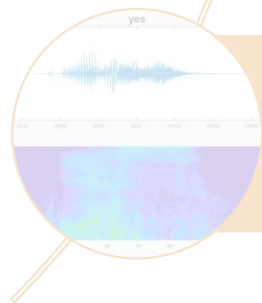
画像のディープラーニング



ニューラルネットワークとは？



実験を高速で繰り返すために



さらに学習したい人に

## 画像のディープラーニング

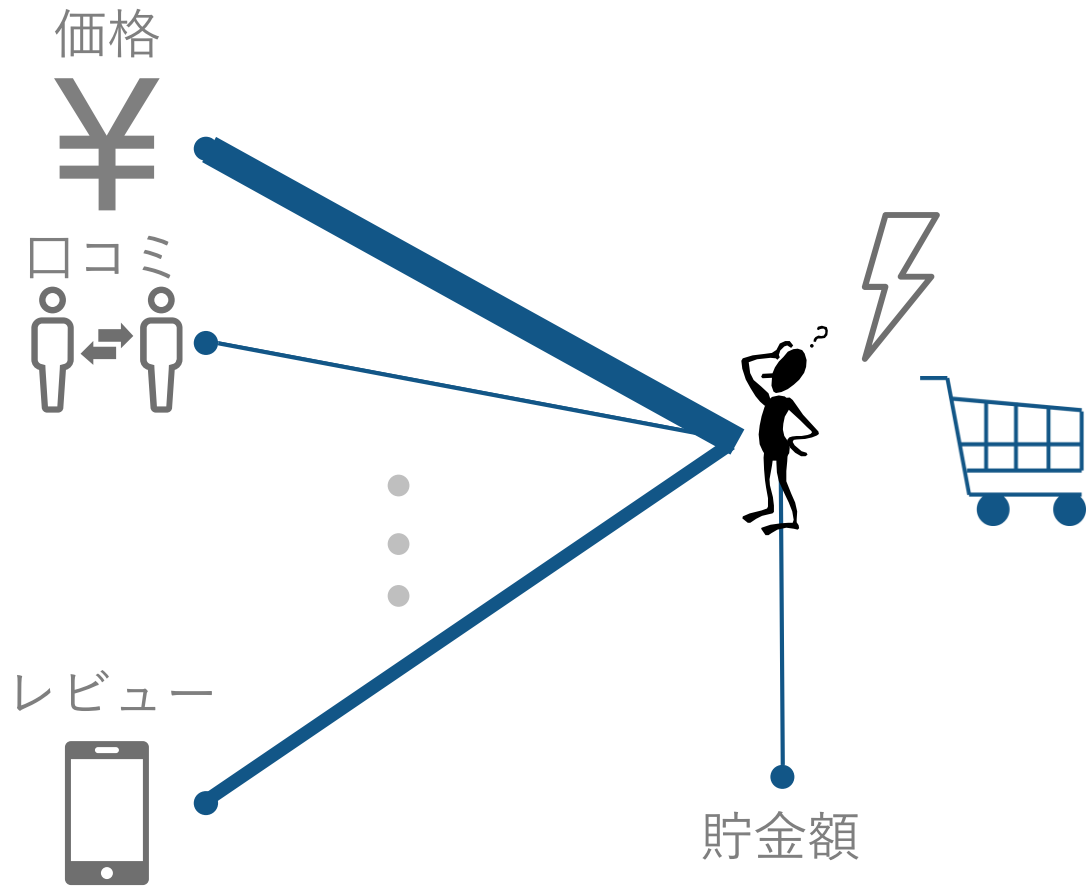
- 畳み込みニューラルネットワーク
- (**CNN: Convolutional Neural Network**)



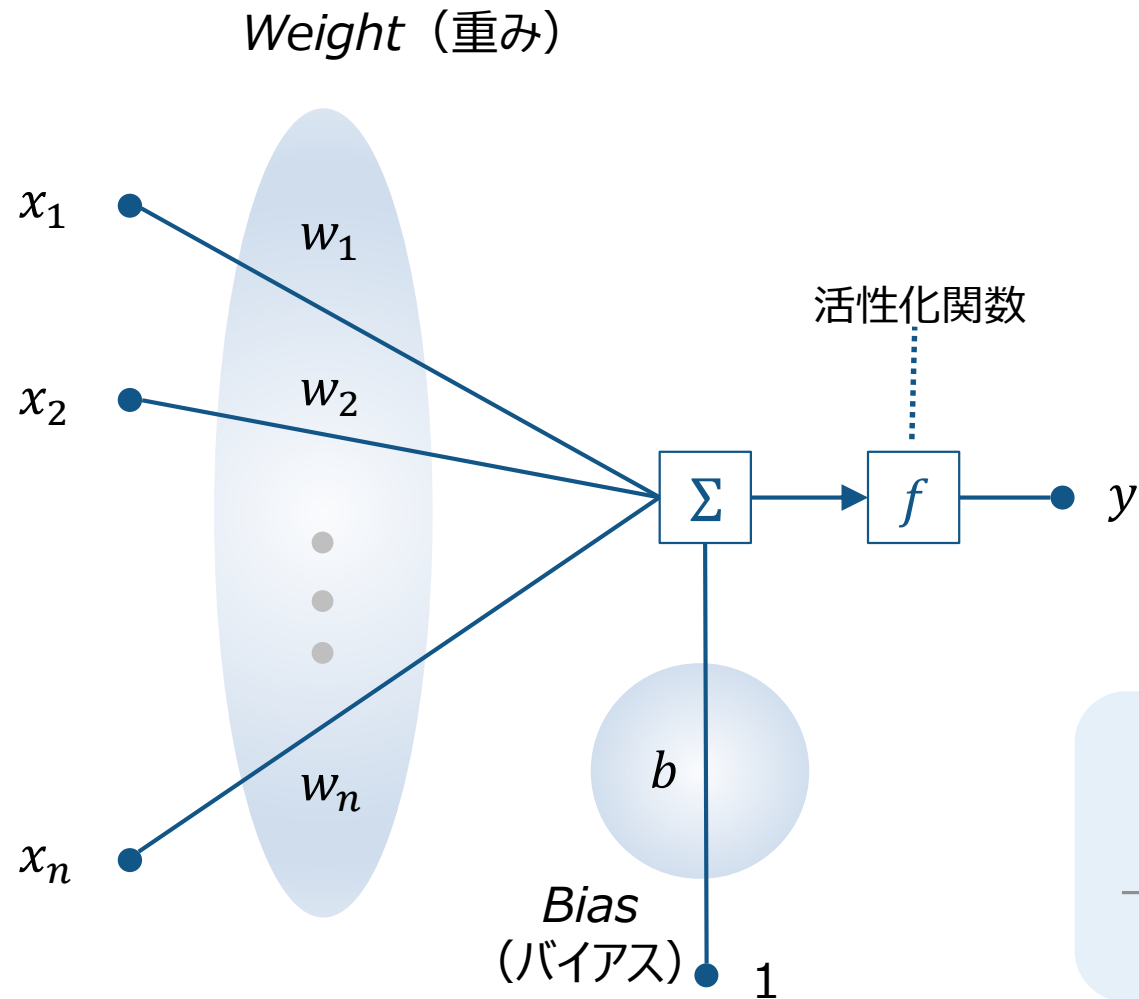
畳み込み？

ニューラル  
ネットワーク？

# 欲しいものがあるとき、どうしていますか？



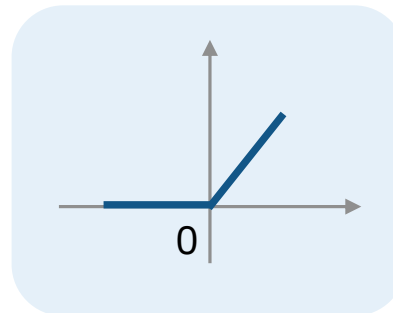
# ニューラルネットワークの基本構成



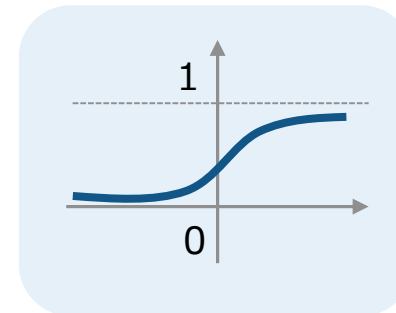
モデル式

$$y = f \left( \sum_{k=1}^n w_k \cdot x_k + b \right)$$

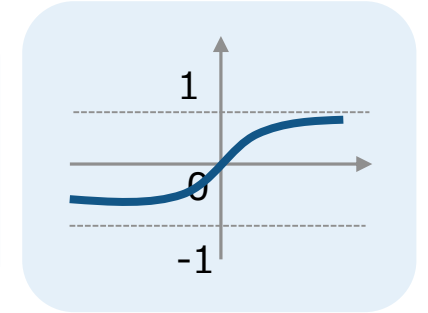
活性化関数  
(非線形)



Rectified Linear Unit  
(ReLU)



sigmoid



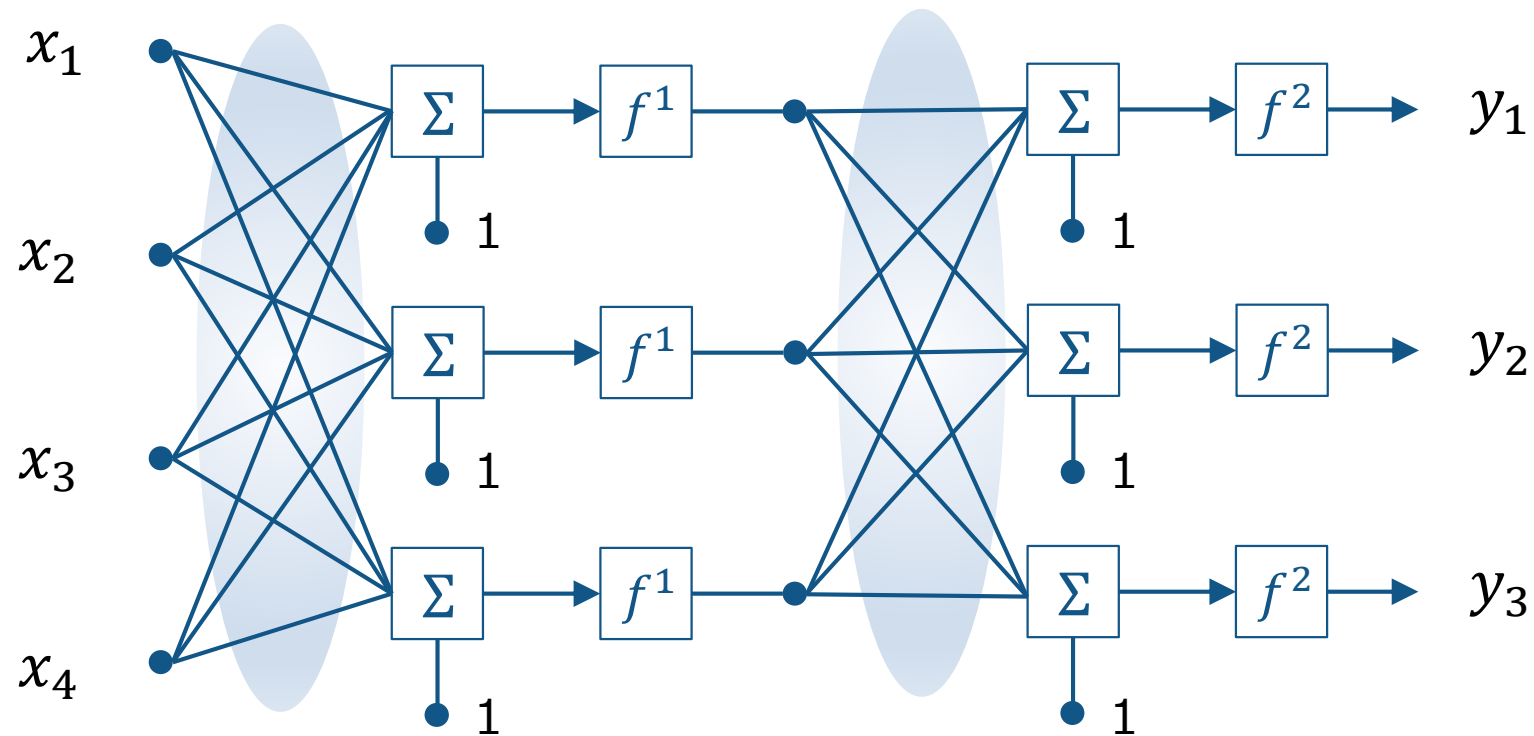
tanh



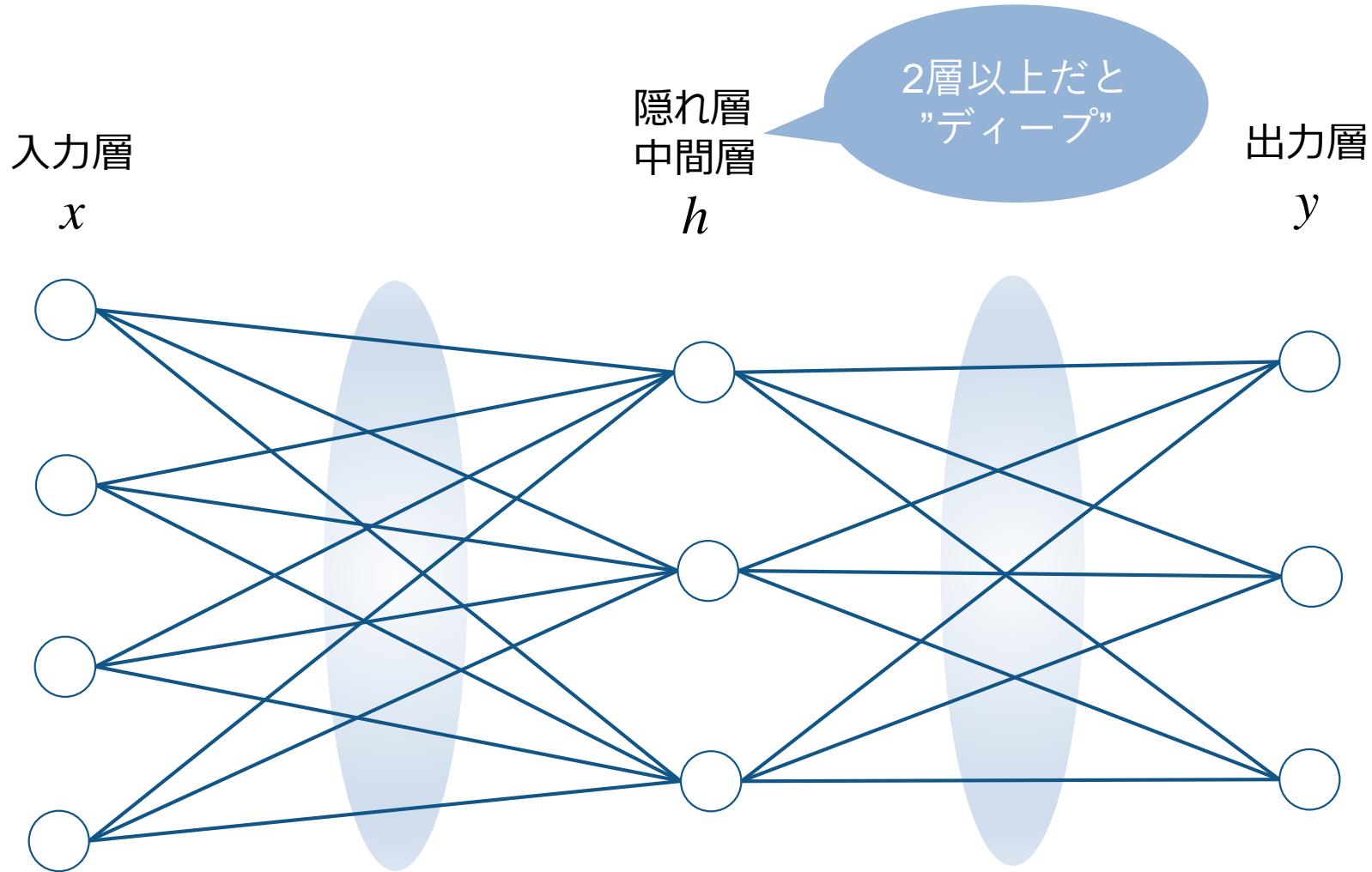
# ニューラルネットワーク

## ニューラルネットワークの普遍性定理

任意の関数を任意の精度で近似できることが証明されている



# ニューラルネットワークの簡略的な表現



重み、バイアスはどのように決定すればいいのか

→学習の仕組みは？

# 最小二乗法でのモデル最適化の復習

- 最小二乗法（1次式の場合）を使った予測

## モデル式

$$f(x) = ax + b$$

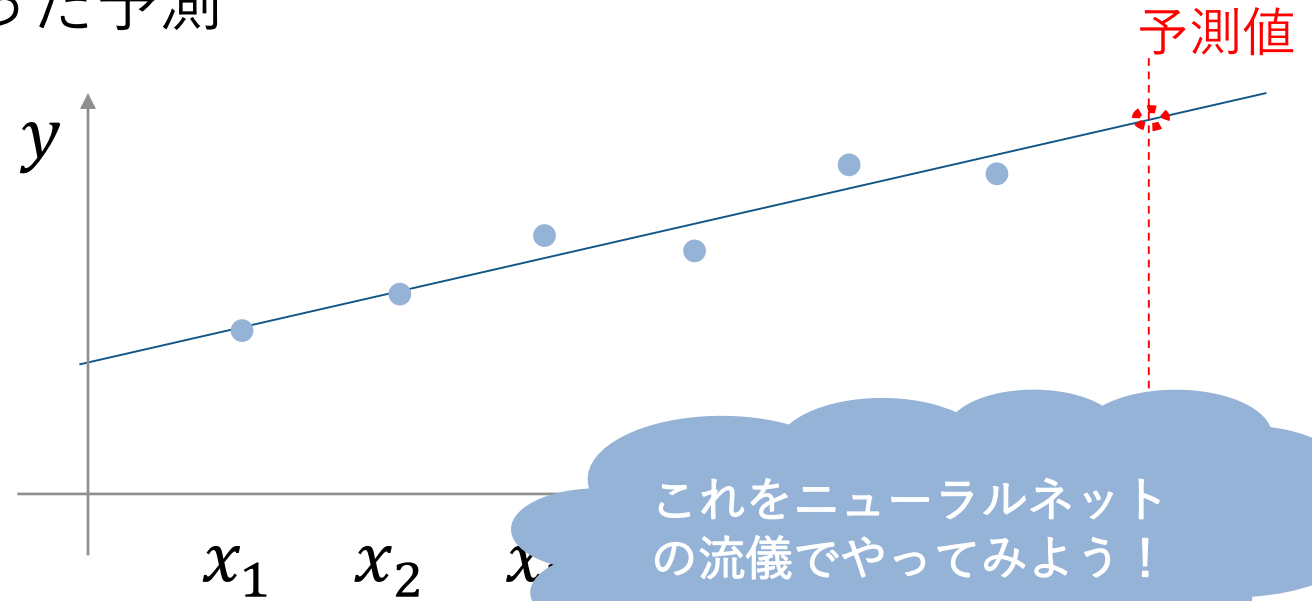
## モデル式による値と実測値との差

$$L = \sum_{i=1}^N (y_i - f(x_i))^2$$

Lが最小となるa,bを決めたい!

$$\frac{\partial L}{\partial a} = 2a \sum x_i^2 - 2 \sum x_i y_i + 2b \sum x_i$$

$$\frac{\partial L}{\partial b} = 2Nb + 2a \sum x_i - 2 \sum y_i$$



これをニューラルネットの流儀でやってみよう!

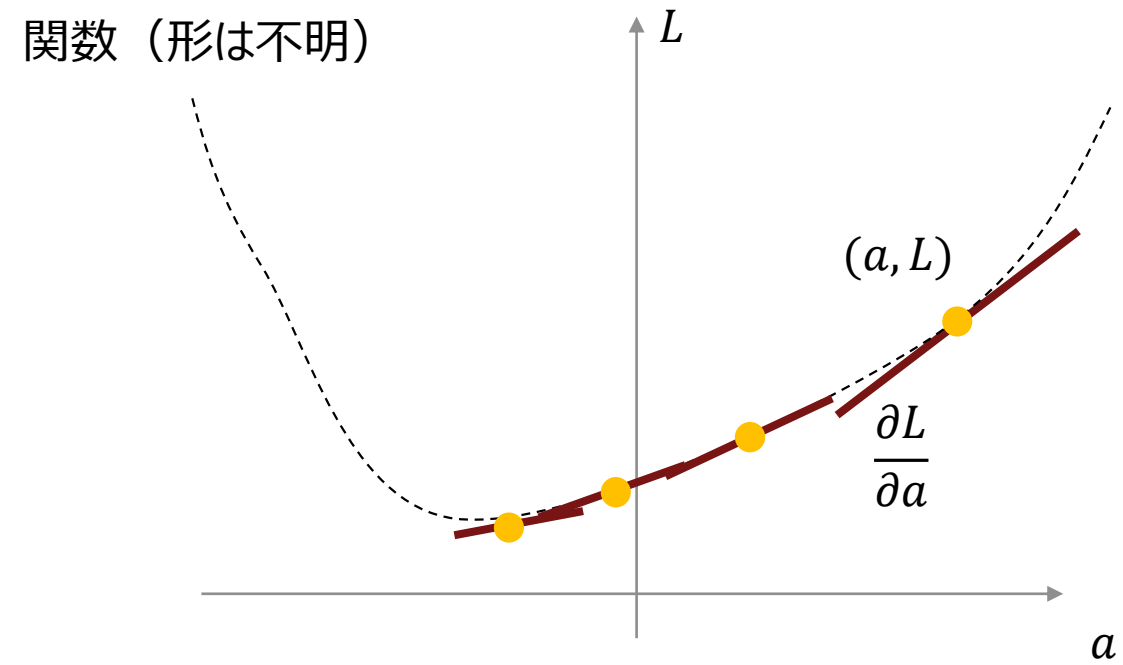
それぞれイコール0と置いて、  
a,bについて連立方程式を立てて解く!



# 勾配法によるパラメータの最適化

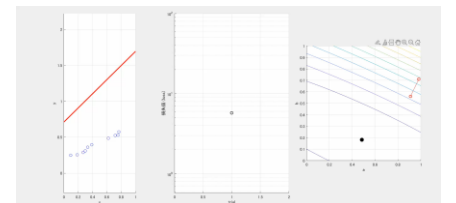
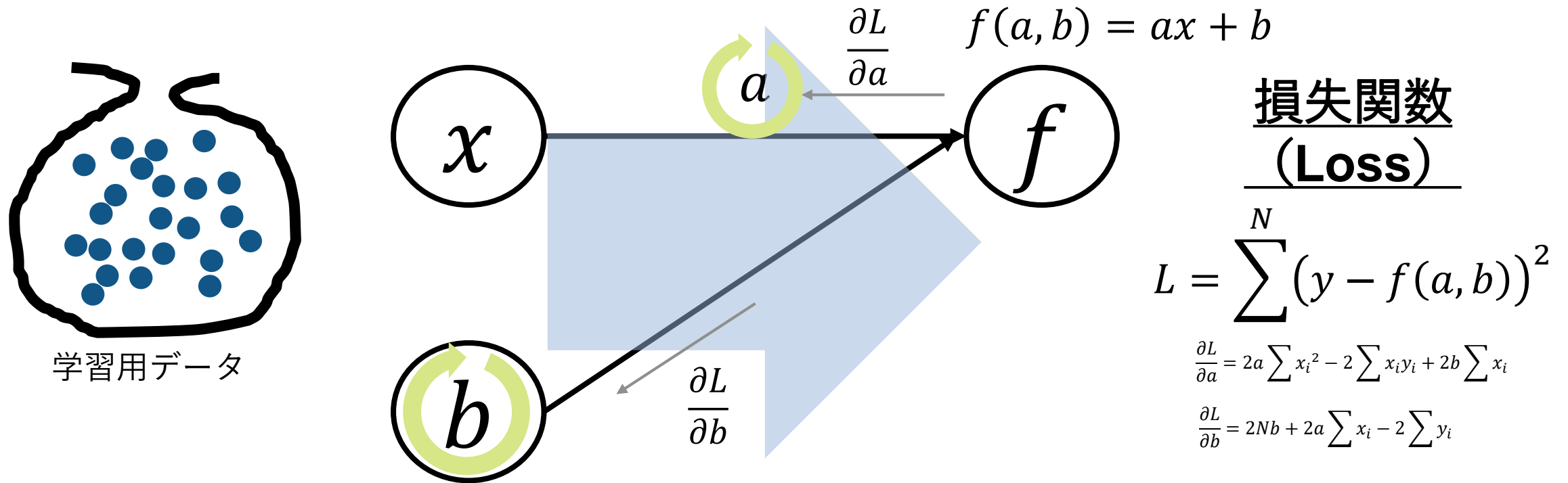
- 微分値（勾配）がわかっているとき、値を小さくする方にパラメータを更新していく

$$\begin{array}{c}
 \text{更新後の重み} \\
 a \leftarrow a - \eta \frac{\partial L}{\partial a} \\
 \text{重み} \qquad \qquad \text{係数} \qquad \qquad \text{勾配}
 \end{array}$$



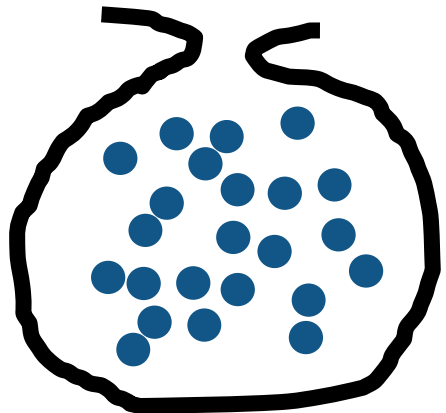
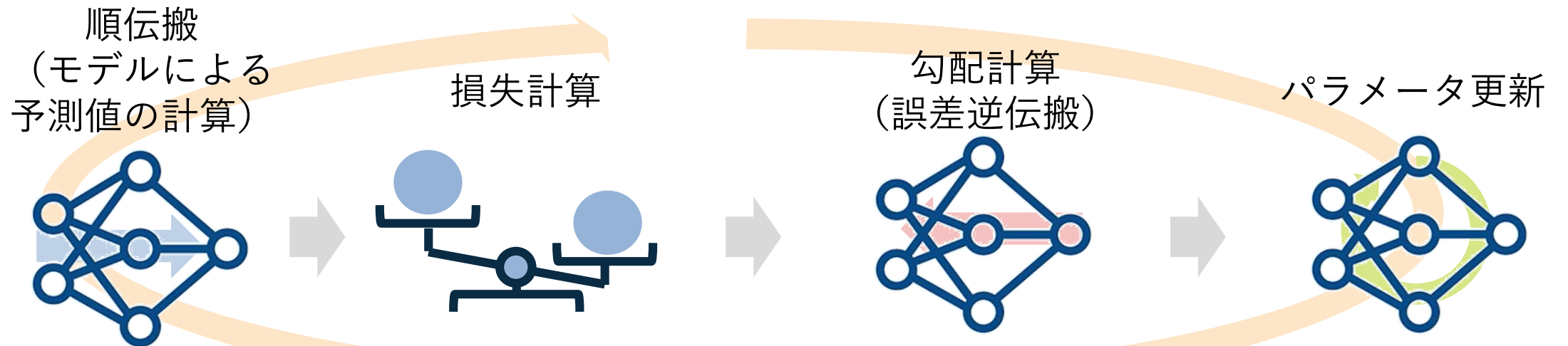
# 1次方程式モデルの勾配法での最適化

## 最小二乗法をニューラルネットワークに変換





# 【重要】ディープラーニングの学習のフロー



学習用データ

# 各学習パラメータにおける勾配の計算 連鎖律 (チェーンルール)



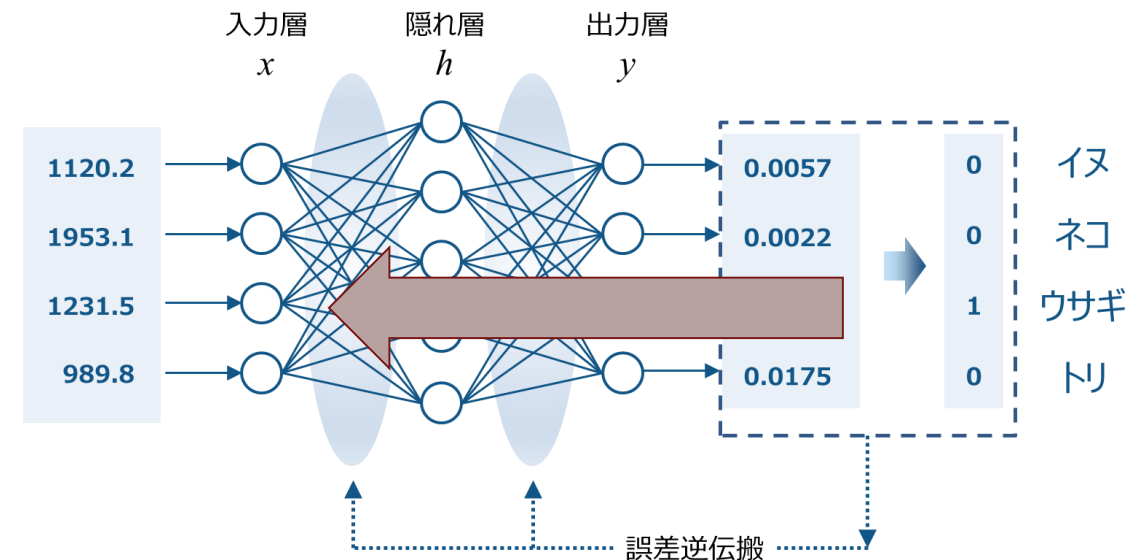
- 合成関数の勾配 (微分) は各関数の勾配の積から求められる。

$$y = L(f(x)) \text{ のとき、 } \frac{\partial y}{\partial x} \text{ は、 } y = L(u), u = f(x) \text{ とすると}$$

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial u} \frac{\partial u}{\partial x} \text{ で求めることができる。}$$

## 誤差逆伝搬

- ネットワーク上の各学習パラメータの勾配も、損失関数の勾配から逆にたどることで芋づる式に求めていける。
- これを自動で行う自動微分という仕組みがプログラミング言語に実装

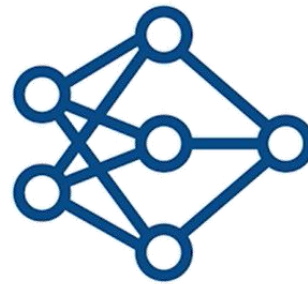


# 手書き数字認識CNNの学習

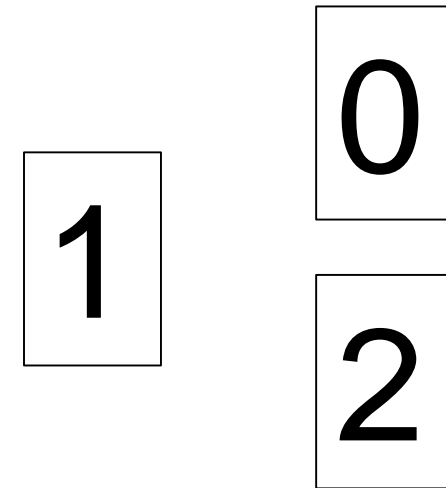
手書き数字の画像



ニューラル  
ネットワーク



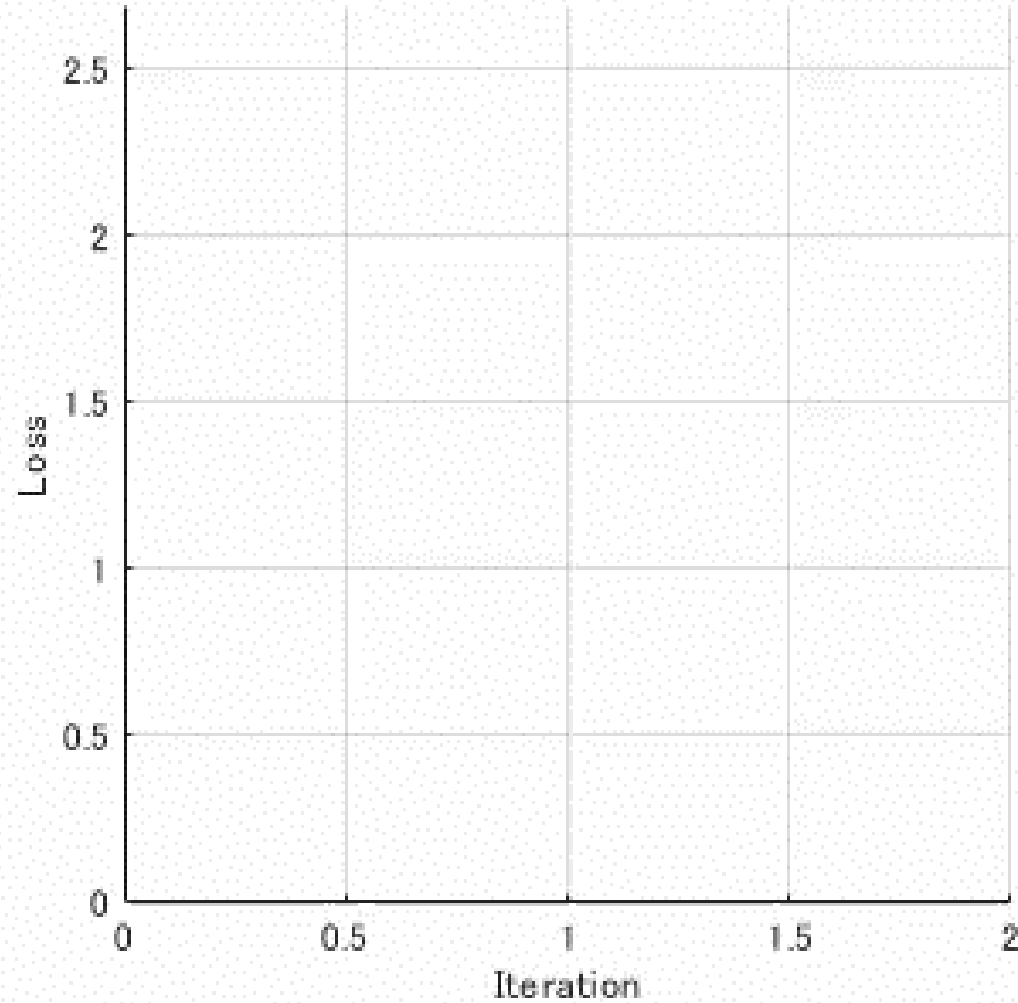
数字のクラス



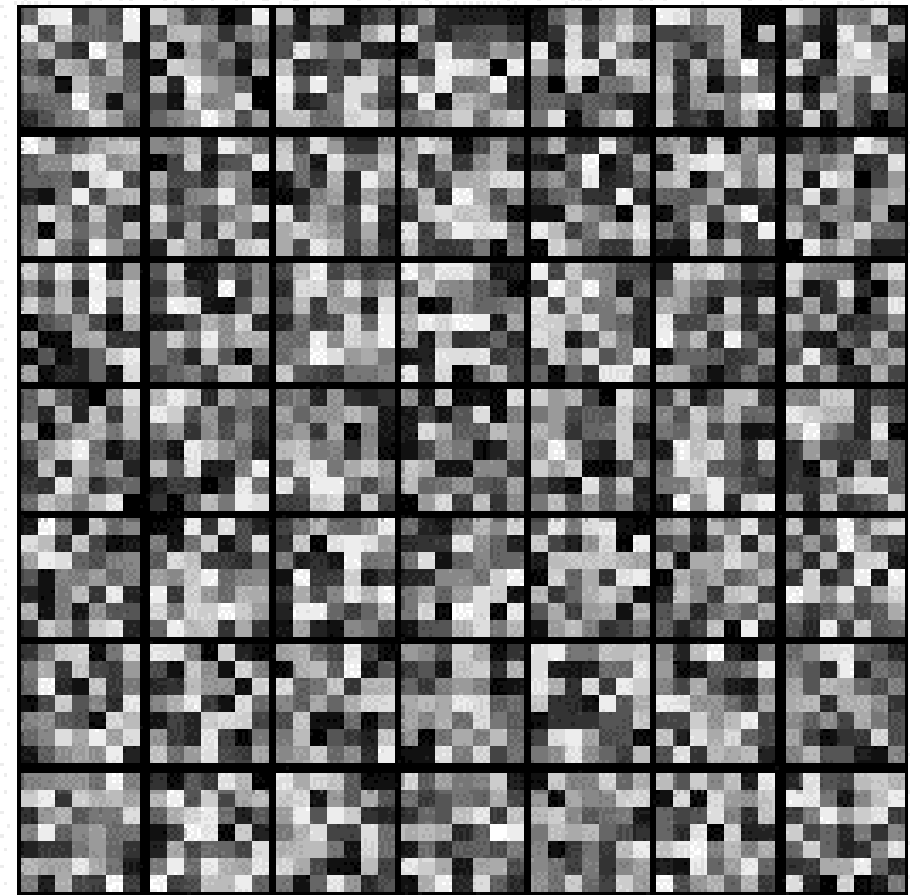
# 学習過程の可視化

## 損失値の推移

Epoch: 1, Elapsed: 00:00:00



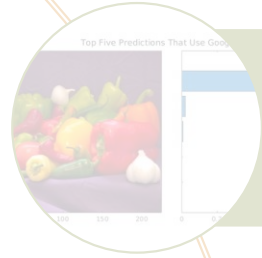
## 1番目の畳み込み層の重み



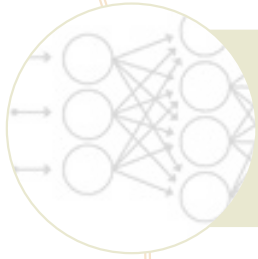
## ニューラルネットワークとは？のまとめ

### 学習過程を可視化して確認

- ニューラルネットワークを使って任意の関数をモデル化（近似）できる。
- モデルのパラメータを最適化するには、損失関数を最小化する。
- 損失関数を最小化するため、勾配法を使う。
- 各パラメータの勾配は、連鎖律で芋づる式に計算できる（自動微分）。
- 中間層の計算および損失関数は全て微分可能でなければならない。



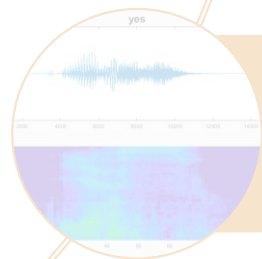
画像のディープラーニング



ニューラルネットワークとは？



実験を高速で繰り返すために



さらに学習したい人に



# 実際にディープラーニングを試し始めてみると...

データセットの量は？

学習率とかのハイパーパラ  
メータはどうしたら？

どのネットワークを使えば  
いいんだろう？

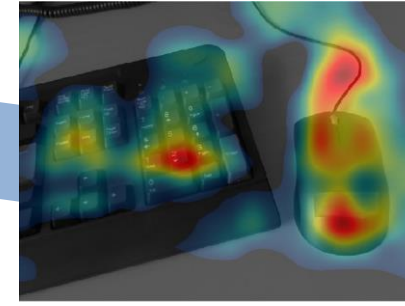
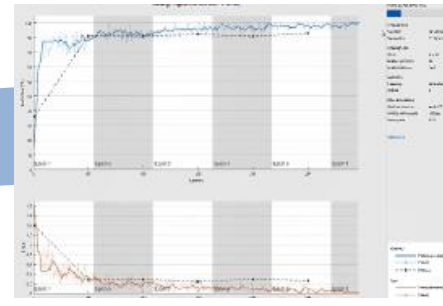
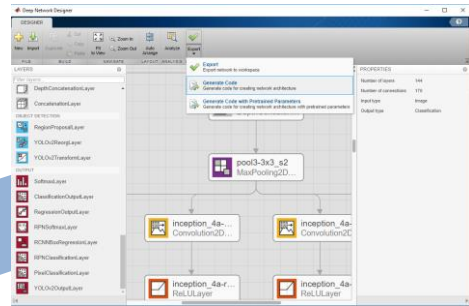
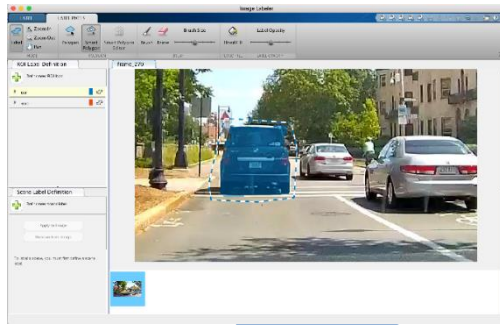


絶対の指針は現状ありません

だからこそ...

**実験を繰り返すことが必要・経験の蓄積**

# MATLABはワンストップの環境で全ワークフローをカバー



データセットの作成と  
アクセス

前処理と変換

予測モデルの開発

実システムへの展開

ディープラーニングの高速・効率的な試行錯誤をサポート

## データセットの作成とアクセス

### ■ データソース



### ■ シミュレーション ■ オグメンテーション

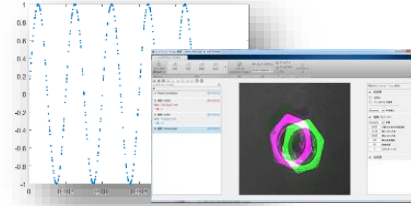


### ■ データラベリング

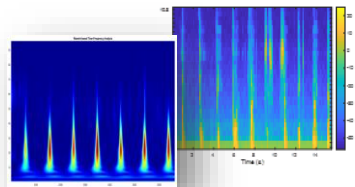


## 前処理と変換

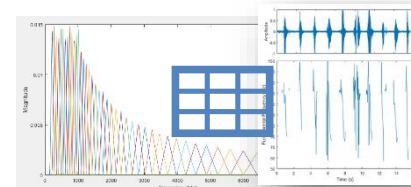
### ■ 前処理



### ■ 変換



### ■ 特徴抽出

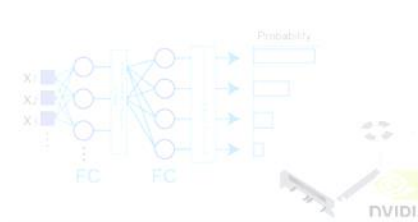


## 予測モデルの開発

- リファレンスモデルの流用
- オリジナルモデルの作成



- GPUによる高速な学習



- ハイパーパラメータの解析とチューニング



## 実システムへの展開

### ■ デスクトップアプリ



### ■ エンタープライズシステム

Java  
MATLAB  
C/C++  
Python

### ■ エッジ（組み込み）デバイス



# MATLABが提供する計測ハードウェアとのインターフェース

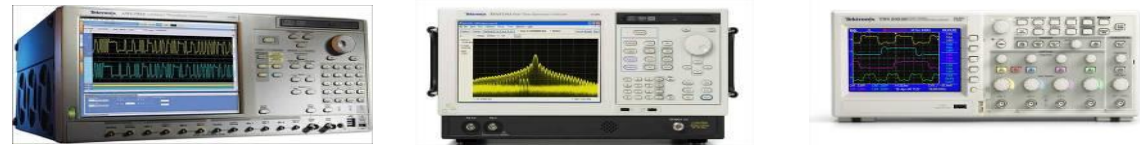
**Image Acquisition Toolbox**  
 イメージキャプチャデバイス



**Data Acquisition Toolbox**  
 プラグインデータ収集  
 カード・ボード



**Instrument Control Toolbox**  
 計測器/ RS-232 etc



**MATLAB**  
 上記ハードウェア以外との  
 通信





# sampleからアクセス可能なテスト用画像データリスト

|                        |                               |                        |                            |                                   |
|------------------------|-------------------------------|------------------------|----------------------------|-----------------------------------|
| <p><b>Digits</b></p>   | <p><b>MNIST</b></p>           | <p><b>Omniglot</b></p> | <p><b>Flowers</b></p>      | <p><b>Example Food Images</b></p> |
| <p><b>CIFAR-10</b></p> | <p><b>MathWorks Merch</b></p> | <p><b>CamVid</b></p>   | <p><b>Vehicle</b></p>      | <p><b>RIT-18</b></p>              |
| <p><b>BraTS</b></p>    | <p><b>Camelyon16</b></p>      | <p><b>COCO</b></p>     | <p><u>and more ...</u></p> |                                   |

# データ管理を簡単に！

## • データストア

- メモリに納まりきれない大量のデータを管理する仕組み
- 学習時に必要な分だけ読み込む
- 数値、画像、時系列データなど用途に応じて多数用意
- 読み込み数の指定や、読み込み時に行う前処理を組み合わせるなどの設定が可能

|  |               |
|--|---------------|
| imageDataStore                                       | <b>R2015b</b> |
| augmentedImageDataStore                              | <b>R2018a</b> |
| audioDataStore<br><i>Audio Toolbox</i>               | <b>R2018b</b> |
| bigimageDatastore<br><i>Image Processing Toolbox</i> | <b>R2019b</b> |
| signalDatastore<br><i>Signal Processing Toolbox</i>  | <b>R2020a</b> |
| arrayDatastore                                       | <b>R2020b</b> |

%% 使い方の例

```
imds = imageDatastore(folderName);
img = read(imds);
```

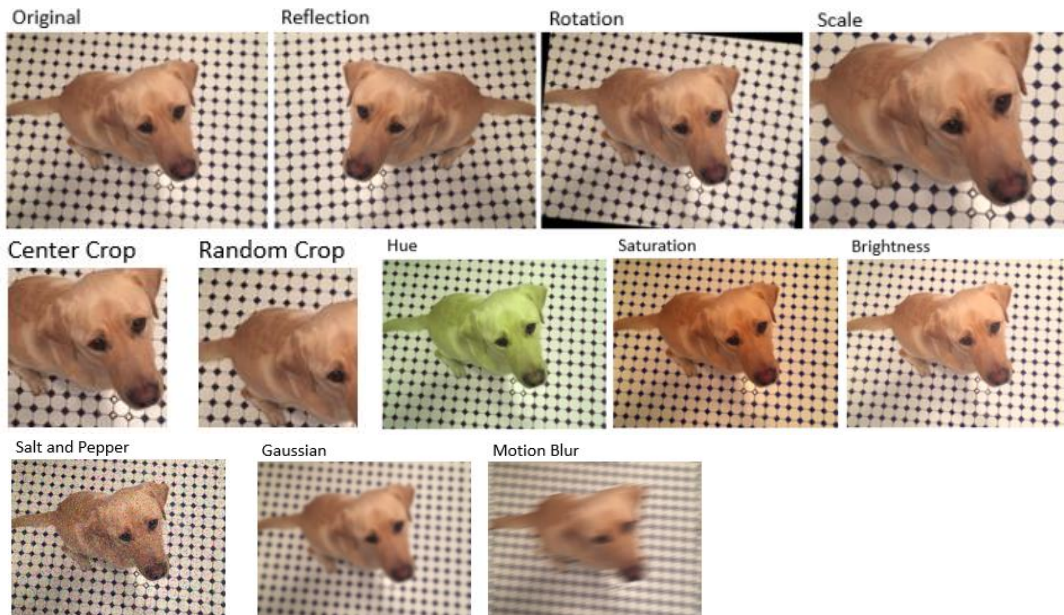
<https://jp.mathworks.com/help/deeplearning/ug/datastores-for-deep-learning.html>

# データ拡張・前処理を短時間で

- データ拡張：データをランダムに加工し疑似的に生成
- 学習時にデータ読み取りとセットで適用

## 画像

- アフィン変換（回転、スケール、せん断、平行移動）
- トリミング
- 色の変換
- ノイズ付加
- ブレ、ボケ

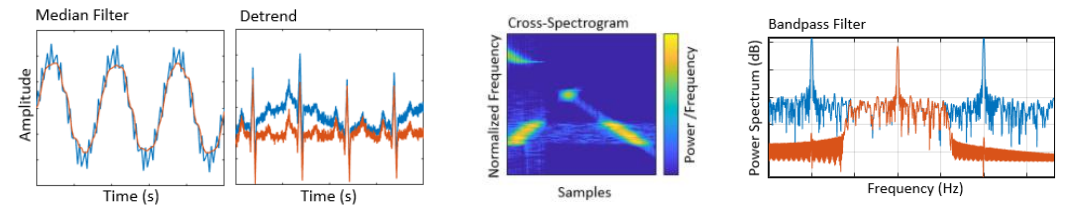


## %% コードの例

```
trds = transform(imds,@preprocess);
```

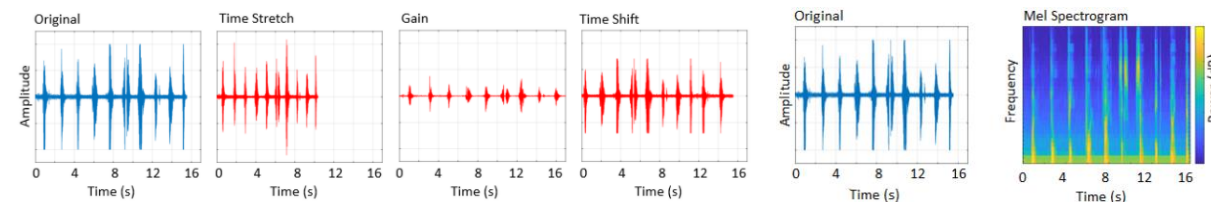
## 一次元信号

ノイズ除去/付加、バンドパスフィルタ  
時間-周波数表現、特徴抽出...など



## 音声信号

- ピッチシフト、ノイズ付加、ボリュームコントロールなどの拡張
- スペクトル統計量などの特徴量
- メルスペクトログラムなどの時間-周波数表現

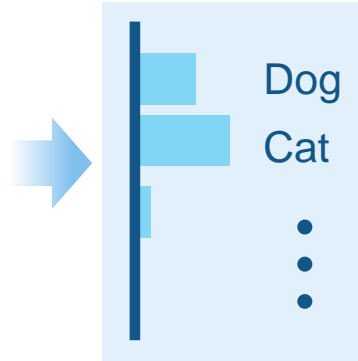




# MATLABによるラベリング

## 物体分類 (1オブジェクト)

CNN (Convolutional Neural Network)



分類

## 物体検出

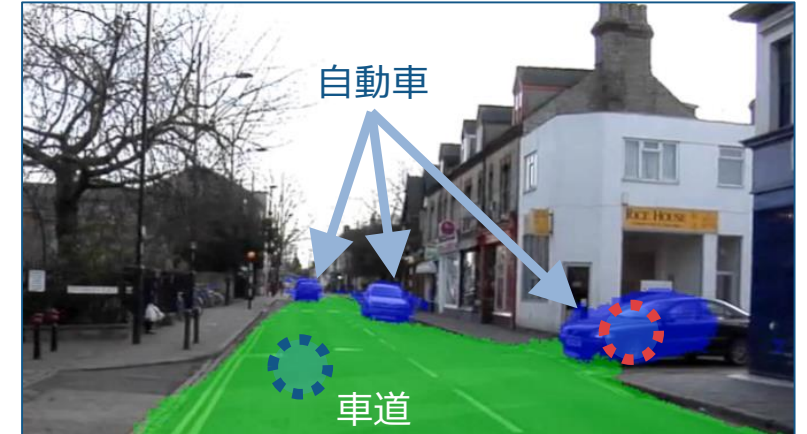
R-CNN / Fast R-CNN / Faster R-CNN  
YOLO v2 / YOLO v3 / SSD



物体の領域 + 分類

## セマンティック セグメンテーション

SegNet / FCN / U-Net / DeepLab v3+



ピクセル単位で分類

詳細度に応じてラベリングにかかる時間も増えていきます

# 専用ツールで効率的なラベリングが可能に！

|            | イメージラベラー  | ビデオラベラー   | 信号ラベラー                                    | オーディオラベラー                                 |
|------------|---|---|---|---|
| 対象         | 画像  | 画像 + ビデオ  | 1次元信号                                     | 音響信号                                      |
| ラベル        | 矩形、直方体、線、ポリゴン<br>Sublabels, Attributes, Scenes                        | 矩形、直方体、線、ポリゴン<br>Sublabels, Attributes, Scenes          | Attributes<br>Regions (区間)<br>Points      | Regions (区間)                              |
| サポート機能・自動化 | ACF車両検出器<br>ACF人物検出器<br>スマートポリゴン<br>塗りつぶし<br>補助付きフリーハンド<br>カスタムアルゴリズム | イメージラベラーの自動化機能一式<br>+ 車線境界線の検出<br>+ ポイントトラッカー<br>+ 時相内挿 | ピーク自動検出<br>スペクトル・スペクトログラム表示<br>カスタムアルゴリズム | 有音部の自動認識<br>サードパーティの自動テキスト変換API利用<br>録音機能 |

## データセットの作成とアクセス

### ■ データソース



### ■ シミュレーション ■ オグメンテーション



### ■ データラベリング

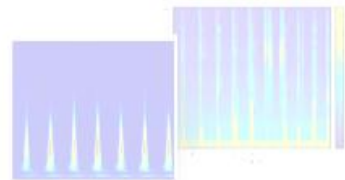


## 前処理と変換

### ■ 前処理



### ■ 変換



### ■ 特徴抽出

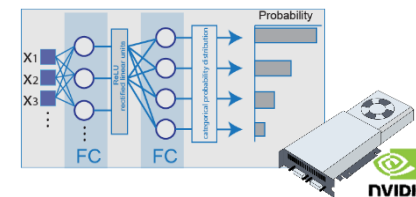


## 予測モデルの開発

### ■ リファレンスモデルの流用 ■ オリジナルモデルの作成



### ■ GPUによる高速な学習



### ■ ハイパーパラメータの 解析とチューニング



## 実システムへの展開

### ■ デスクトップアプリ



### ■ エンタープライズシステム

Java  
MATLAB  
C/C++  
Python

### ■ エッジ（組み込み）デバイス

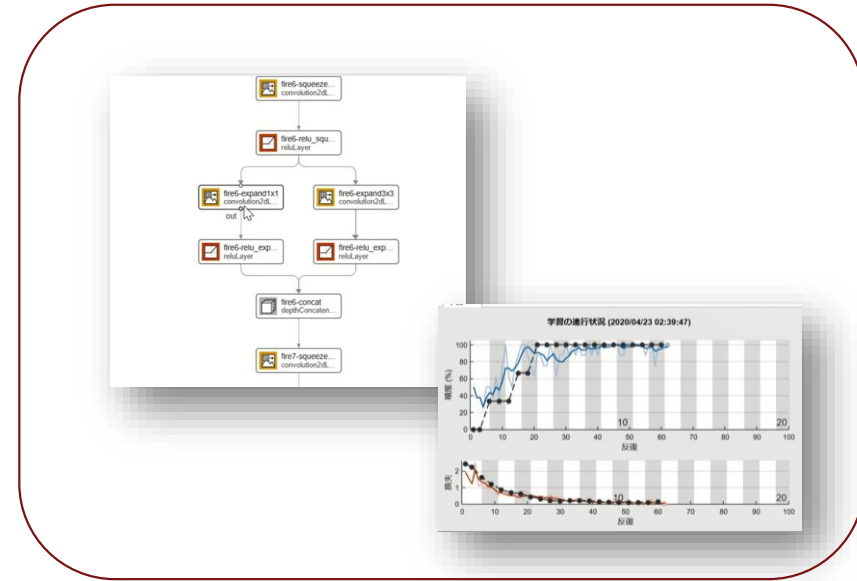


# MATLABでのディープニューラルネットワークの構築

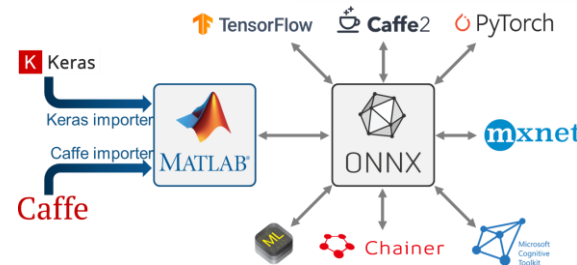


事前学習済み  
モデル

Keras  
ONNX  
Caffe



学習済み  
モデル

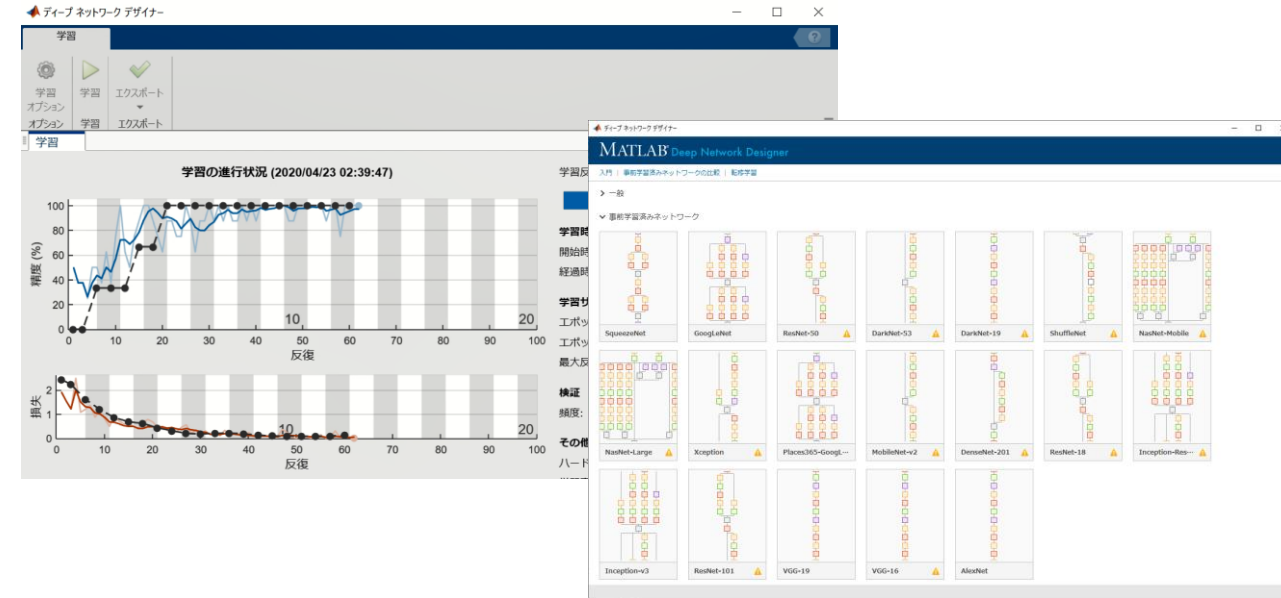


様々なネットワーク構築のパスをサポート

# GUIで素早くモデル学習できる！

- ディープネットワークデザイナー
  - 各種学習済みモデルの提供 **R2020b**
  - ワークスペースからのレイヤ読み込み
  - データの読み込み・水増し設定
  - 学習オプションの設定&学習
  - 自動コード生成

| Network                           | Depth | Size   | Parameters (Millions) | Image Input Size |
|-----------------------------------|-------|--------|-----------------------|------------------|
| <a href="#">squeezeNet</a>        | 18    | 5.2 MB | 1.24                  | 227-by-227       |
| <a href="#">googLeNet</a>         | 22    | 27 MB  | 7.0                   | 224-by-224       |
| <a href="#">inceptionv3</a>       | 48    | 89 MB  | 23.9                  | 299-by-299       |
| <a href="#">densenet201</a>       | 201   | 77 MB  | 20.0                  | 224-by-224       |
| <a href="#">mobilenetv2</a>       | 53    | 13 MB  | 3.5                   | 224-by-224       |
| <a href="#">resnet18</a>          | 18    | 44 MB  | 11.7                  | 224-by-224       |
| <a href="#">resnet50</a>          | 50    | 96 MB  | 25.6                  | 224-by-224       |
| <a href="#">resnet101</a>         | 101   | 167 MB | 44.6                  | 224-by-224       |
| <a href="#">xception</a>          | 71    | 85 MB  | 22.9                  | 299-by-299       |
| <a href="#">inceptionresnetv2</a> | 164   | 209 MB | 55.9                  | 299-by-299       |
| <a href="#">shufflenet</a>        | 50    | 5.4 MB | 1.4                   | 224-by-224       |
| <a href="#">nasnetmobile</a>      | *     | 20 MB  | 5.3                   | 224-by-224       |
| <a href="#">nasnetlarge</a>       | *     | 332 MB | 88.9                  | 331-by-331       |
| <a href="#">darknet19</a>         | 19    | 78 MB  | 20.8                  | 256-by-256       |
| <a href="#">darknet53</a>         | 53    | 155 MB | 41.6                  | 256-by-256       |
| <a href="#">efficientnetb0</a>    | 82    | 20 MB  | 5.3                   | 224-by-224       |
| <a href="#">alexnet</a>           | 8     | 227 MB | 61.0                  | 227-by-227       |
| <a href="#">vgg16</a>             | 16    | 515 MB | 138                   | 224-by-224       |
| <a href="#">vgg19</a>             | 19    | 535 MB | 144                   | 224-by-224       |



**R2020b**

- 学習可能な入力データ
  - 画像データ、時系列データ、複数入力
- 学習可能なタスク **R2020b**
  - 分類
  - セグメンテーション
  - 画像から画像への回帰 **R2020b**

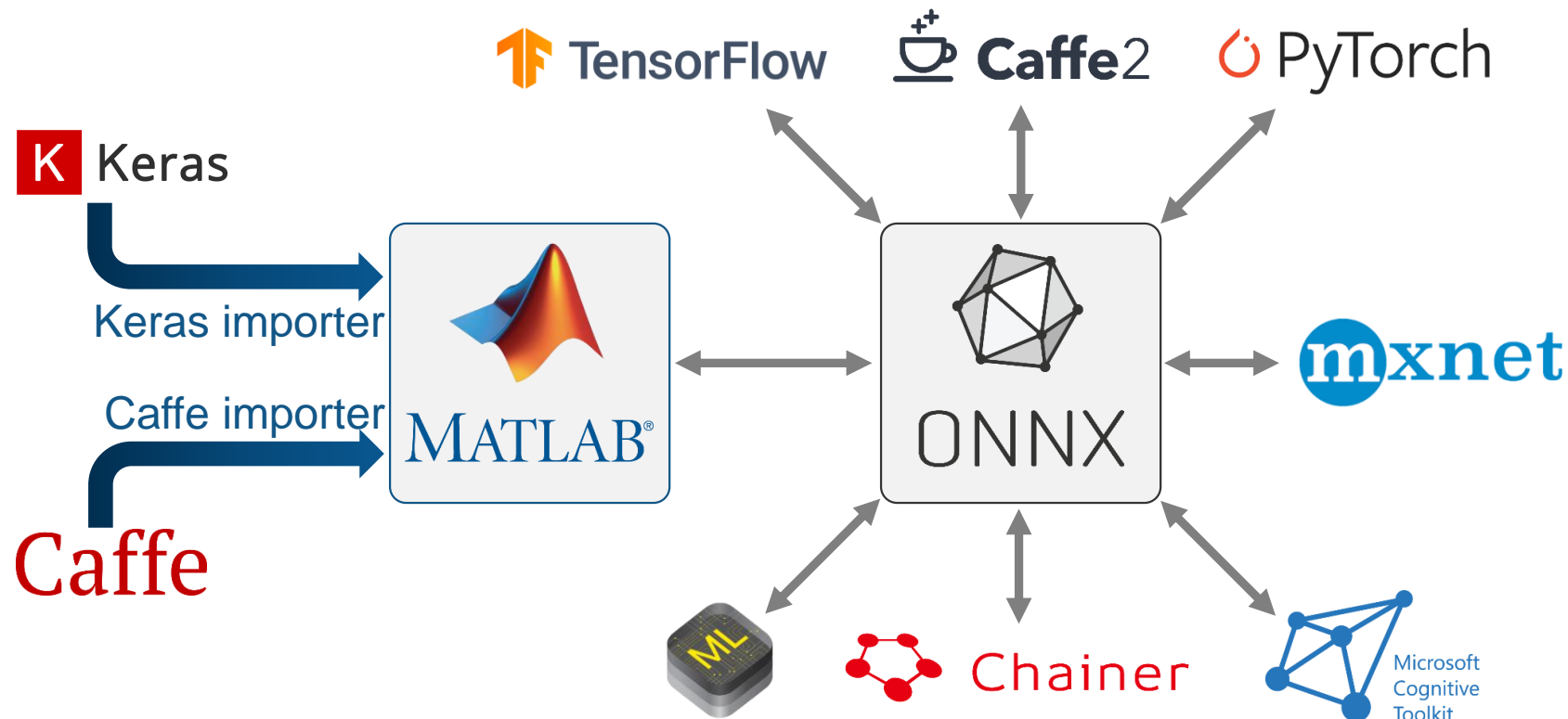


# OSSフレームワークとの連携

- 他フレームワークによる学習済みモデルの取り込み
  - Keras, Caffeモデルの取り込み
  - ONNXフォーマットを介したモデルのやり取り

## 他フレームワークとの連携

```
net = importKerasNetwork(modelfile)
net = importCaffeNetwork(protofile,datafile)
net = importONNXNetwork(modelfile, ...
    'OutputLayerType',outputtype)
```



# 実験をもっと高速に！

## 実験マネージャー app

- ✓ 複数の実験を管理
- ✓ パラメータの履歴をトラッキング
- ✓ グリッドサーチ or ベイズ最適化
- ✓ 結果を解析 & 比較
- ✓ 並列計算が可能

R2020b

(Parallel Computing Toolboxが必要)

The screenshot shows the Experiment Manager app interface. The top toolbar includes buttons for New, Open, Save, Duplicate, Layout, Run, Stop, Training Plot, Confusion Matrix, Filter, and Export. The main area is divided into an EXPERIMENT BROWSER on the left and a Result Details panel on the right. The Result Details panel shows a progress bar for 'Baseline Tuning' (7/16 Trials) and a table of results.

| Trial | Status   | Progress | Elapsed Time      | myInitialLearn... | convFilterSize | Training Accu... | Training Loss | Validation Ac.. |
|-------|----------|----------|-------------------|-------------------|----------------|------------------|---------------|-----------------|
| 1     | Complete | 100.0%   | 0 hr 0 min 16 sec | 1.0000e-6         | 3.0000         | 12.5000          | 2.6441        | 10.             |
| 2     | Complete | 100.0%   | 0 hr 0 min 15 sec | 1.0000e-5         | 3.0000         | 25.7813          | 2.1228        | 20.             |
| 3     | Complete | 100.0%   | 0 hr 0 min 14 sec | 0.0001            | 3.0000         | 64.8438          | 1.0878        | 42.             |
| 4     | Complete | 100.0%   | 0 hr 0 min 16 sec | 0.0005            | 3.0000         | 90.6250          | 0.4648        | 49.             |
| 5     | Complete | 100.0%   | 0 hr 0 min 15 sec | 1.0000e-6         | 4.0000         | 11.7188          | 2.4967        | 6.              |
| 6     | Complete | 100.0%   | 0 hr 0 min 15 sec | 1.0000e-5         | 4.0000         | 23.4375          | 2.1213        | 14.             |
| 7     | Complete | 100.0%   | 0 hr 0 min 17 sec | 0.0001            | 4.0000         | 72.6563          | 1.0283        | 39.             |
| 8     | Running  | 51.4%    | 0 hr 0 min 7 sec  | 0.0005            | 4.0000         |                  |               |                 |
| 9     | Queued   | 0.0%     |                   | 1.0000e-6         | 5.0000         |                  |               |                 |
| 10    | Queued   | 0.0%     |                   | 1.0000e-5         | 5.0000         |                  |               |                 |
| 11    | Queued   | 0.0%     |                   | 0.0001            | 5.0000         |                  |               |                 |
| 12    | Queued   | 0.0%     |                   | 0.0005            | 5.0000         |                  |               |                 |
| 13    | Queued   | 0.0%     |                   | 1.0000e-6         | 6.0000         |                  |               |                 |
| 14    | Queued   | 0.0%     |                   | 1.0000e-5         | 6.0000         |                  |               |                 |
| 15    | Queued   | 0.0%     |                   | 0.0001            | 6.0000         |                  |               |                 |
| 16    | Queued   | 0.0%     |                   | 0.0005            | 6.0000         |                  |               |                 |

The screenshot shows the Experiment Manager app interface. The top toolbar includes buttons for New, Open, Save, Duplicate, Layout, Run, Stop, Training Plot, Confusion Matrix, Filter, and Export. The main area is divided into an EXPERIMENT BROWSER on the left and a Result Details panel on the right. The Result Details panel shows a progress bar for 'Baseline Tuning' (7/16 Trials) and a table of results.

ハイパーパラメーター テーブル:

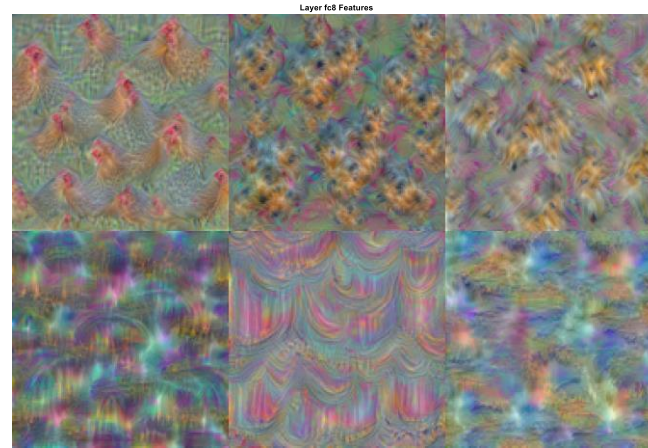
| 名前                 | 値                  |
|--------------------|--------------------|
| myInitialLearnRate | [0.5e-4 1e-4 2e-4] |
| RateFactor         | [1 2 3]            |

モデルパラメータを指定

# CNN 可視化機能

|            |          |            |      |     |      |     |      |       |      |       |
|------------|----------|------------|------|-----|------|-----|------|-------|------|-------|
| airplane   | 923      | 4          | 21   | 8   | 4    | 1   | 5    | 5     | 23   | 6     |
| automobile | 5        | 972        | 2    |     |      |     |      | 1     | 5    | 15    |
| bird       | 26       | 2          | 892  | 30  | 13   | 8   | 17   | 5     | 4    | 3     |
| cat        | 12       | 4          | 32   | 826 | 24   | 48  | 30   | 12    | 5    | 7     |
| deer       | 5        | 1          | 28   | 24  | 898  | 13  | 14   | 14    | 2    | 1     |
| dog        | 7        | 2          | 28   | 111 | 18   | 801 | 13   | 17    |      | 3     |
| frog       | 5        |            | 16   | 27  | 3    | 4   | 943  | 1     | 1    |       |
| horse      | 9        | 1          | 14   | 13  | 22   | 17  | 3    | 915   | 2    | 4     |
| ship       | 37       | 10         | 4    | 4   |      | 1   | 2    | 1     | 931  | 10    |
| truck      | 20       | 39         | 3    | 3   |      |     | 2    | 1     | 9    | 923   |
|            | airplane | automobile | bird | cat | deer | dog | frog | horse | ship | truck |

Confusion matrix



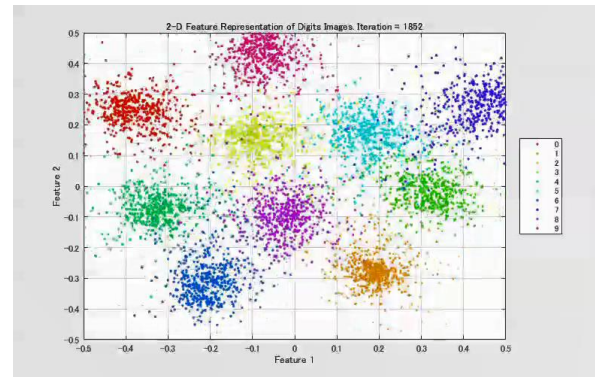
DeepDream Images



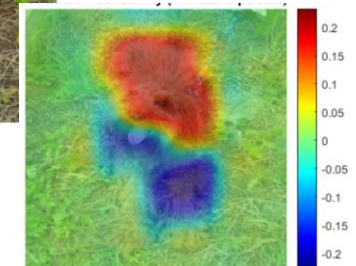
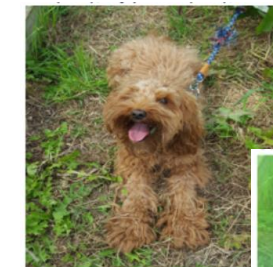
Grad-CAM R2019b



Layer Activations



tsneを使った特徴ベクトルの可視化



Occlusion sensitivity R2019b



# シンボリック表現を使ったグラフの描画

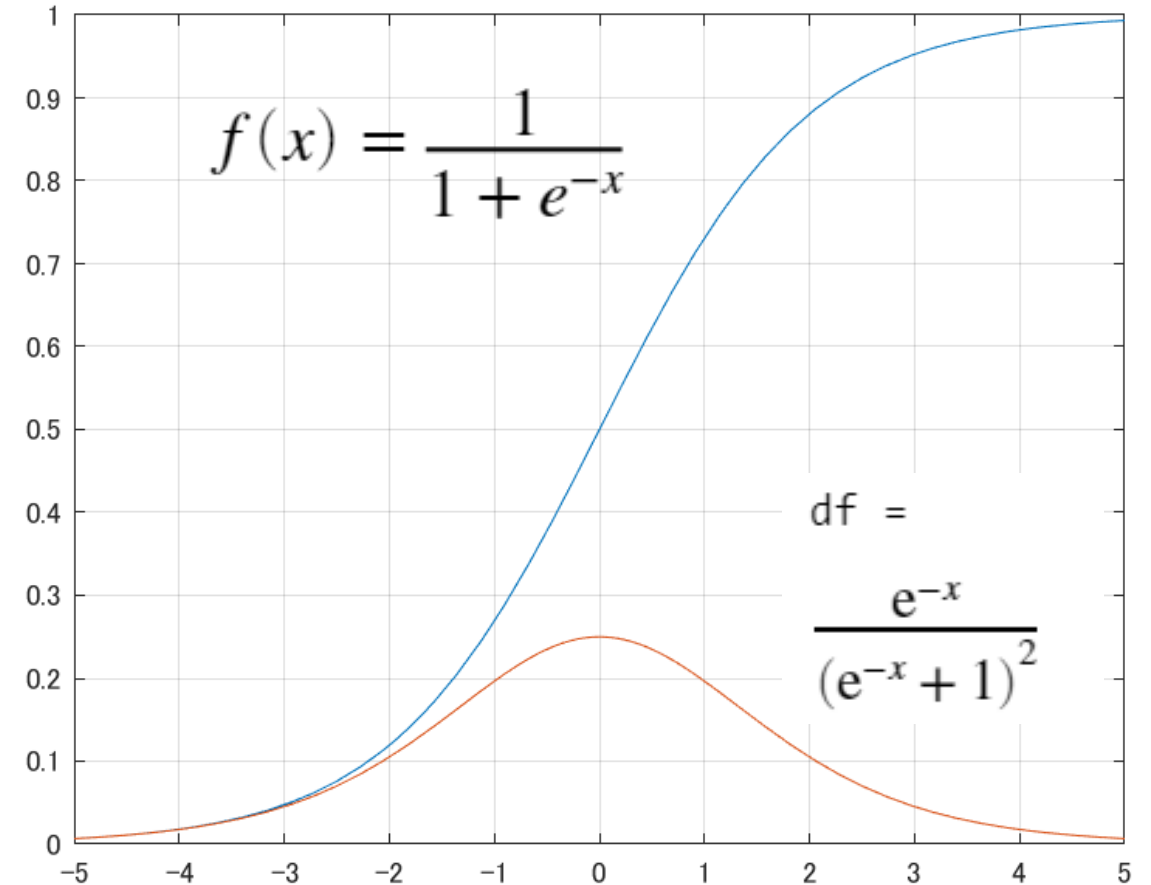
## 活性化関数のグラフ表示 微分の計算（手計算の検算）

シグモイド関数

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

％ シグモイド関数の表示

```
syms x;  
f = 1 / (1+exp(-x));  
df = diff(f)  
figure; hold on; grid on;  
fplot(f)  
fplot(df)
```

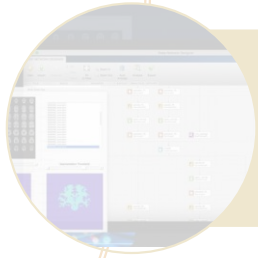




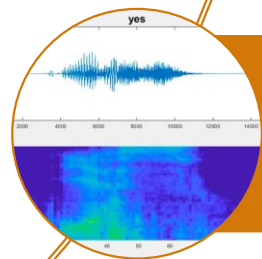
画像のディープラーニング



ニューラルネットワークとは？



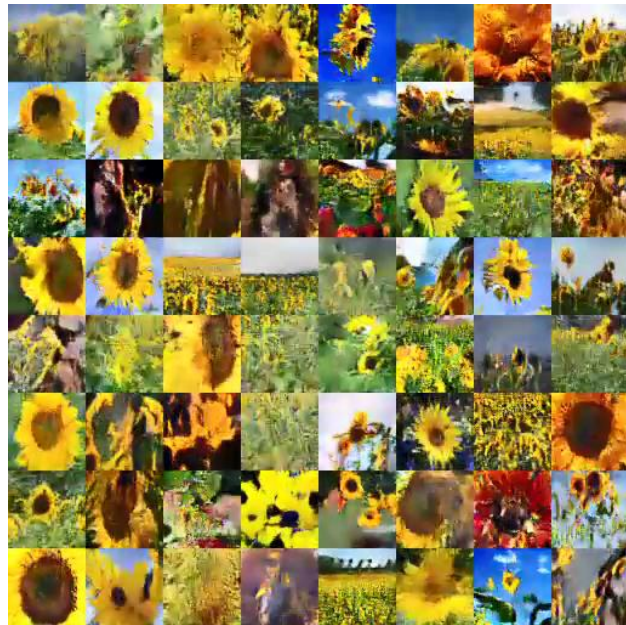
実験を高速で繰り返すために



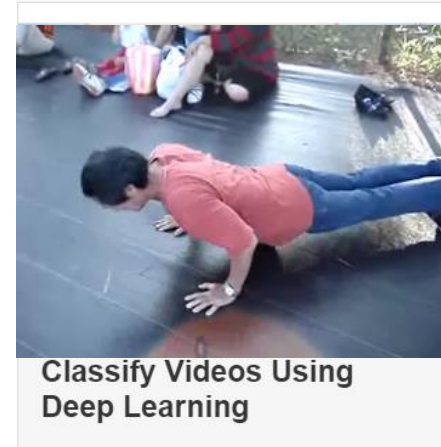
さらに学習したい人に

# 画像のディープラーニングをさらに学びたい方へ

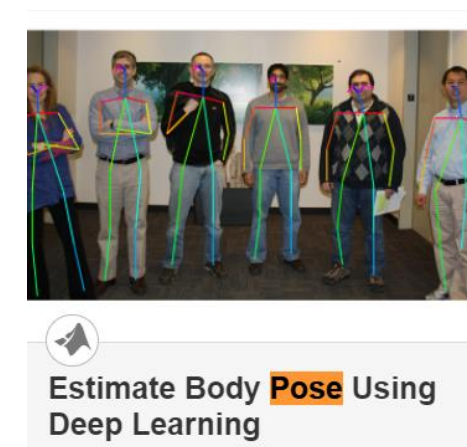
- GAN、VAEなど生成系のネットワーク



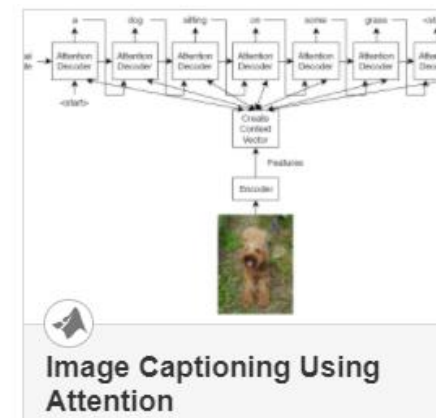
## 動画の認識



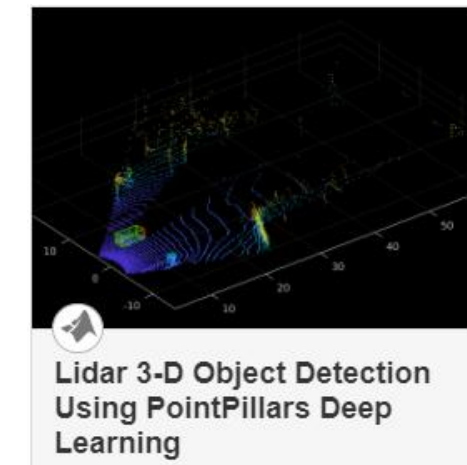
## 骨格検出



## キャプション生成



## Lidar系



[画像系の過去Webinar](#)

# 信号処理に適用するディープラーニング

- 近年の発展が著しい
- センサデータや音声データなど

直近のセミナー

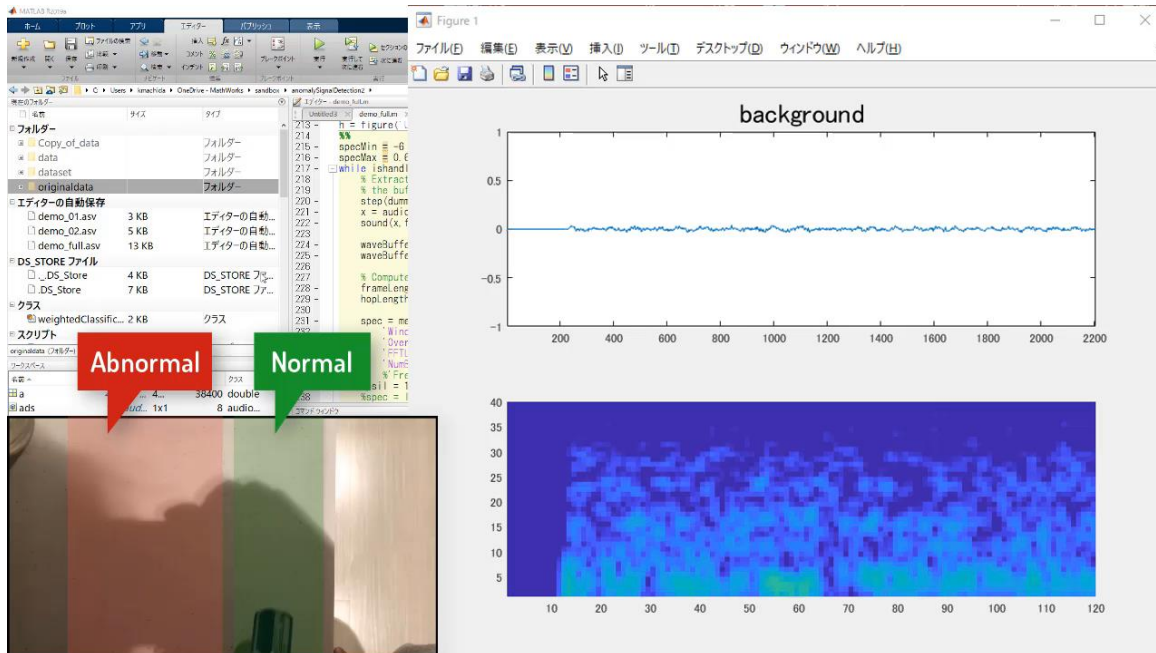
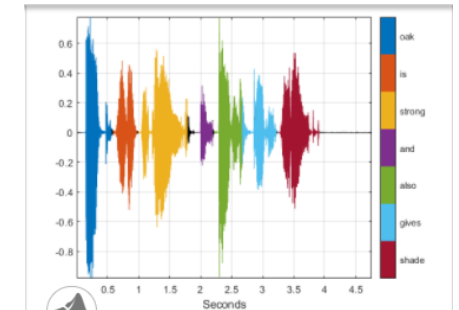
2020年11月27日 14:00~

「ディープラーニング・機械学習のための信号処理」

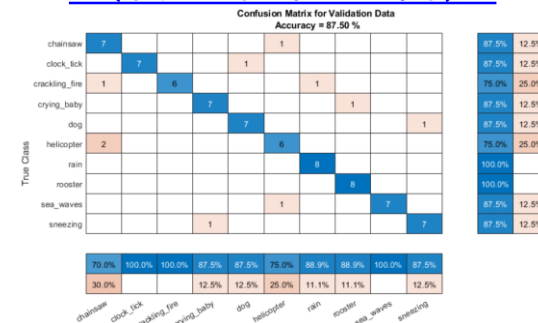
Webinar



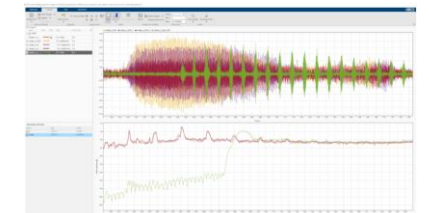
リファレンスアプリ



VGGish  
(音の転移学習)



信号アナライザー



サンプルコード : <https://jp.mathworks.com/matlabcentral/fileexchange/72430>



# テキスト解析に適用するディープラーニング

- センチメント（感情）分析
- テキストの分類
- テキストの生成

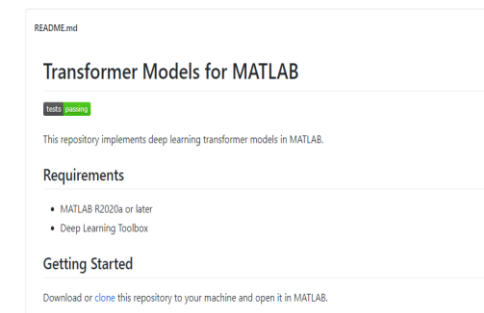


[Webinar](#)

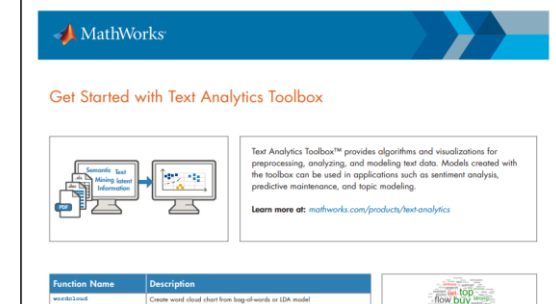
[リファレンスアプリ](#)



[GPT-2](#)

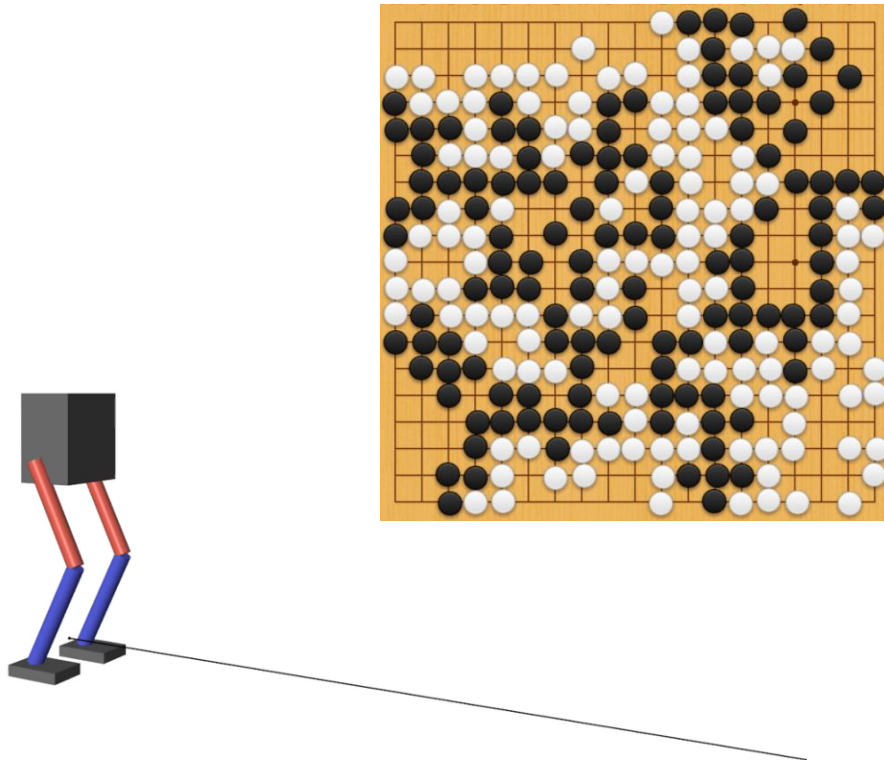


[チートシート](#)

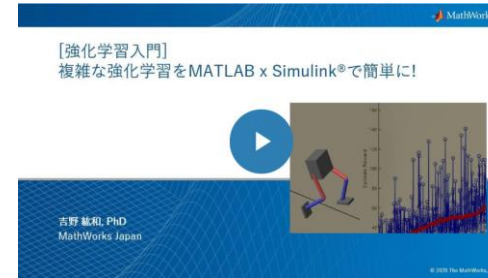


# 強化学習

- 報酬を最大化する一連の行動の仕方を学ぶ



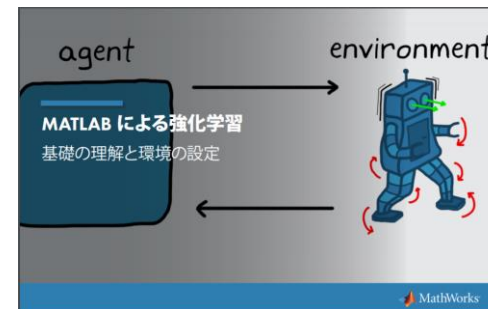
## Webinar



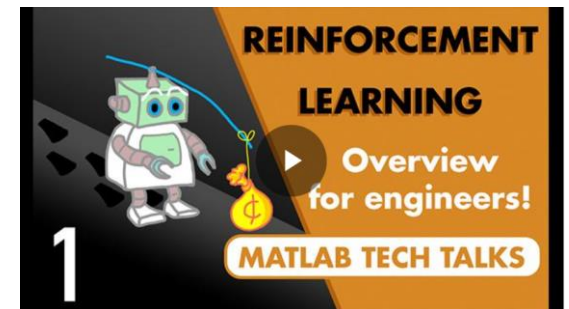
## リファレンスアプリ



## e-book



## ビデオ教材シリーズ



# オンライン無料学習

誰でも利用できる 11 時間分の  
**無料**コンテンツ

- **ブラウザ上**で動かせます。
- いつでも開始、終了、再開
- ステップバイステップ
- 勉強したいところだけでも大丈夫
- **ゲーム感覚**で取り組みやすい

**無料**

**MATLAB 入門 (日本語)**  
最短でMATLABの基礎を学びましょう。

[コースを開始](#) [コース詳細](#)

**無料**

**Simulink 入門 (日本語)**  
最短でSimulinkの基礎を学びましょう。本コースはSimulinkをインストールすると受講できます。

[詳細を確認する](#)

**無料**

**機械学習 入門(日本語)**  
分類問題のための実用的な機械学習手法の基礎を学びます。

[コースを開始](#) [コース詳細](#)

**無料**

**ディープラーニング入門 (日本語)**  
ディープラーニング手法を使用した画像認識を行う方法を学びましょう

[コースを開始](#) [コース詳細](#)

**NEW** **無料**

**画像処理入門 (英語)**  
MATLABで実用的な画像処理の基本を学びます。

[コースを開始](#) [コース詳細](#)

**NEW** **無料**

**信号処理入門 (英語)**  
スペクトル解析のための実践に即した信号処理方法を対話形式で説明します。

[コースを開始](#) [コース詳細](#)

**無料**

**Stateflow 入門 (日本語)**  
Stateflowでステートマシンを作成、編集、およびシミュレーションするための基礎を学びます。

[詳細を確認する](#)

**NEW** **無料**

**Simulink による制御設計入門 (英語)**  
Simulinkで基礎的なフィードバック制御系の設計方法を学びます。

[詳細を確認する](#)





# 豊富な学習用コンテンツ

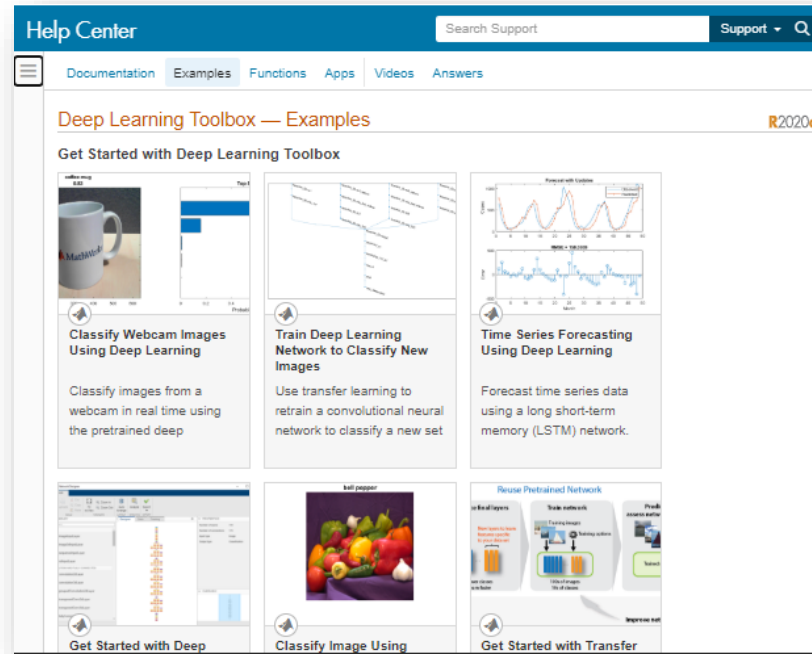
## ビデオ/ウェビナー集

計12時間以上の動画教材



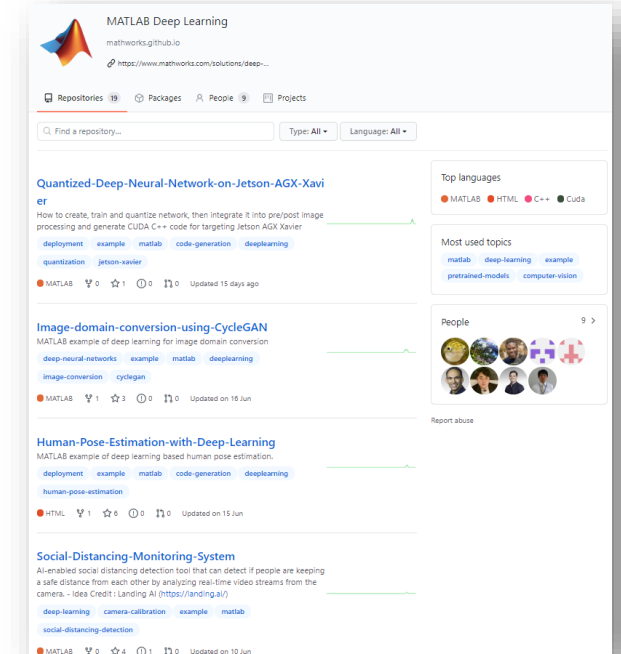
## ディープラーニングサンプルコード

プロトタイピングに最適



## GitHub リポジトリ MATLAB Deep Learning

最新のサンプルも！





## 本セミナーで伝えたかったこと

1. ディープラーニングの仕組み自体は、そんなに難しくない！

2. データを可視化して確認することが、理解への近道です！

3. 理解ができれば、ツール（ライブラリ）を活用してください！

データセットの作成と  
アクセス

前処理と変換

予測モデルの開発

実システムへの展開

学習過程の可視化・実験の高速化 → 理解の促進・研究の加速

## 授業開発についてご質問・ご要望ございましたら

- お問い合わせ先
- MathWorks Japan
- Education Customer Success Engineering 部
- Email: [cse-jp@groups.mathworks.com](mailto:cse-jp@groups.mathworks.com)



© 2020 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [www.mathworks.com/trademarks](http://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.