

# ディープラーニング・機械学習のための 信号処理

MathWorks

アプリケーションエンジニア

竹本 佳充

# このセミナーに参加される方は...

信号の扱いに慣れた  
信号処理エンジニアである必要はありません

機械学習や深層学習に詳しい  
データサイエンティストである必要はありません

# 概要

## Access Data



Sensors



Files



Databases

## Analyze Data



Data exploration



Preprocessing



Domain-specific algorithms

## Develop



Model



Algorithm development



Modeling & simulation

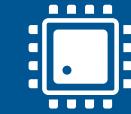
## Deploy



Desktop apps



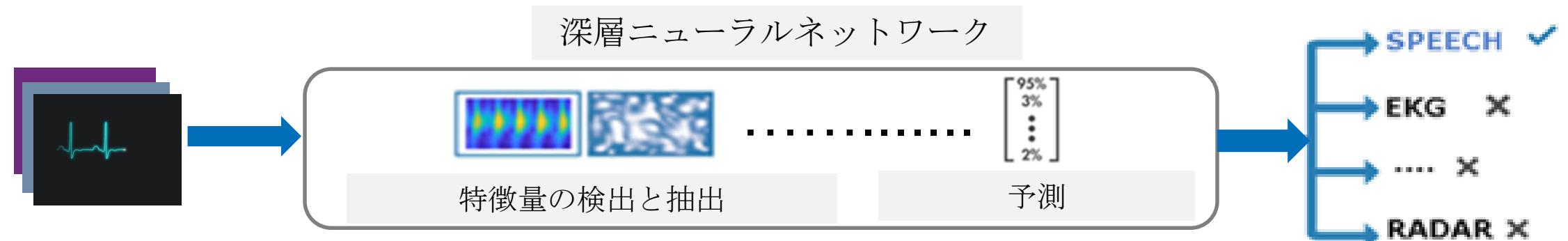
Enterprise systems



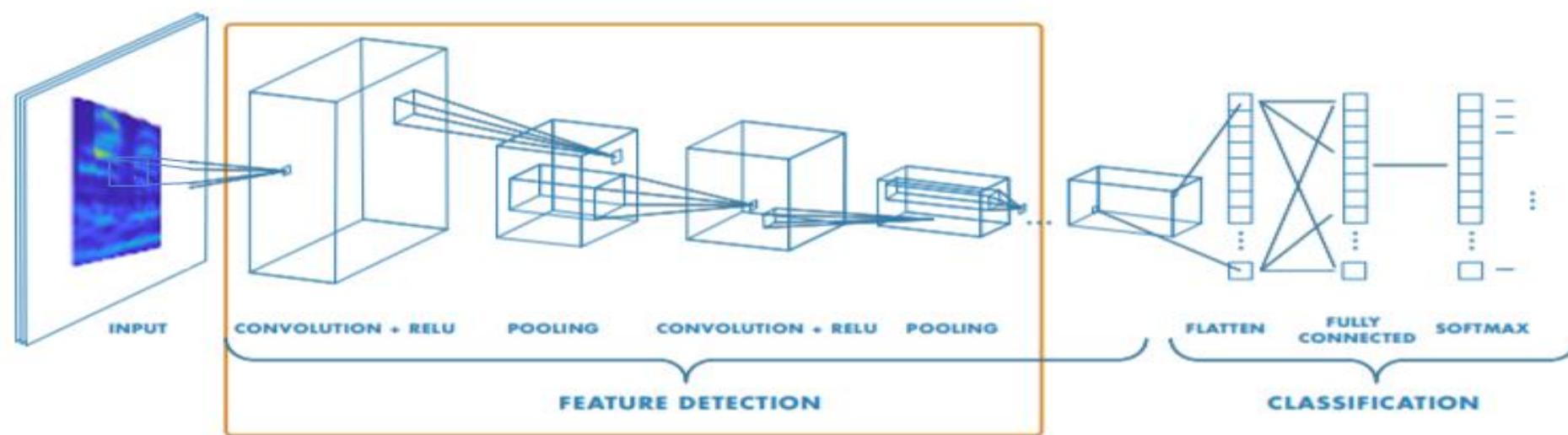
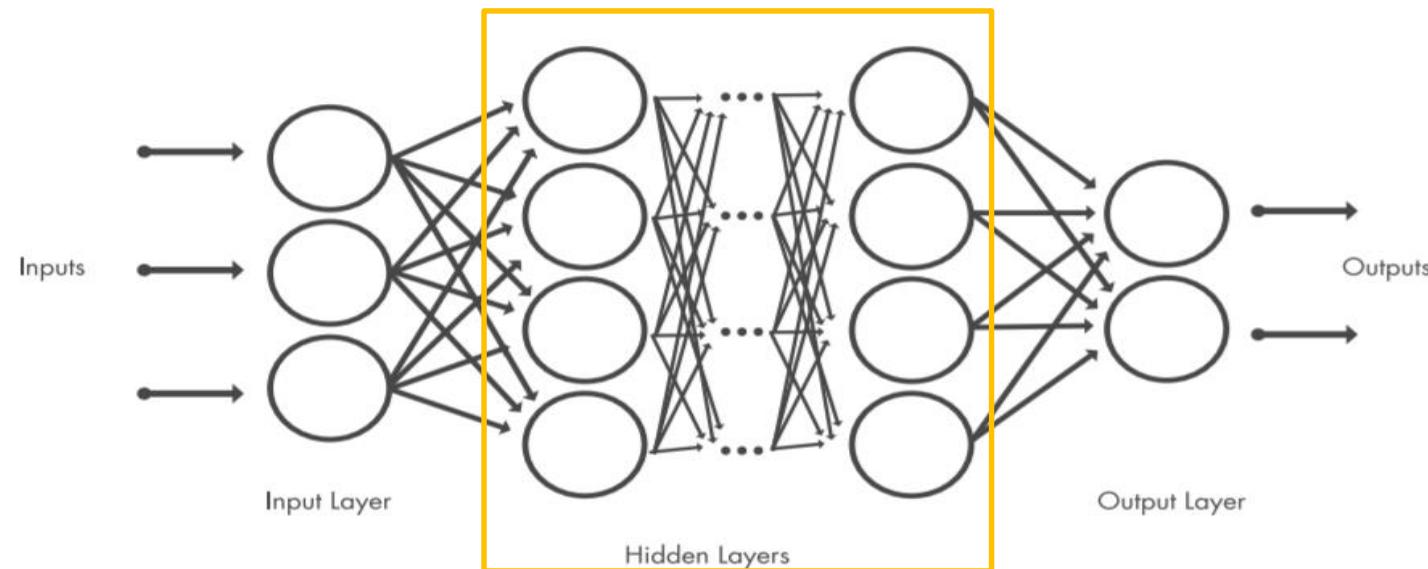
Embedded devices

# ディープラーニングとは？

## ディープラーニングの概要

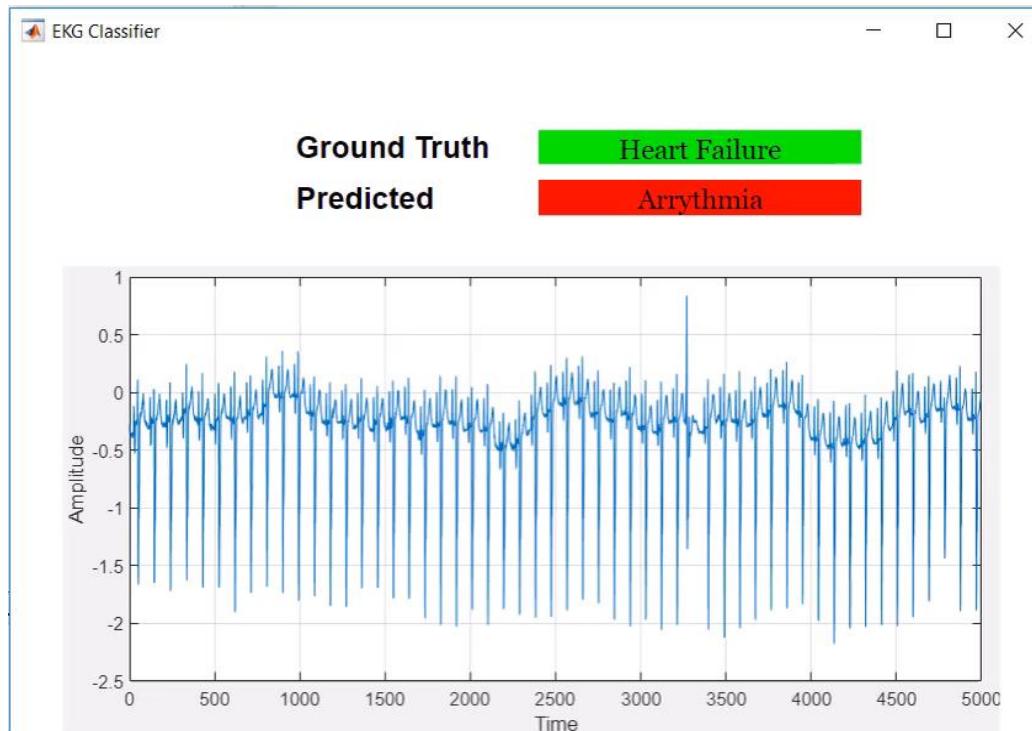


# ディープラーニングで行われている処理



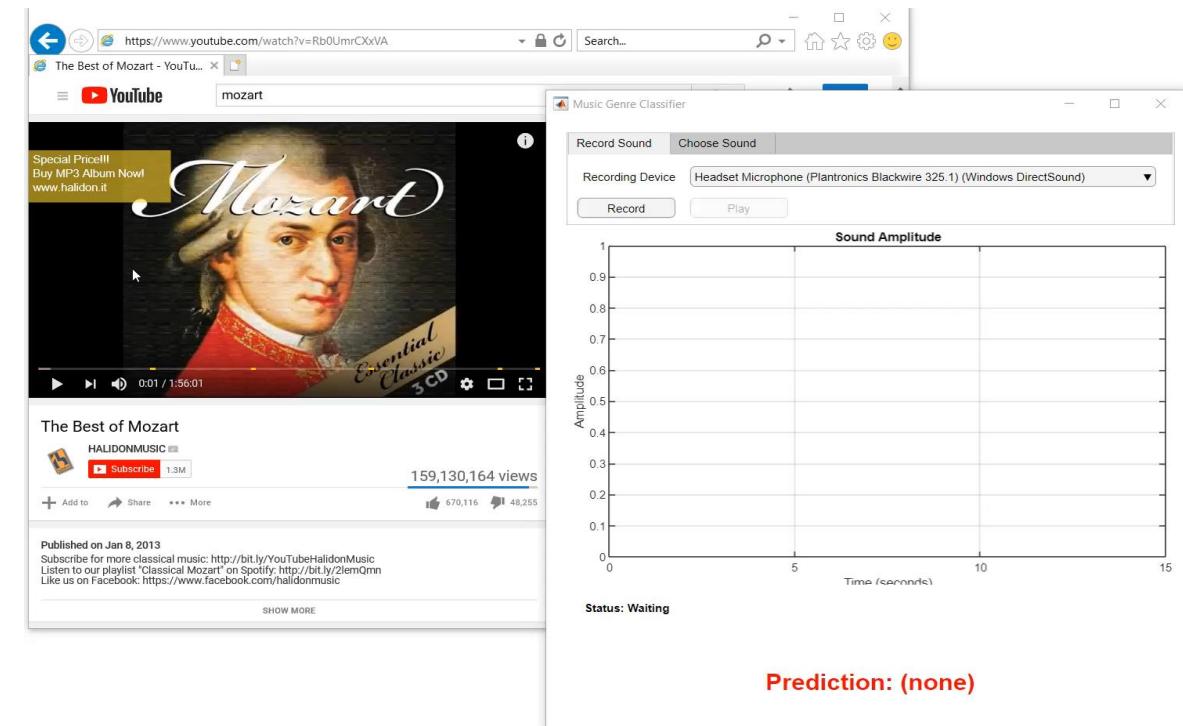
# ご紹介する二つのデモ...

## EKG 分類



AlexNetを使用した転移学習

## 音楽ジャンル認識



ウェーブレットスキャッタリングによる  
自動特徴量抽出

# 信号分類のアプローチ

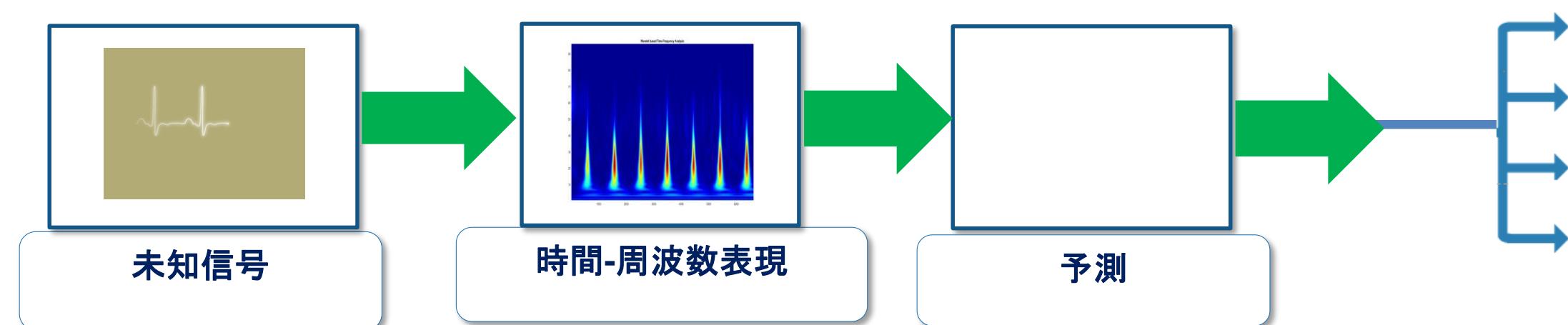
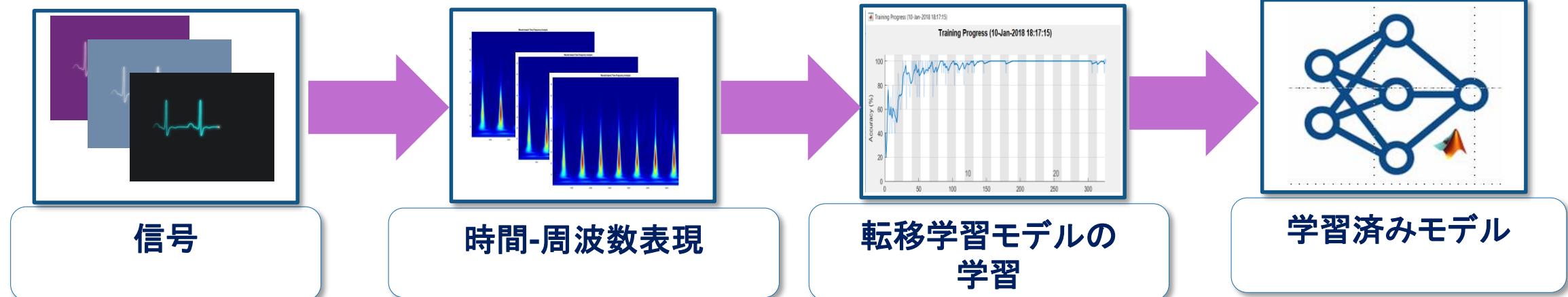
- 転移学習による信号分類
- ウェーブレットスキャッタリングによる自動特徴量抽出
- LSTMネットワークによる分類

# 信号分類のアプローチ

- 転移学習による信号分類
- ウェーブレットスキャッタリングによる自動特徴量抽出
- LSTMネットワークによる分類

# ワークフロー

## – 時系列信号に対する転移学習



# 転移学習の利点

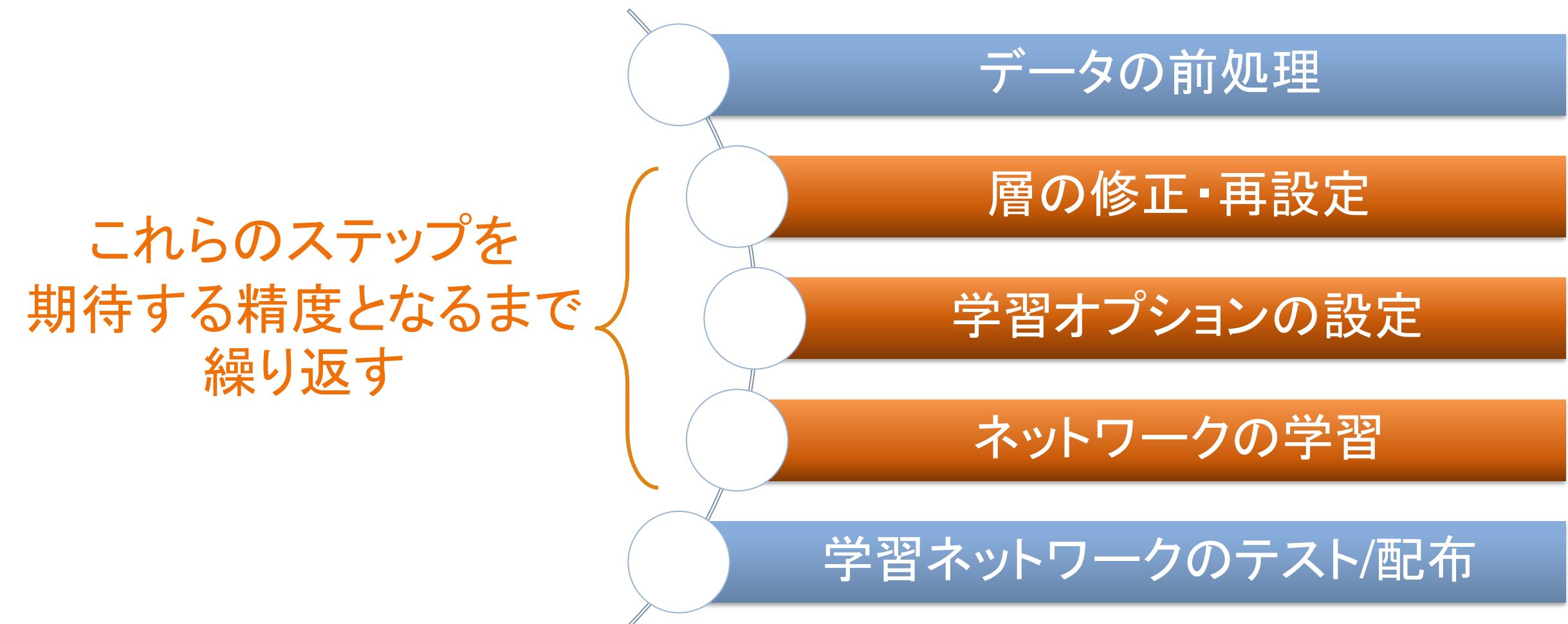
- 実績のあるモデルは特徴抽出に優れている
  - 初期レイヤでは、エッジの様なローレベル特徴量を学習する等
- 既存のネットワークに対して最小限の修正で実現可能
  - 目的のデータのクラス数に応じて最終層を置き換え
- 最初のとっかかりとして
  - 何はともあれ転移学習

**AlexNet**  
PRETRAINED  
MODEL

**VGG-16**  
PRETRAINED  
MODEL

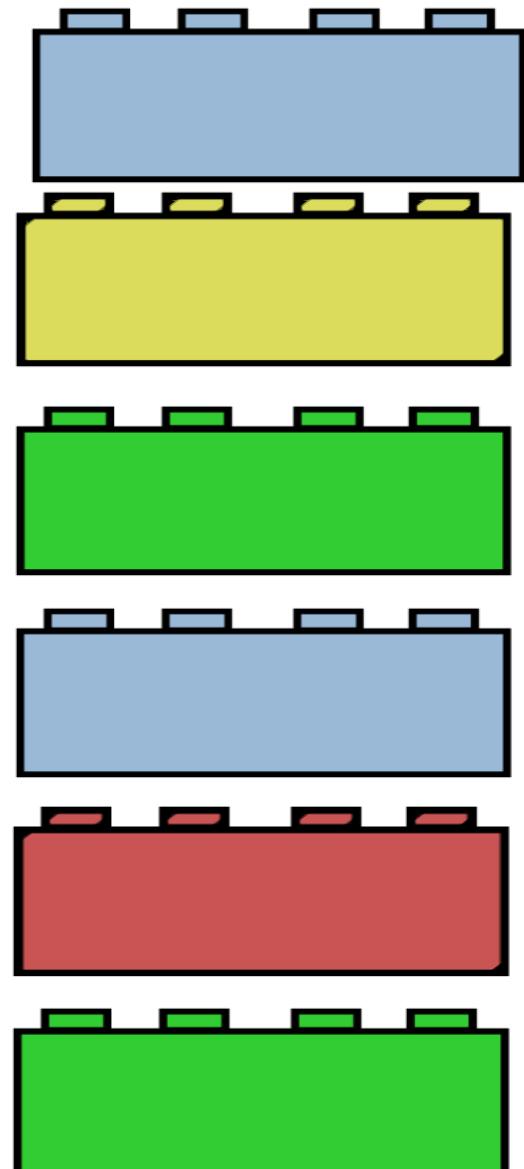
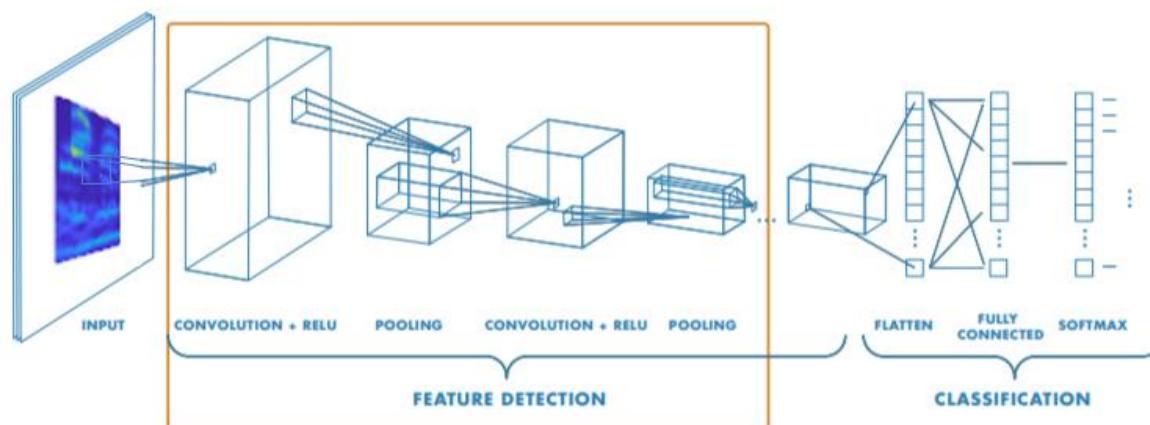
**GoogLeNet**  
PRETRAINED MODEL

# 転移学習ワークフローの流れ



# 層について考える

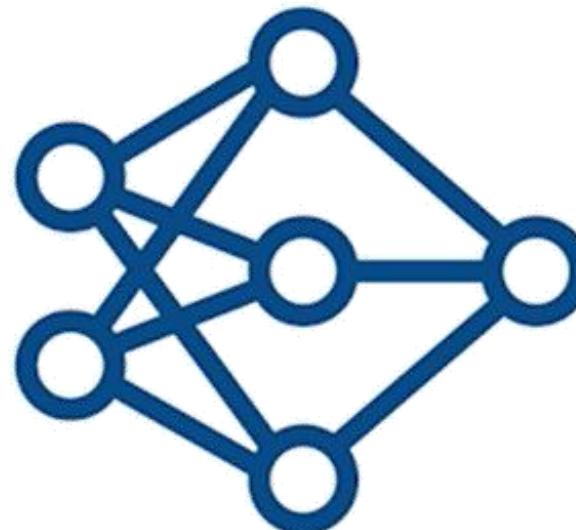
- 層はレゴブロックの様なもの
  - 互いに積み上げていく
  - 他のブロックと容易に入れ替えられる
- 各隠れ層は、前段の層からの情報を処理するための特有の機能を持つ



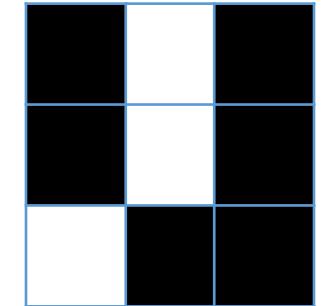
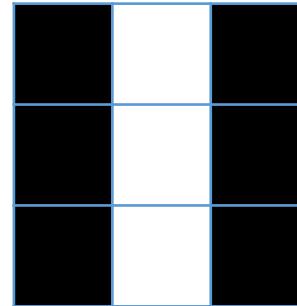
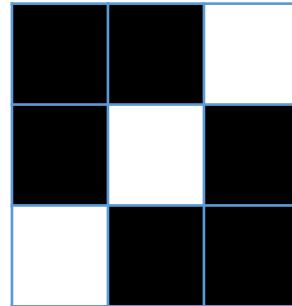
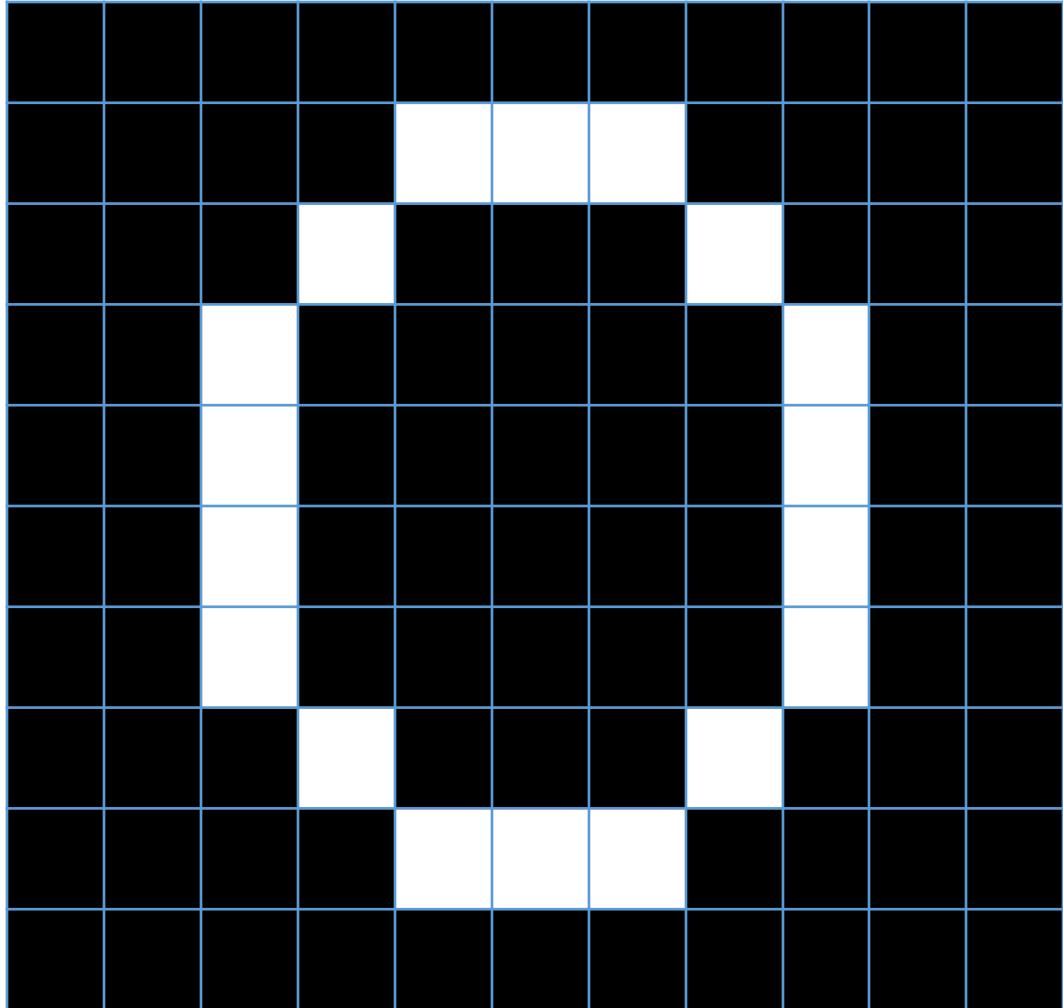
# 畳み込みニューラルネットワーク (CNNs)

- 層の組み合わせを工夫することで最良の分類器となる

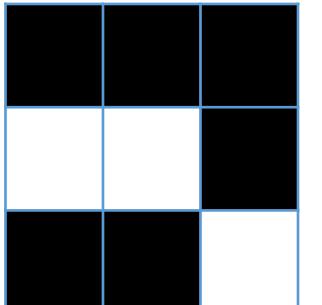
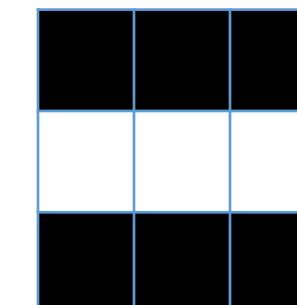
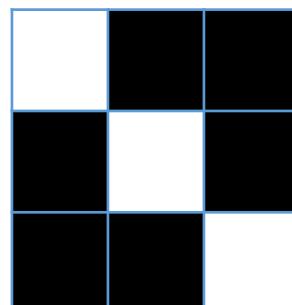
- 畳み込み層
- マックスプーリング層
- ReLU層



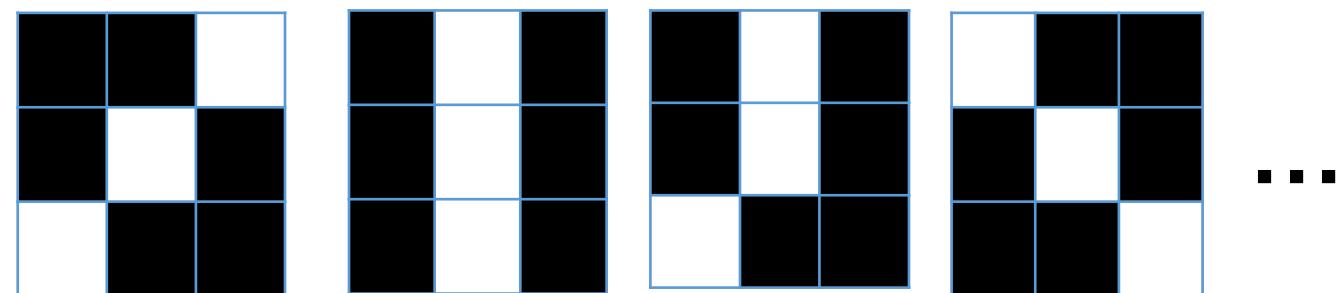
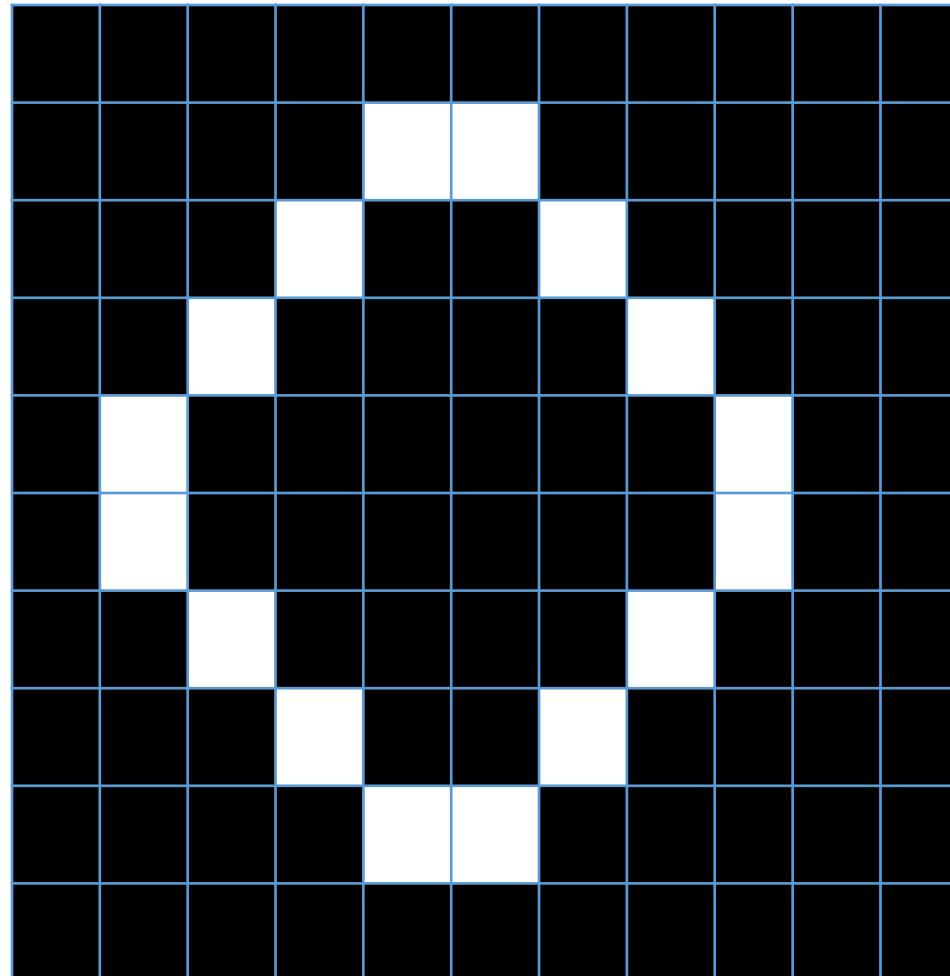
# 畳み込み層はパターンを探す



数字「0」では、これらのパターンが共通

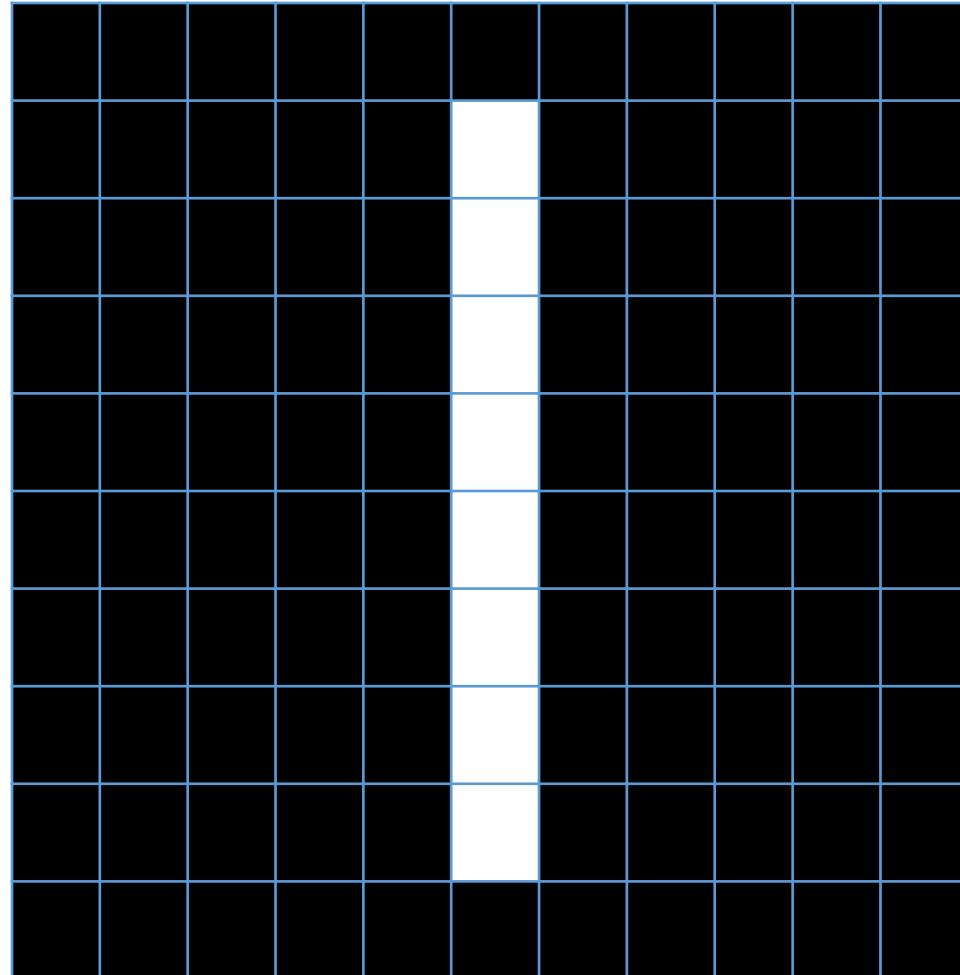


# 対象画像に対して全パターンが比較される

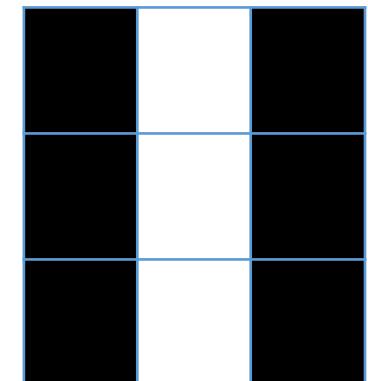


- パターンは左端からスタート  
比較処理  
ピクセルごとにスライド  
→画像の最後に到達  
→次のパターンで繰り返す

# 畳み込みにおいてパターンがうまく合えば オブジェクトを正しく分類できるチャンス

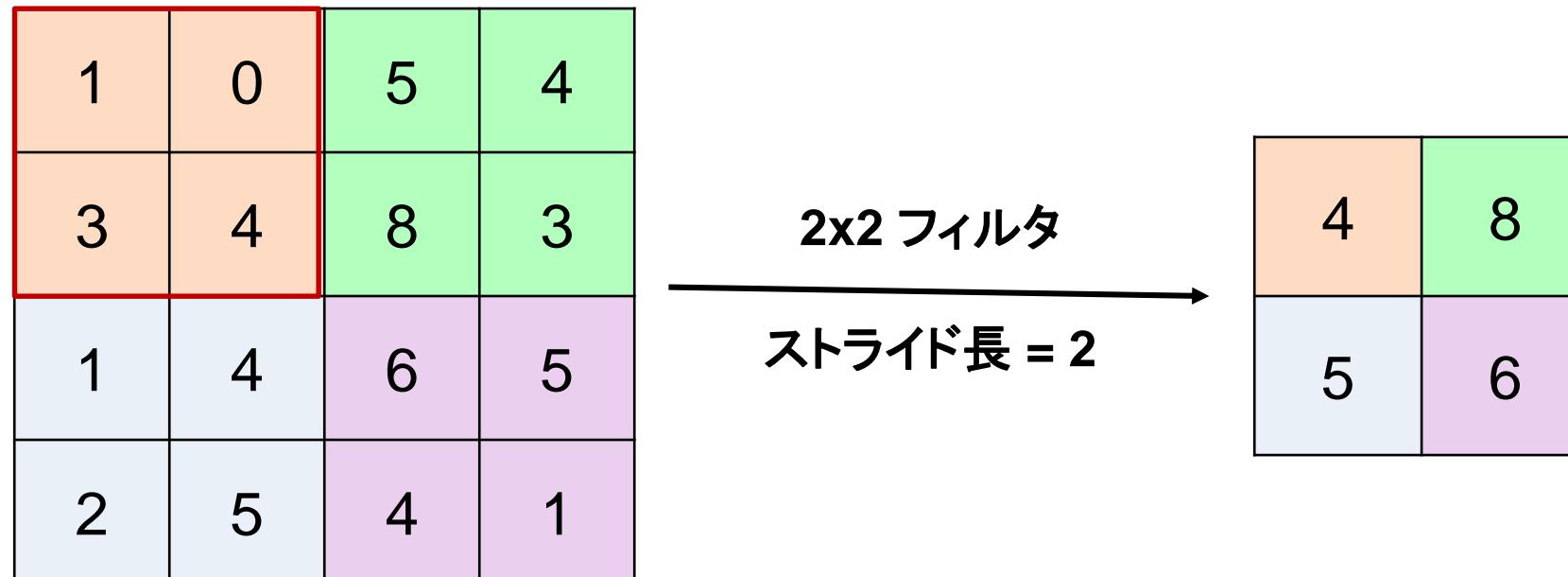


- この画像は数字0のパターンとは  
うまく合わないと思われる
- この画像は、このパターンにのみ  
うまく合う



# マックスプーリングは間引き処理

重要な情報を保持しながら次元を削減



# 正規化線形ユニット層 (ReLU)

負の数値をゼロに変換

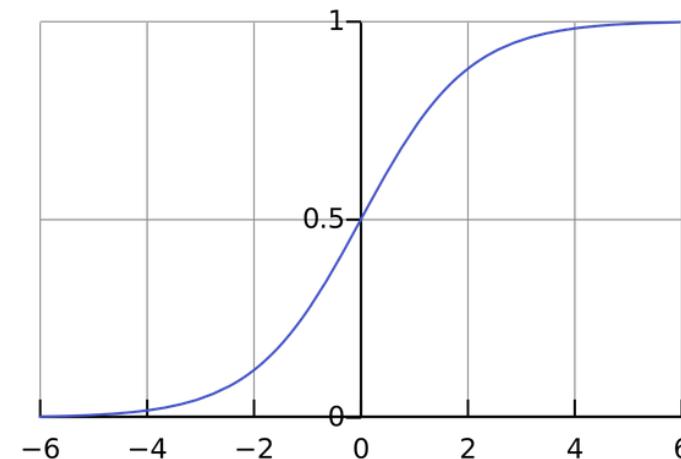
-1	0	5	4
3	-4	-8	3
1	4	6	-5
-2	-5	4	1



0	0	5	4
3	0	0	3
1	4	6	0
0	0	4	1

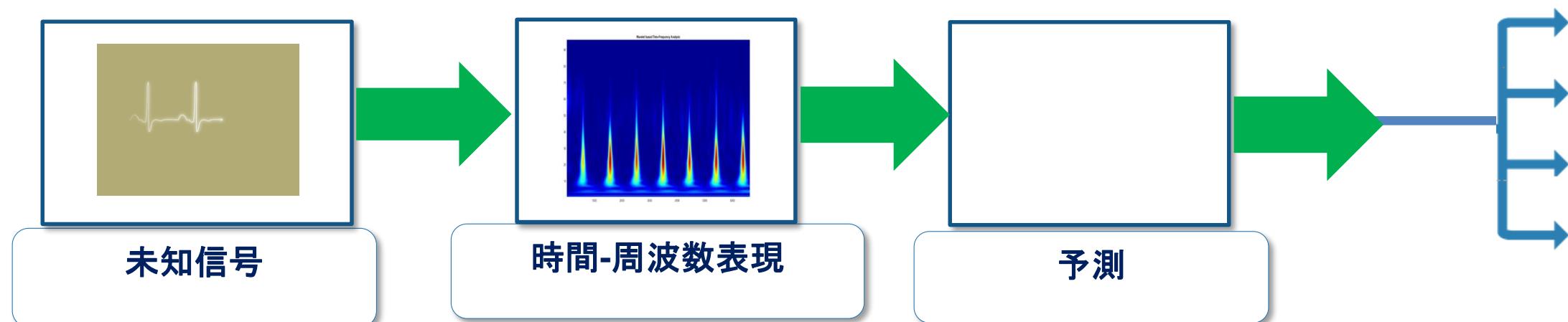
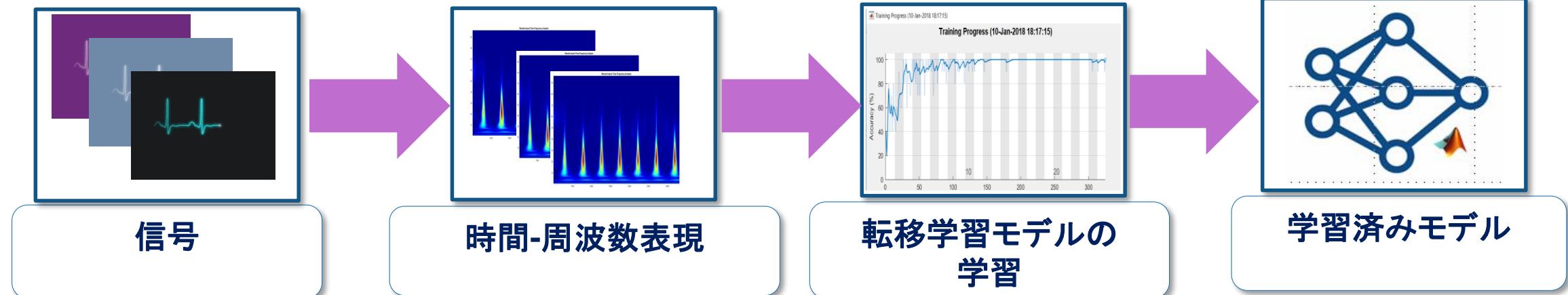
# CNNは3つの層で終端

- **全結合層** : Fully Connected Layer
  - 特定のカテゴリに対応して特徴量が高くなる位置を探索
  - 各カテゴリに対してスコアを計算 (最高スコアのものが採用される)
- **ソフトマックス層** : Softmax Layer
  - スコア値を確率値に変換
  - 総和は1
- **分類層** : Classification Layer
  - ネットワークが学習したクラスのうちの一つに画像を分類



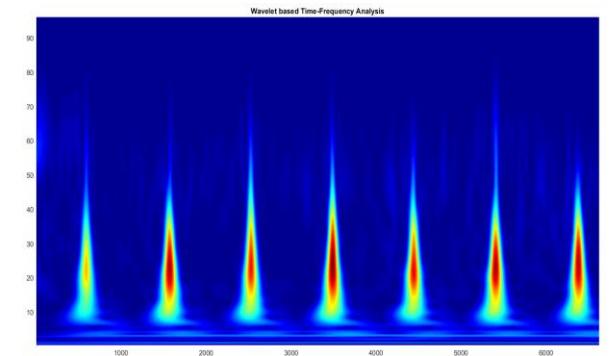
# ワークフロー(再掲)

## – 時系列信号に対する転移学習



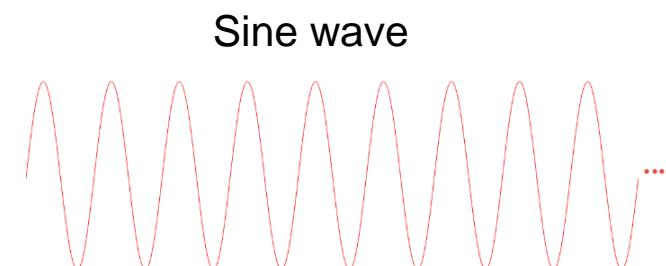
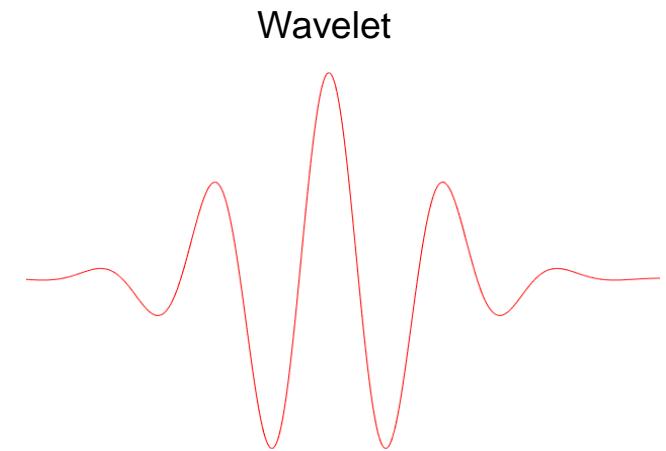
# 時間周波数表現に信号を変換

- 時間周波数表現は、周波数成分がどの様に変化するのかを時間軸で観測
  - パターンは画像として保存できる
- 提供される例題:
  - スペクトログラム、メル周波数スペクトログラム
  - 定Q変換
  - スカラグラム (連続ウェーブレット変換)、 (良好な時間周波数分解能、設定パラメータが少ない)
- 再掲: 置み込み層はパターンを探す
  - シャープな時間周波数表現は、迅速なモデルの学習に役立つ
  - シャープな時間周波数表現は、非常に似ているように見えて、実際には異なるクラスに属する信号内の微妙な情報を強調することができる



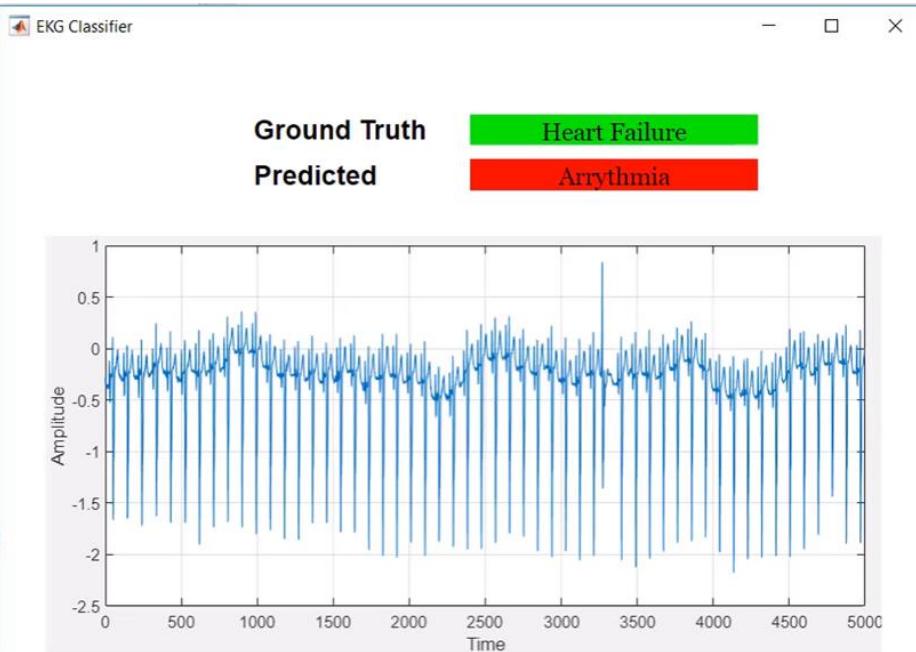
# ウェーブレットとは？

- ウェーブレットは、平均0で、急速に減衰する振動の様な波
- ウェーブレットは、実世界の信号の局所的な周波数成分に適応させるのに適している
- MATLAB はあらかじめウェーブレットの初期設定が用意されているため、容易に試すことができる



# 例題 1: EKG分類

- 目的:  
ラベル付きデータを用いて迅速に分類器を作成する
- データセット: 各65Kサンプルの30~個のデータ
  - 正常 (Class I)
  - 心房細動 (Class II)
  - 心不全 (Class III)
- アプローチ: 学習済みモデル
  - AlexNet
- 考慮しないこと: CNNの構造、パラメータチューニング



# Let's try it out!

*Exercise: DeepLearningForSignals.mlx*

# 信号分類のアプローチ

- 転移学習による信号分類
- ウェーブレットスキヤッタリングによる自動特徴量抽出
- LSTMネットワークによる分類

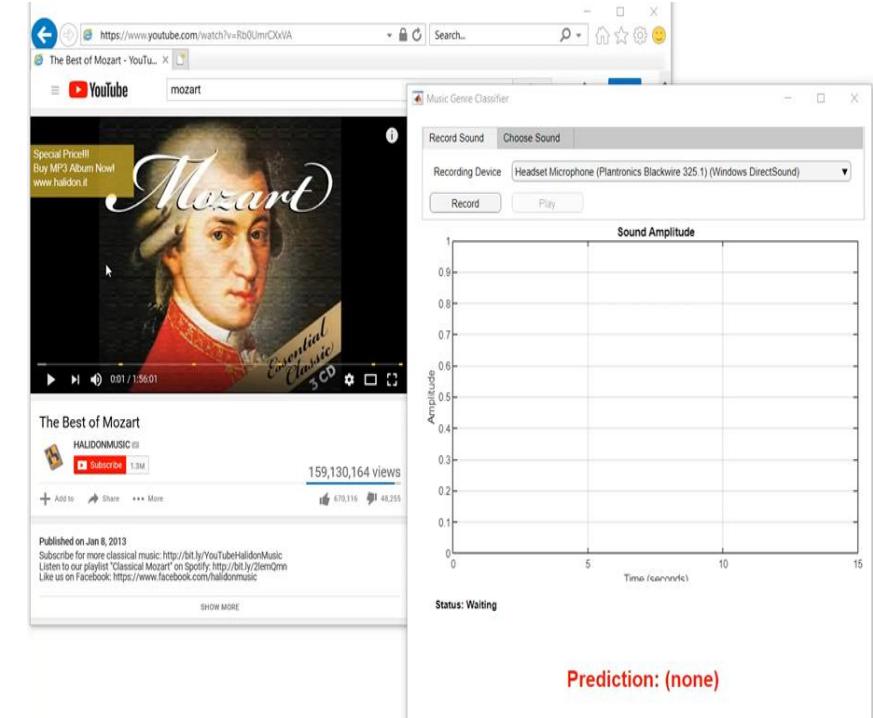
# 例題 2: ウェーブレットスキャッタリングによる音楽ジャンルの認識

- データセット: GTZAN Genre Classification<sup>[1]</sup>

- 手法: ウェーブレットスキャッタリングによる自動特徴量抽出

- 主な利点:

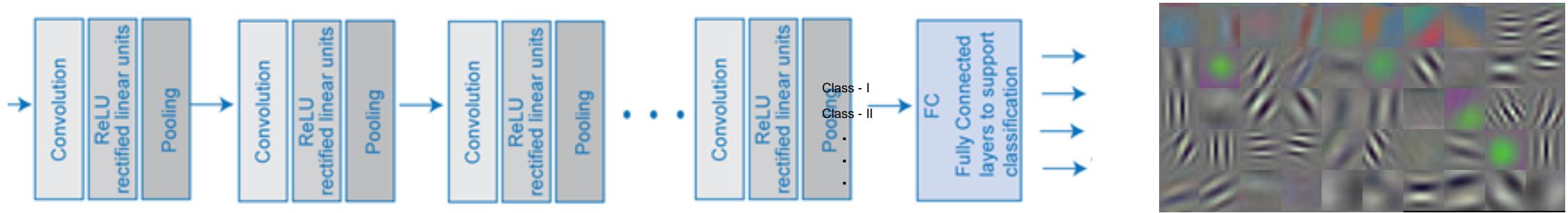
- 手法の試行錯誤が不要 (ハイパーパラメータチューニング等)
  - 特徴抽出処理を自動化 → 2行で実現



[1] Tzanetakis, G. and Cook, P. 2002. Music genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, Vol. 10, No. 5, pp. 293-302.

# 背景

- うまく学習されたCNNの処理はウェーブレットによるフィルタリング処理と似ている

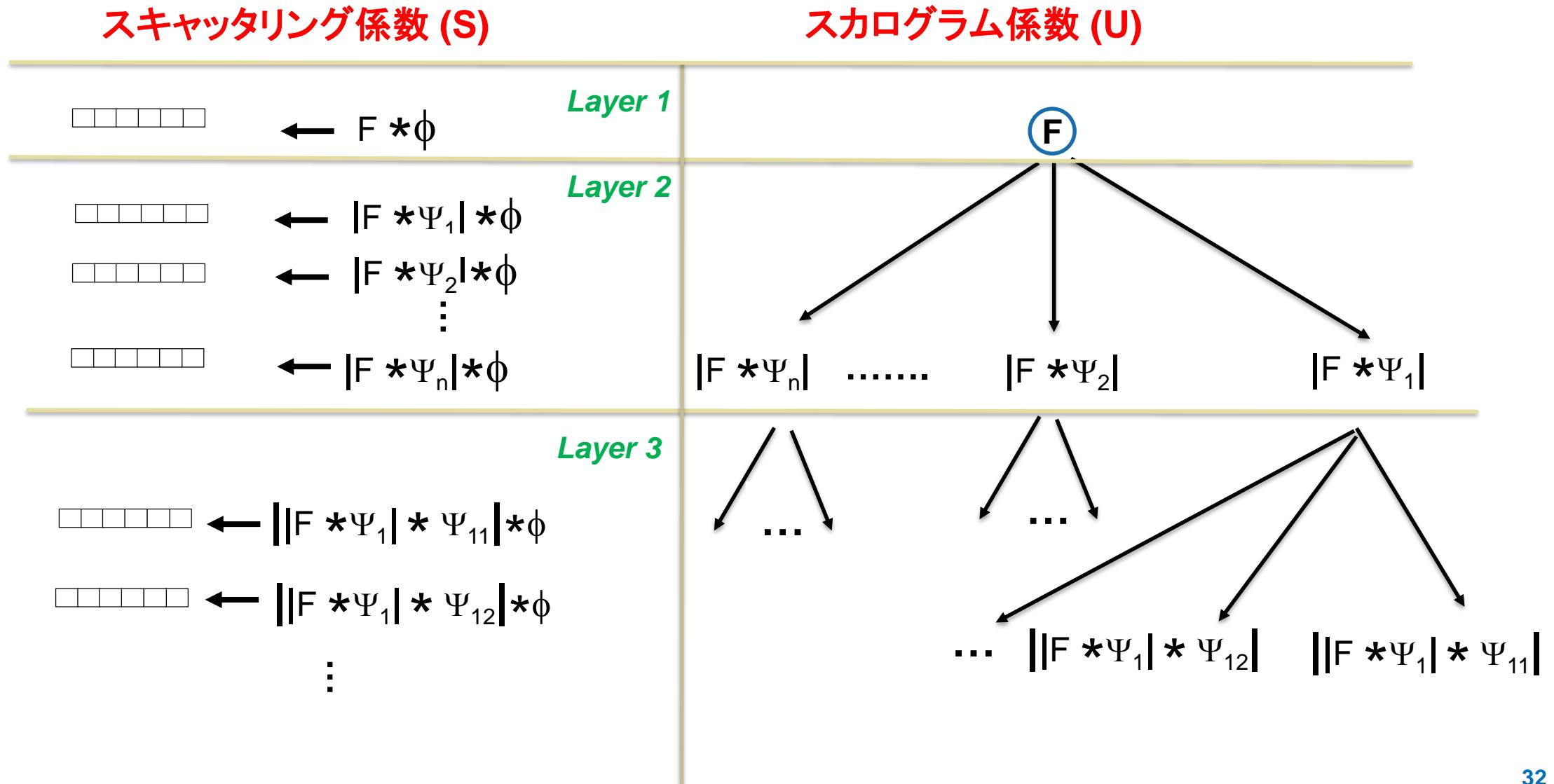


- ウェーブレットスキャッタリングフレームワーク [1]
  - 自動特徴抽出
  - 大量の学習データが用意できない場合の最初のトライとして最適
  - データの次元削減とコンパクトな特徴量

[1] Joan Bruna, and Stephane Mallat, P. 2013. Invariant Scattering Convolution Networks. [IEEE Transactions on Pattern Analysis and Machine Intelligence](#), Vol. 35, No. 8, pp. 1872-1886.

## 内部処理: ウェーブレットスキャッタリングフレームワーク

$\phi$  → スケーリングフィルタ  
 $\Psi$  → ウエーブレットフィルタ



# スキヤッタリングフレームワーク(続き)

- Q: 深層ネットワークとは?

A: 以下を行うネットワーク:

畳み込み

非線形処理

アベレージング

- 比較:

手法	ウェーブレットスキヤッタリング フレームワーク	畳み込みニューラルネットワーク
出力	各層の出力を使用	最終層の出力を使用
フィルタの重み	フィルタの重みは固定	フィルタの重みは都度学習
目的	特徴抽出	分類・回帰

# ウェーブレットスキャッタリングの流れ



# Let's try it out!

# 信号分類のアプローチ

- 転移学習による信号分類
- ウェーブレットスキャッタリングによる自動特徴量抽出
- LSTMネットワークによる分類

# LSTMによるディープラーニング例

- ディープラーニングによるシーケンス分類

この例では、長期短期記憶(LSTM)ネットワークを用いてシーケンスデータを分類する方法を示します。

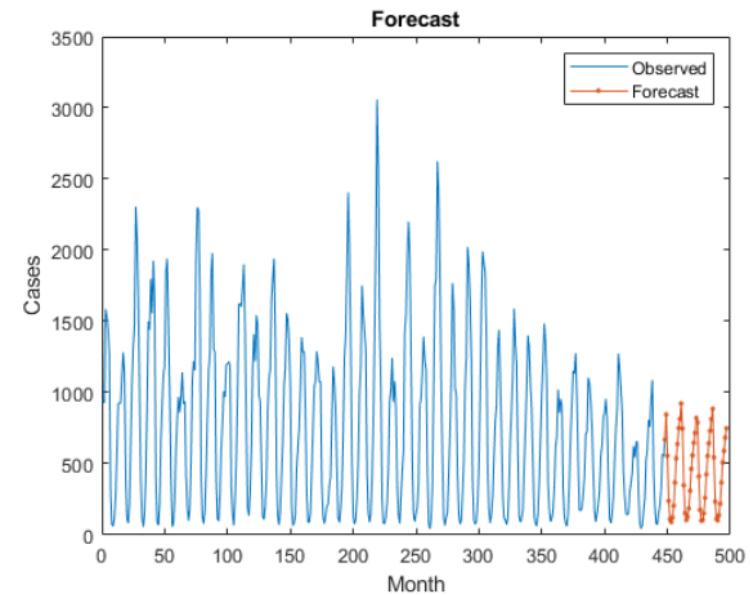
- ディープラーニングによるシーケンスからシーケンスへの分類

本実施例では、シーケンスデータの各時間ステップを長短記憶(LSTM)ネットワークを用いて分類する方法を示します。

- ディープラーニングによるシーケンスからシーケンスへの回帰

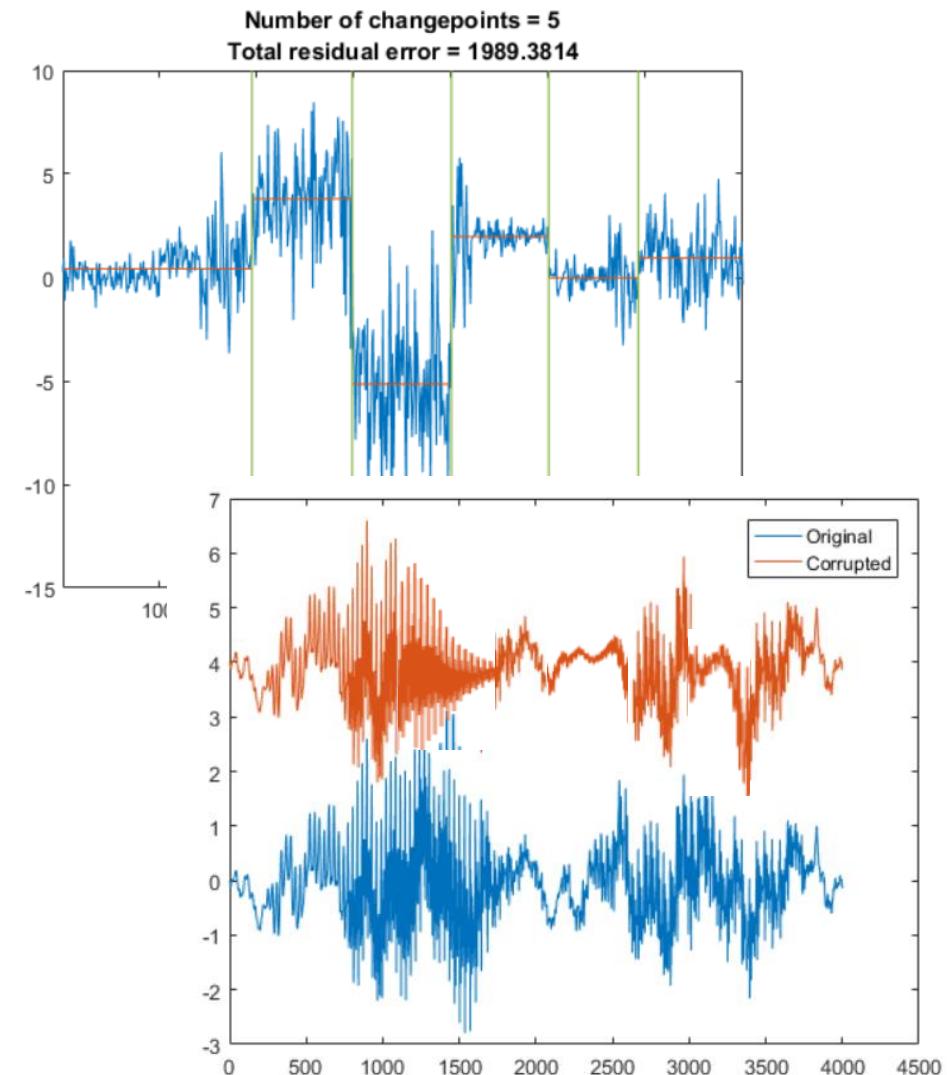
この例では、ディープラーニングを用いてエンジンの残存耐用年数(RUL)を予測する方法を示しています。

- *...and many more*



# 信号の前処理/ 特徴抽出

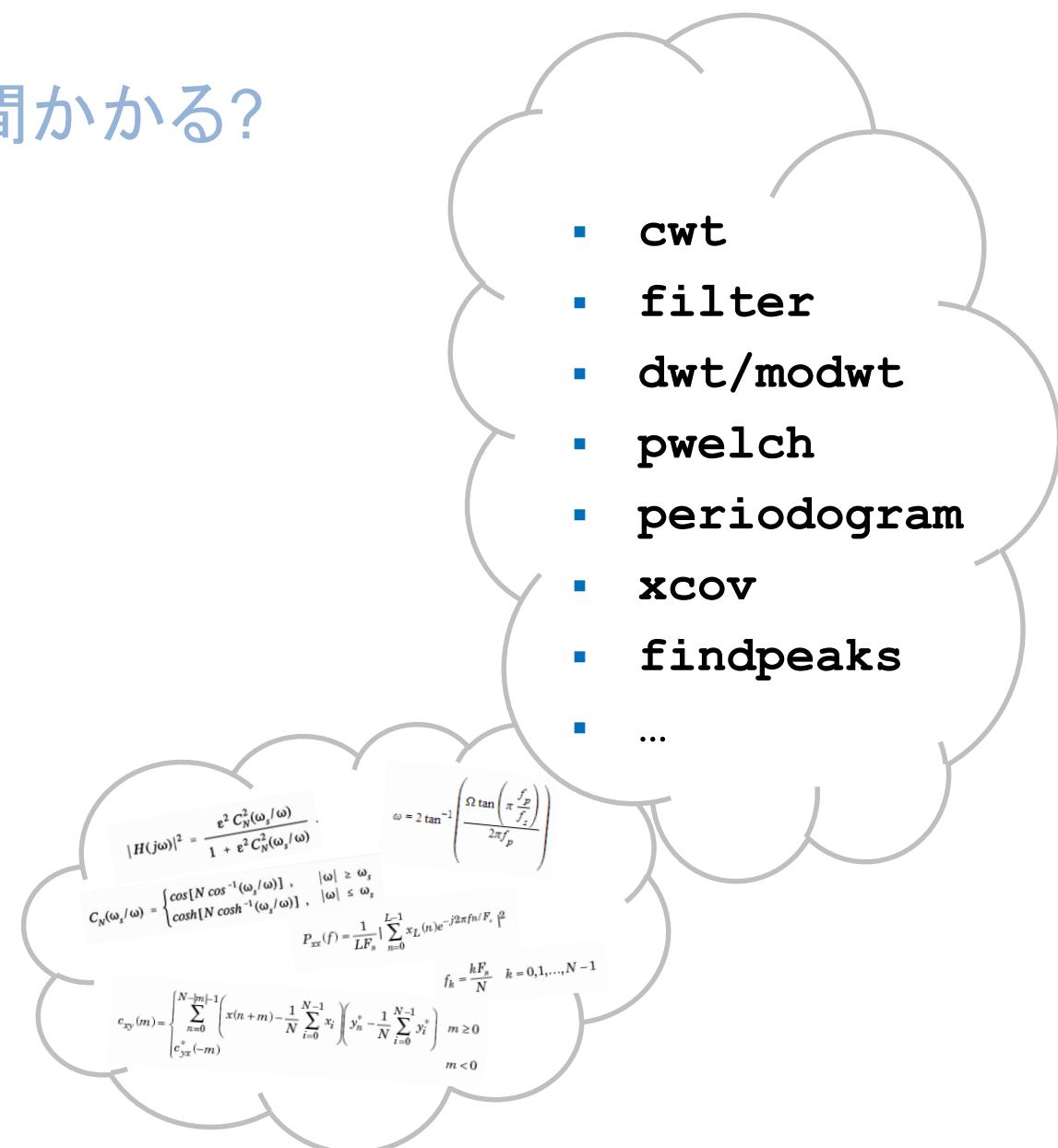
- 信号の前処理
  - ウェーブレット信号デノイズ
- 変化点抽出
- ダイナミックタイムワーピングによる信号の類似性比較
- 欠損サンプルの復元
- ... <and many more> ..



# ビルトインアルゴリズムの活用

## 時前でプログラミングすると何時間かかる?

- **Signal Processing Toolbox**
- **Wavelet Toolbox**
- Deep Learning Toolbox
- Statistics and Machine Learning Toolbox
- Parallel Computing Toolbox

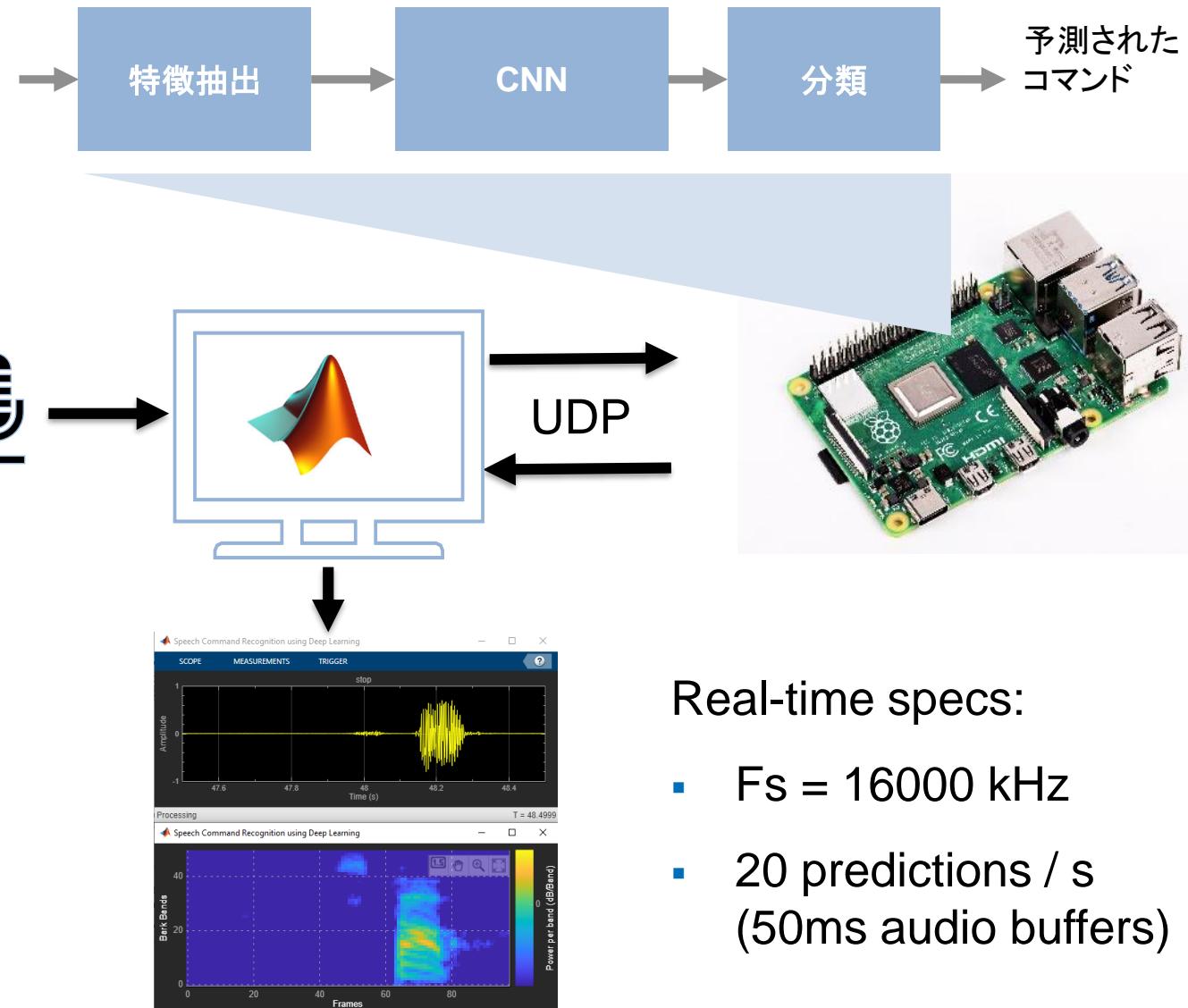


# 補足: Raspberry Piによるスピーチコマンド認識

R2020b

Example workflow sections:

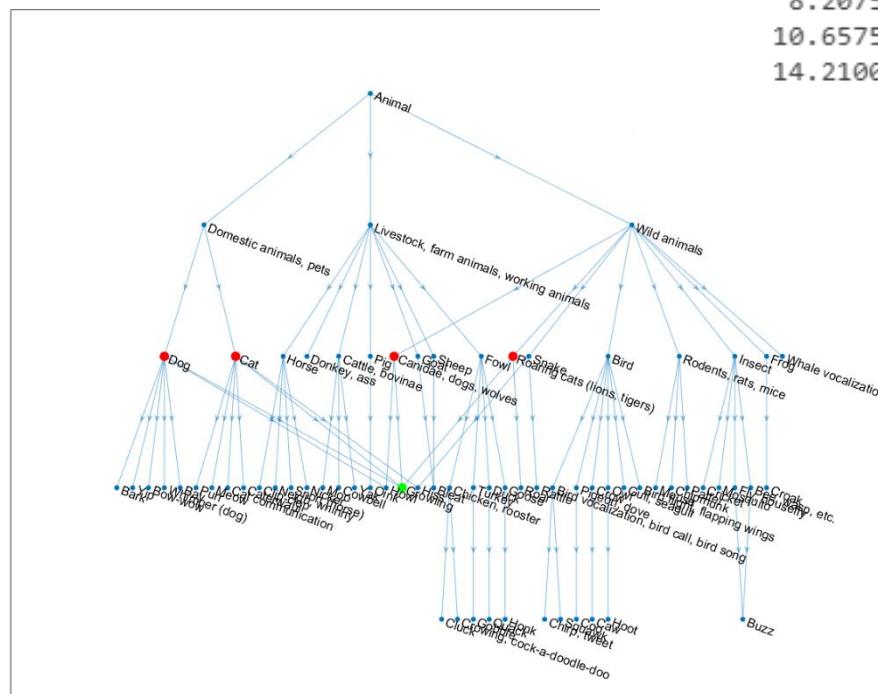
- MATLAB環境でのリアルタイムシミュレーション  
([existing example](#))
- 実装のためのコード修正
- MATLABからRaspberry Pi の実行ファイルを生成
- Raspberry Pi上でリアルタイムに実行しながら、UDP経由で結果を転送し、MATLAB で可視化
- PILによるプロファイリング



Real-time specs:

- $F_s = 16000$  kHz
- 20 predictions / s  
(50ms audio buffers)

- サウンド分類:
    - 521 の異なる類義種別
    - ワードクラウド
  - 学習済みネットワーク
    - yamnet()
    - vggish()



## Classify sounds in signal

```
[sounds, timeStamps] = classifySound(audioIn, fs)
```

```
sounds = 1x4 string
    "Music"      "Speech"      "Music"      "Explosion"
```

```
timeStamps = 4x2
            0    7.2275
        8.2075    9.5550
    10.6575   13.4750
    14.2100   15.0675
```

Busy signal  
Alarm  
Sine wave  
Telephone

Chirp tone

Dial tone

# Thank You