

センサーデータ解析のためのニューラルネットワーク

MathWorks Japan

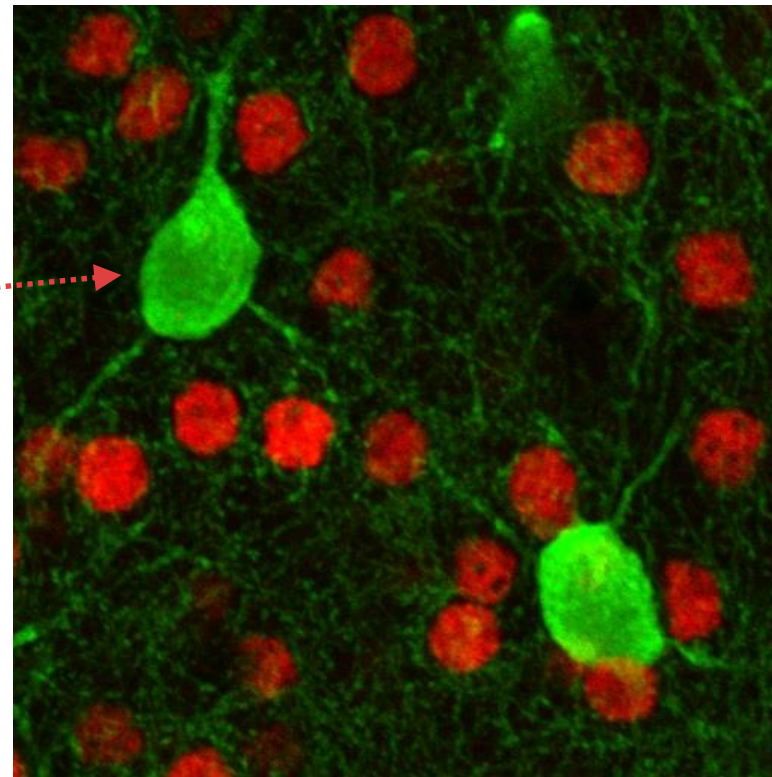
アプリケーション エンジニアリング部 テクニカルコンピューティング
太田 英司

ニューラルネットワークとは？

神経細胞（ニューロン）の数学的なモデル化に起源を持つ学習器

神経細胞（ニューロン）

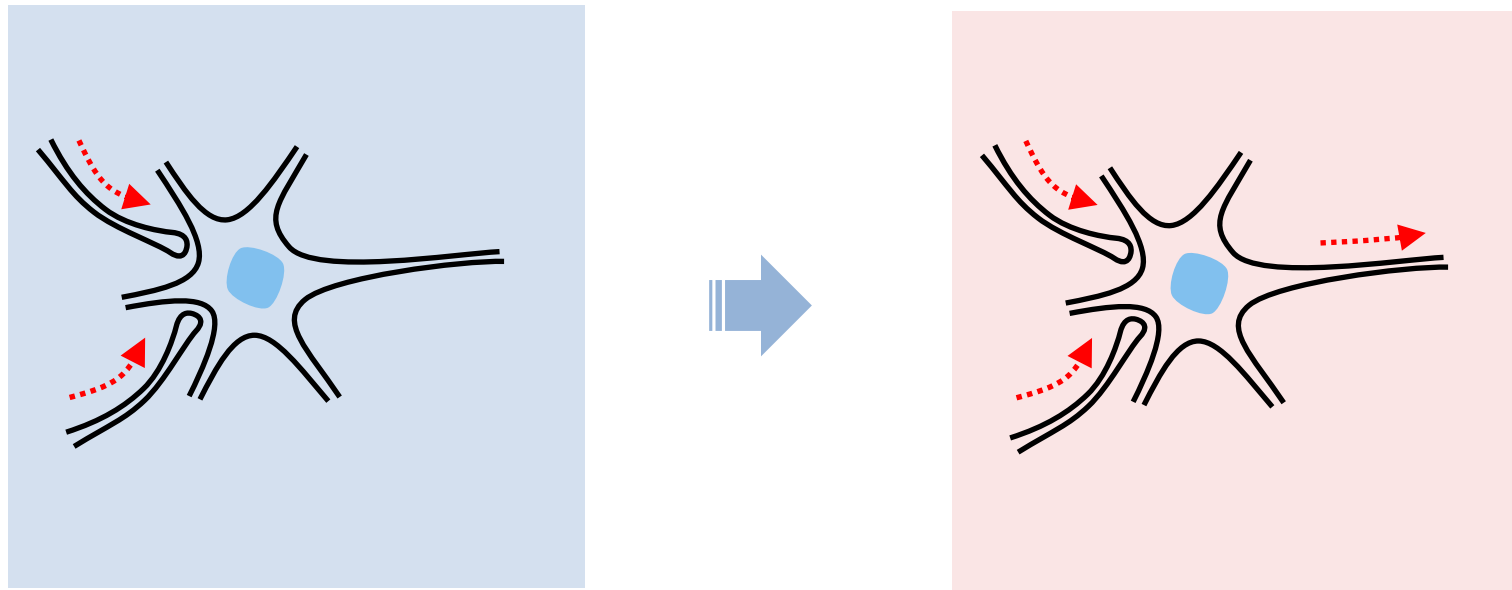
- 軸索によりネットワークを構成
- 電気的な興奮状態を伝え合う



画像提供： 独立行政法人 理化学研究所様

ニューラルネットワークとは？

他の神経細胞からの興奮状態が伝わり、電位のレベルが上昇する

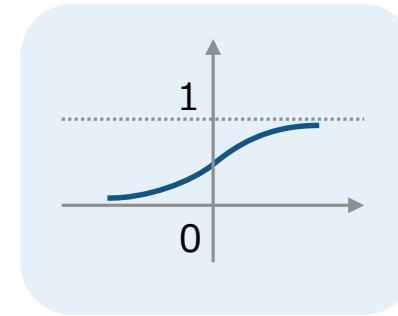
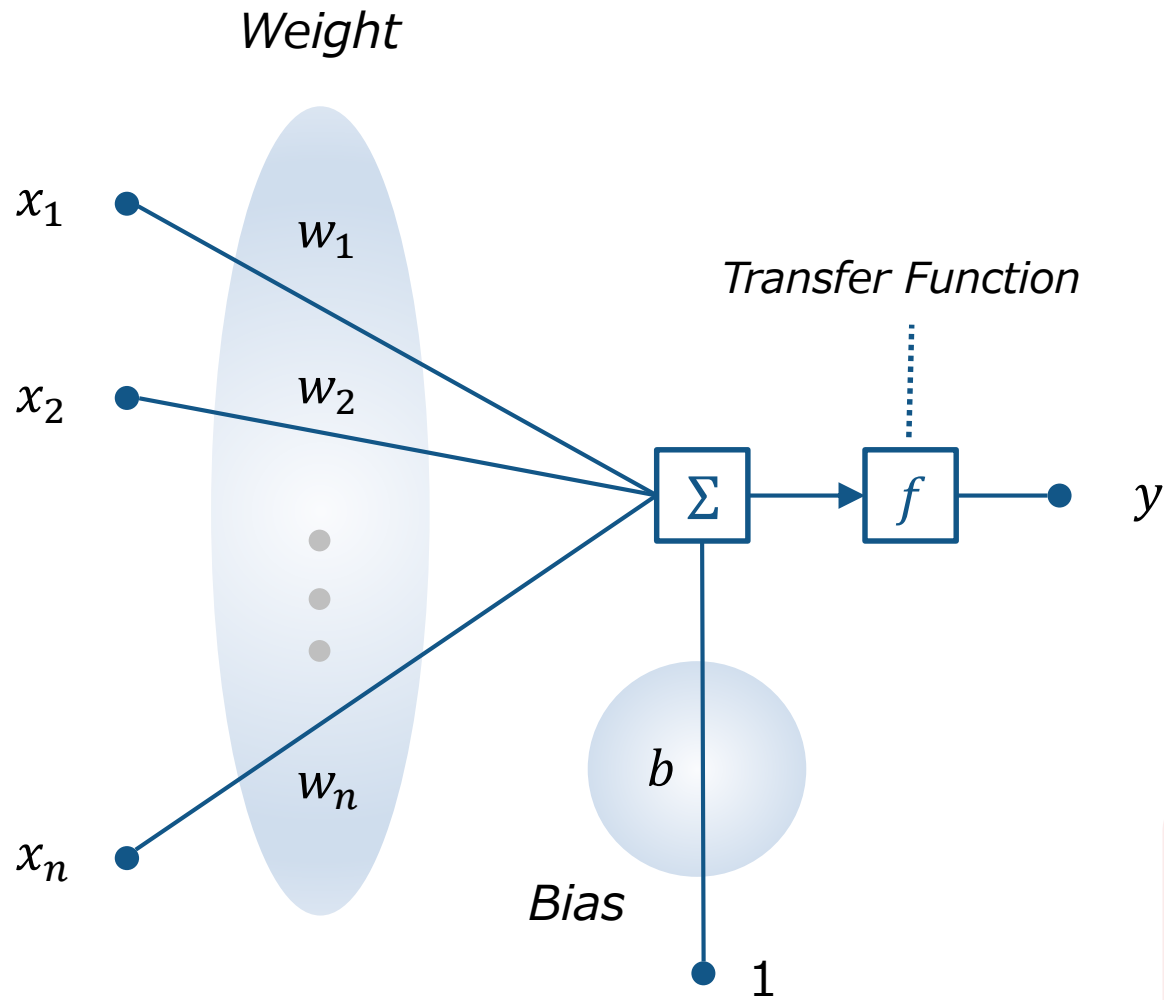


非興奮状態

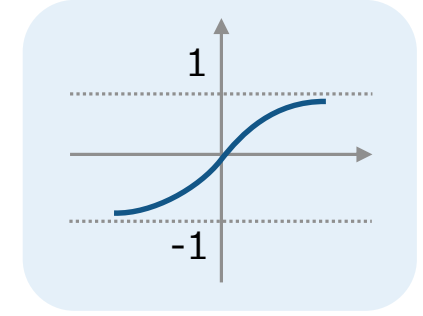
興奮状態

電位のレベルが閾値を超えると自身が発火する

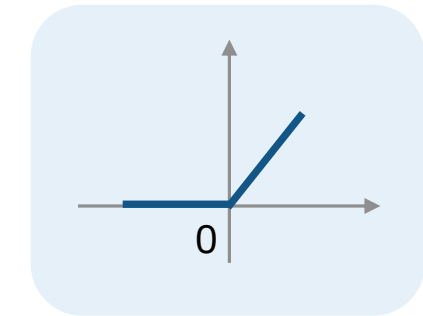
ニューラルネットワークとは？



Logistic Sigmoid



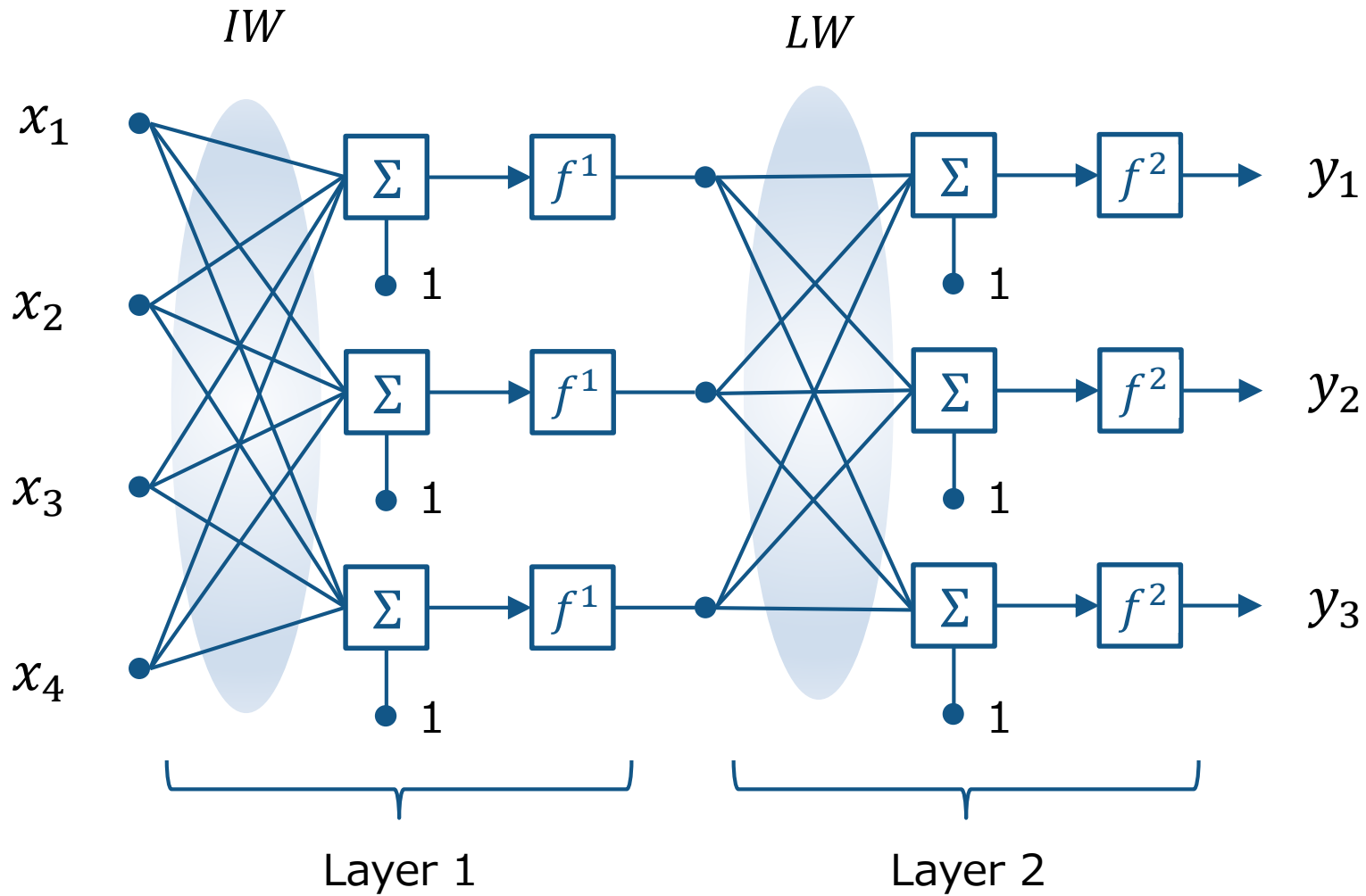
Tangent Sigmoid



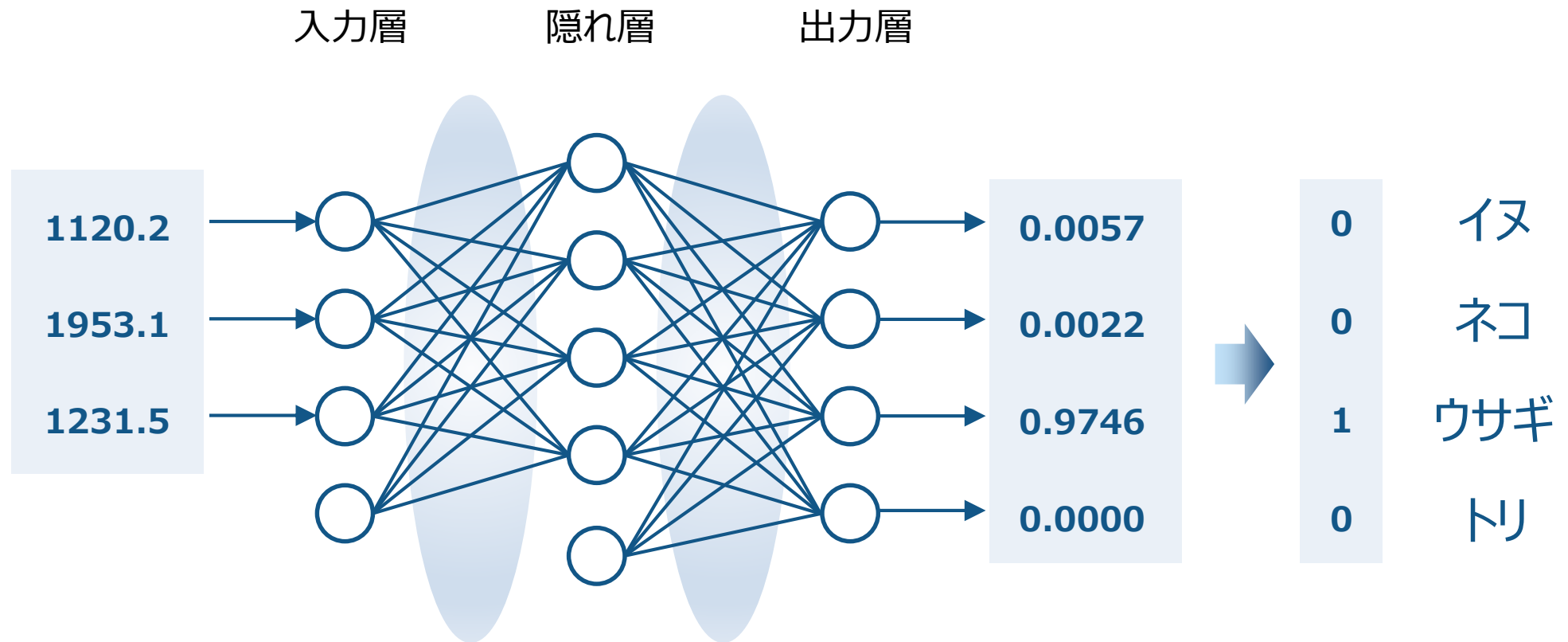
Rectified Linear Unit

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

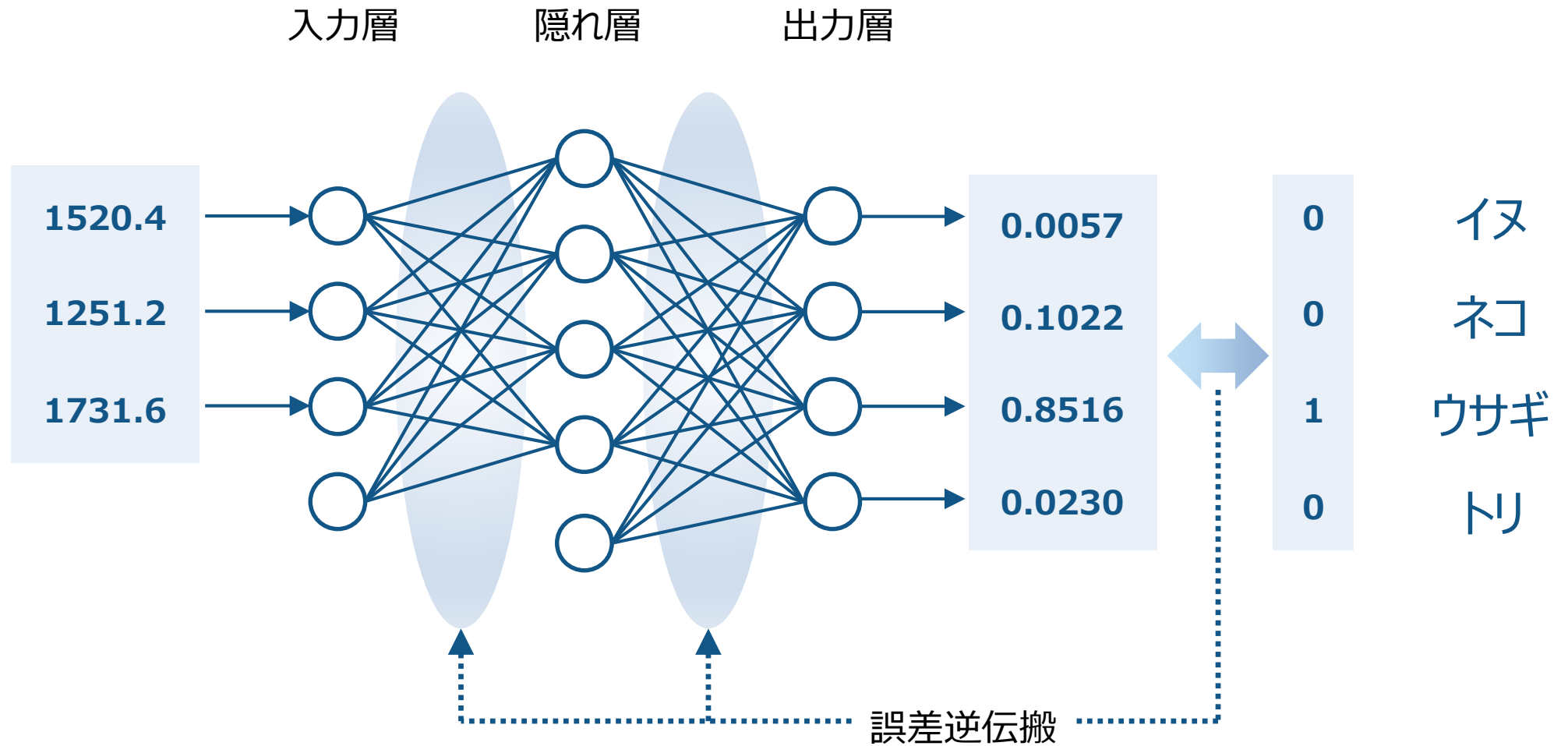
ニューラルネットワークとは？



ニューラルネットワークによる推論

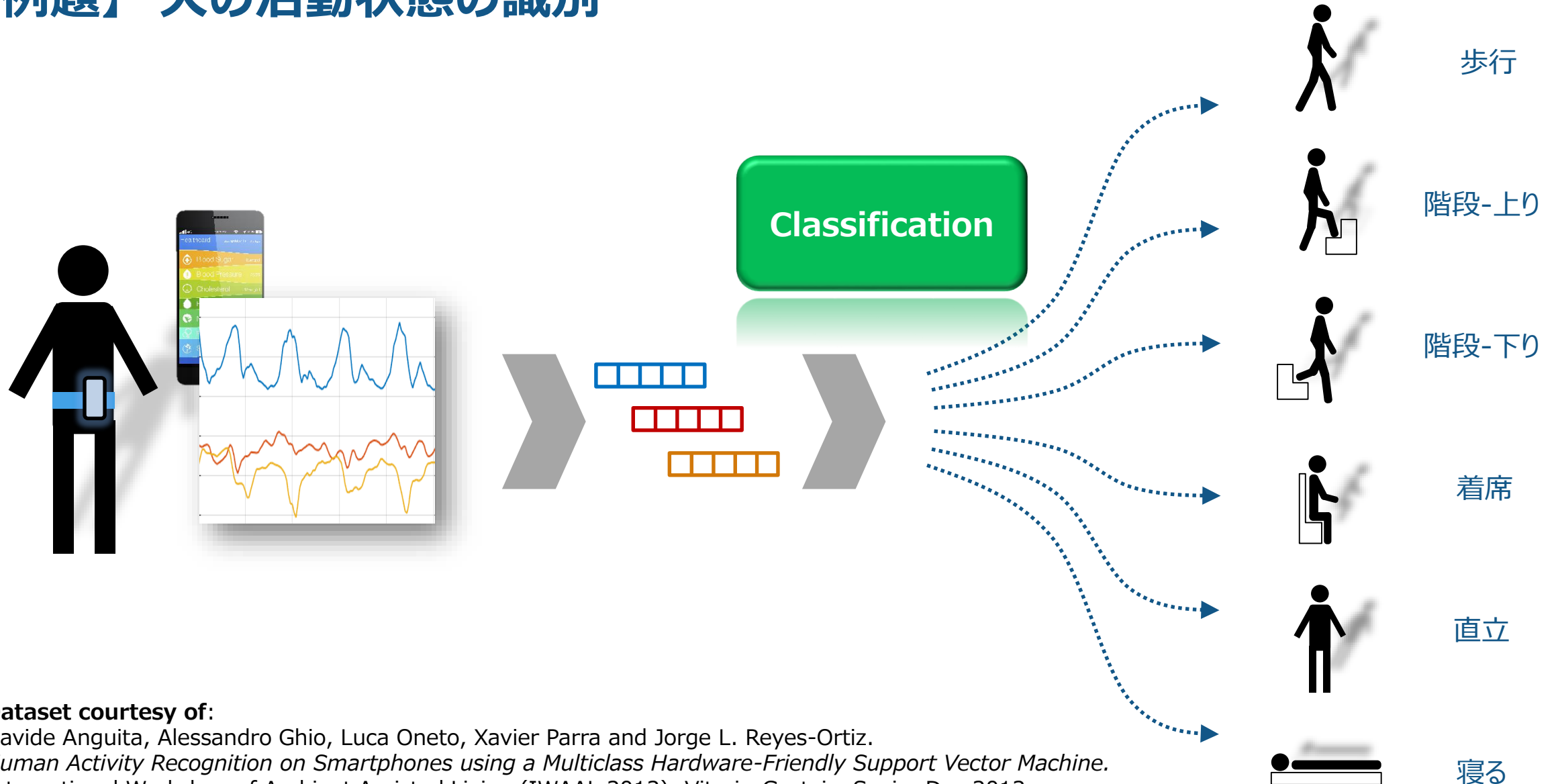


ニューラルネットワークの学習



時系列データの分類

【例題】人の活動状態の識別



Dataset courtesy of:

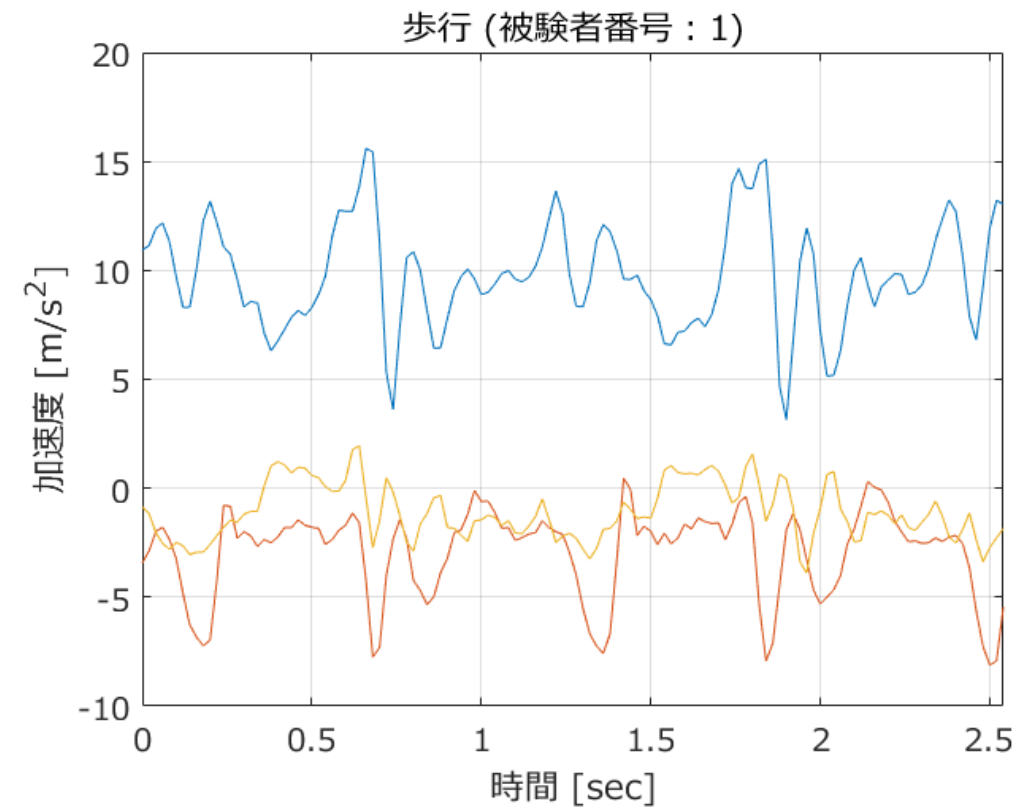
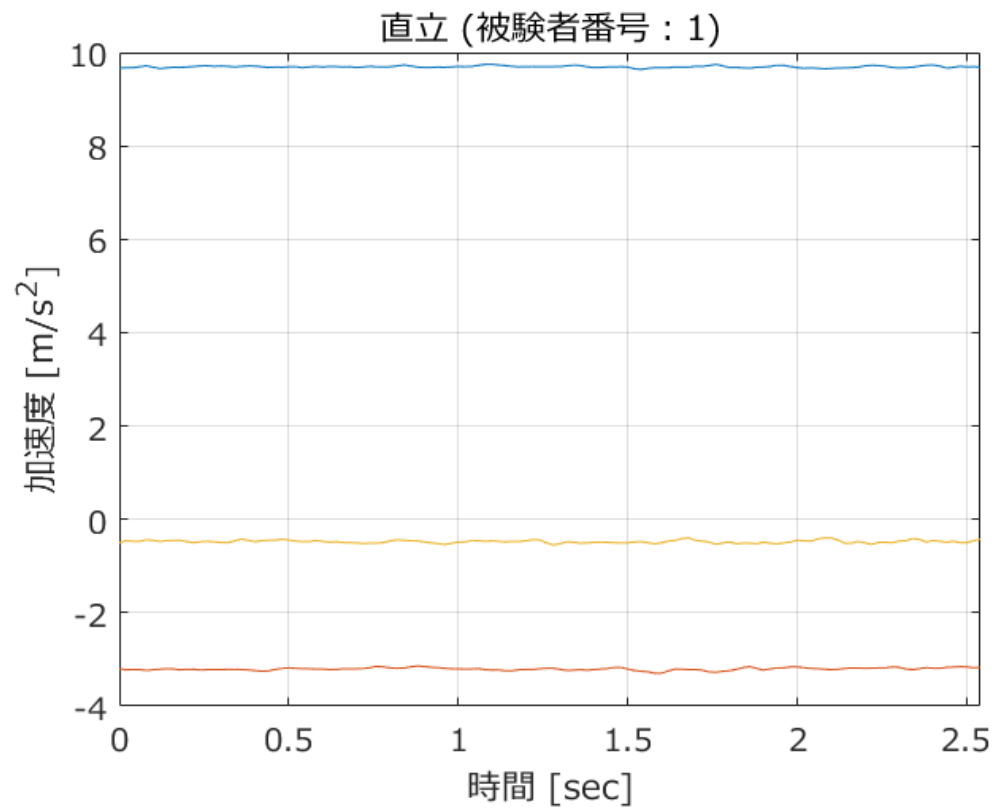
Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz.

Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine.

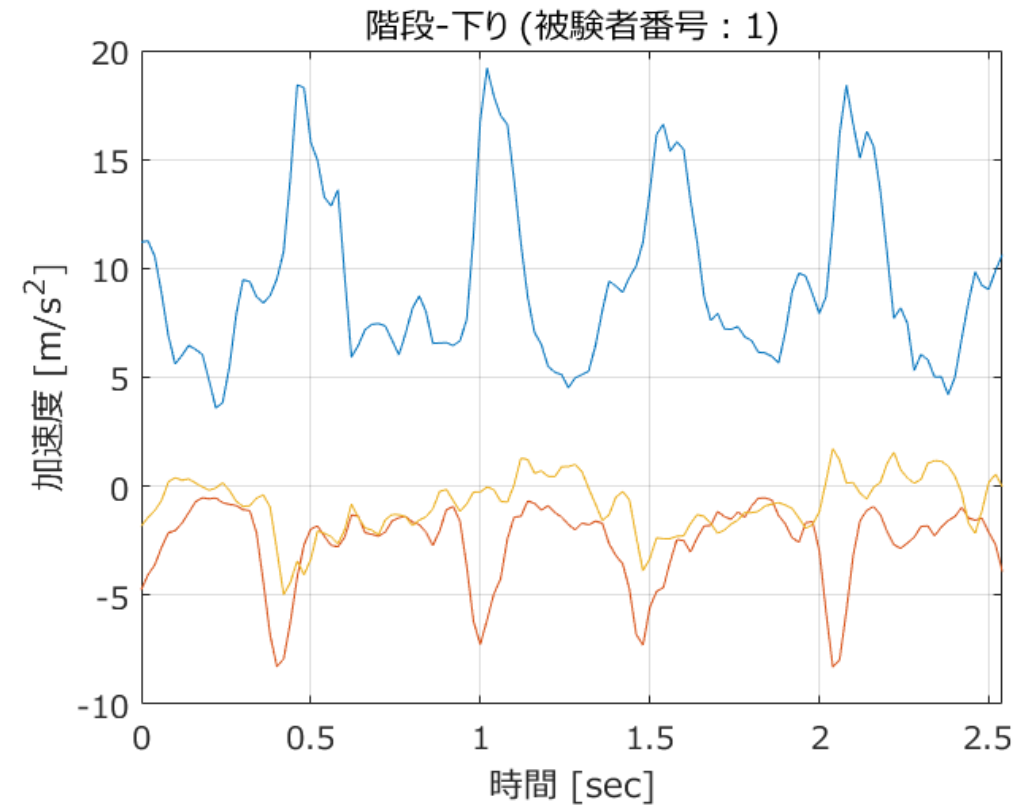
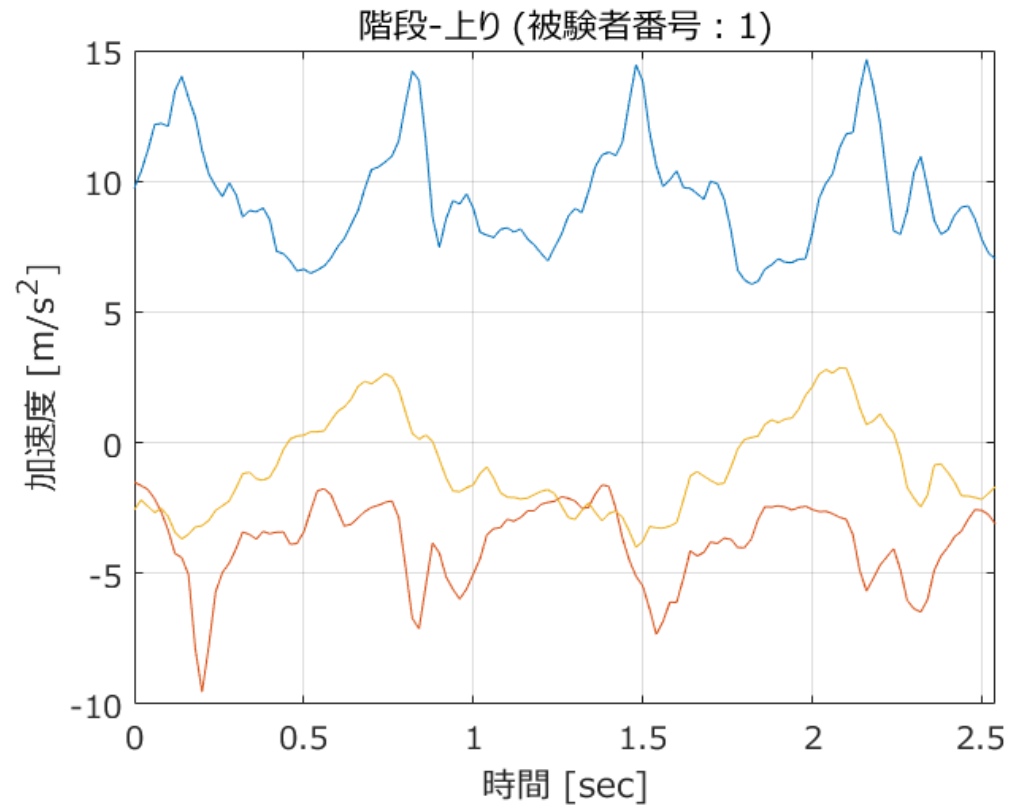
International Workshop of Ambient Assisted Living (IWAAL 2012). Vitoria-Gasteiz, Spain. Dec 2012

<http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones>

【例題】 人の活動状態の識別

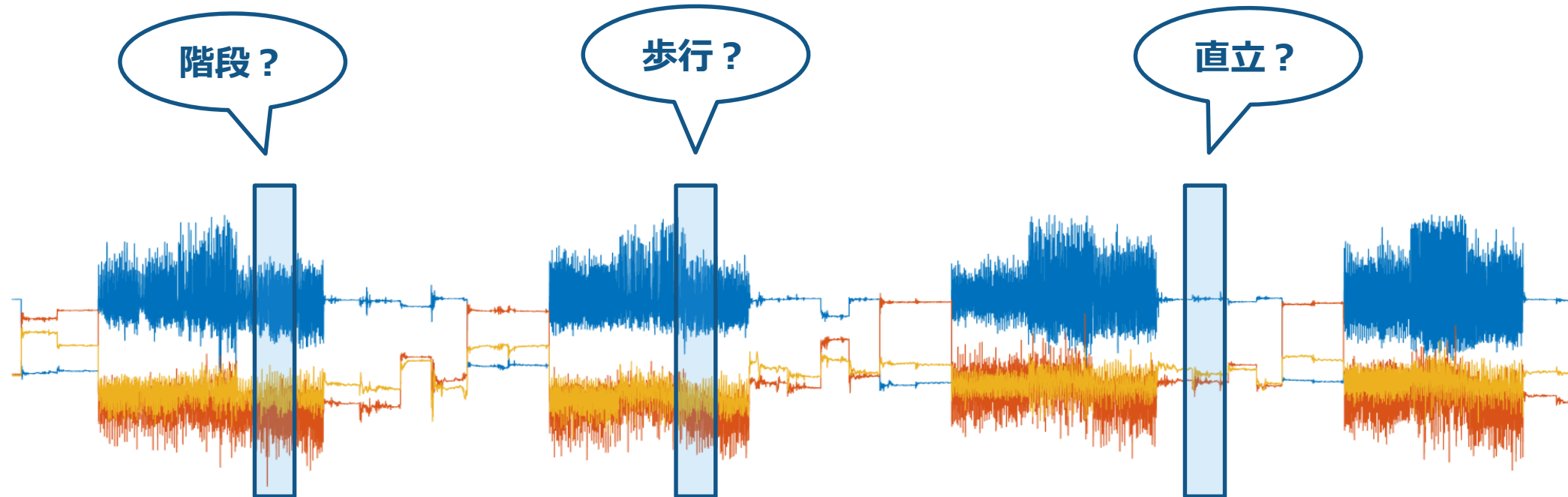


【例題】人の活動状態の識別



【例題】人の活動状態の識別

3軸の加速度センサーの情報を使って、人の活動状態を識別したい

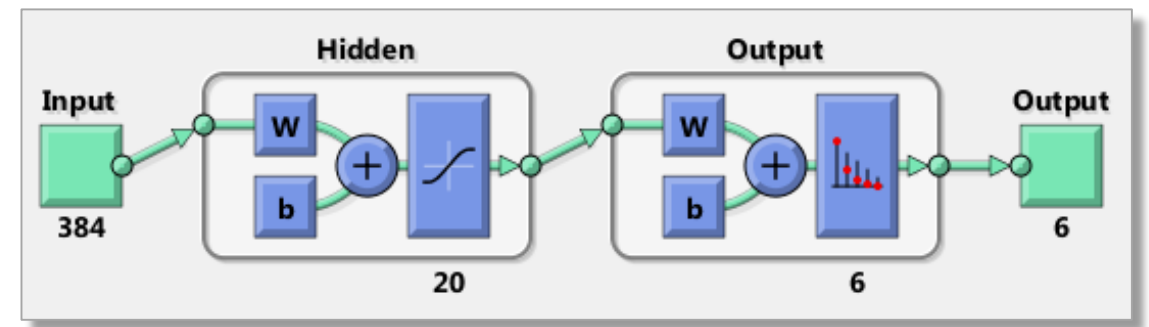
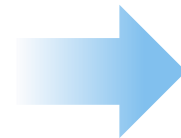
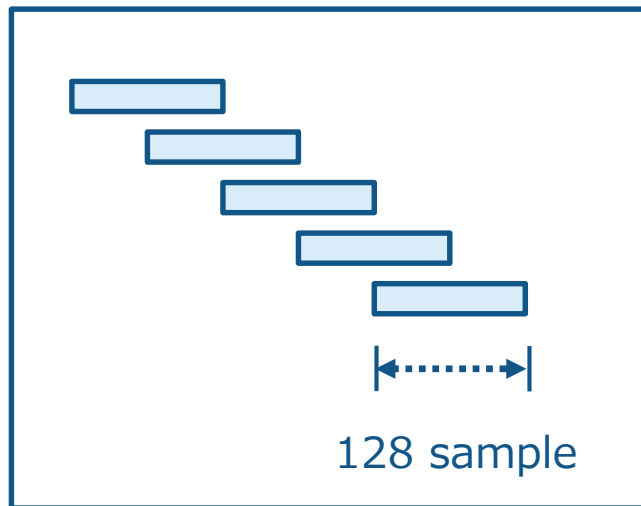


【例題】 人の活動状態の識別

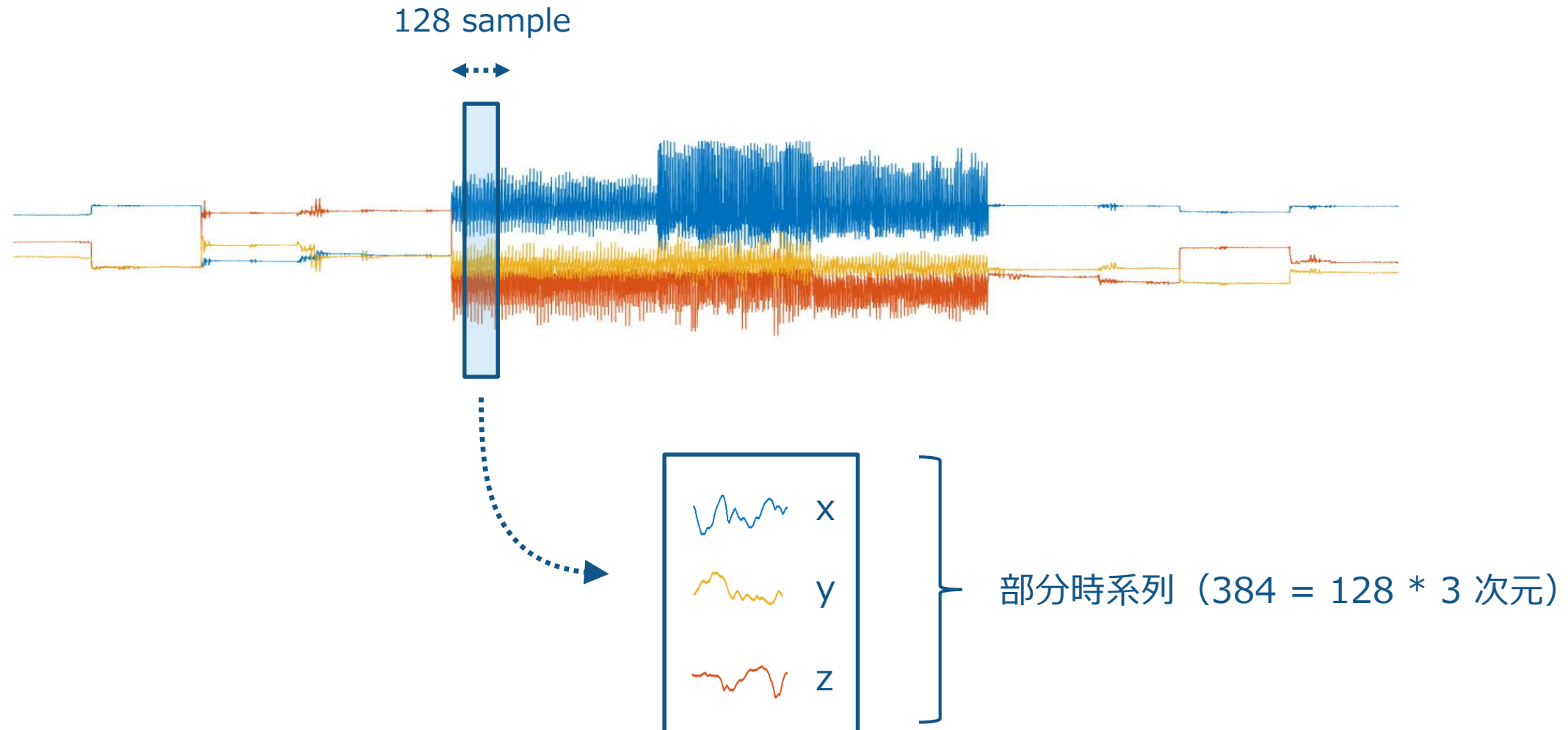
3軸加速度センサーのデータを窓幅128で切り出し、ニューラルネットより活動状態を識別する



部分時系列 (窓幅 128)

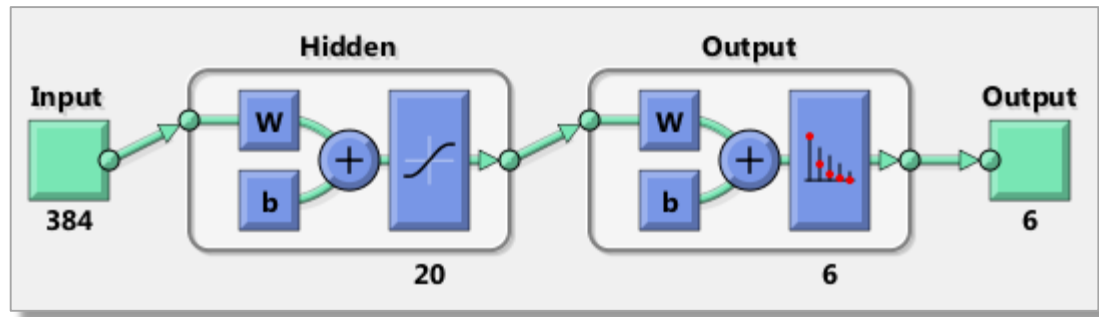


【例題】 人の活動状態の識別



【例題】 人の活動状態の識別

3軸の加速度センサーのデータを使って、人の活動状態を識別したい



入力：384 次元
部分時系列（加速度）

⋮	⋮	⋮	⋮	⋮

```
net = patternnet(20);
```

```
net = train(net, x, t);
```

```
t_hat = net(x);
```

```
plotconfusion(t, t_hat)
```

出力：6 次元
人の活動状態（ラベル）

1	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	0	0	0	0
0	0	0	1	1
0	0	0	0	0

1 of K 符号化 (one-hot ベクトル表現)

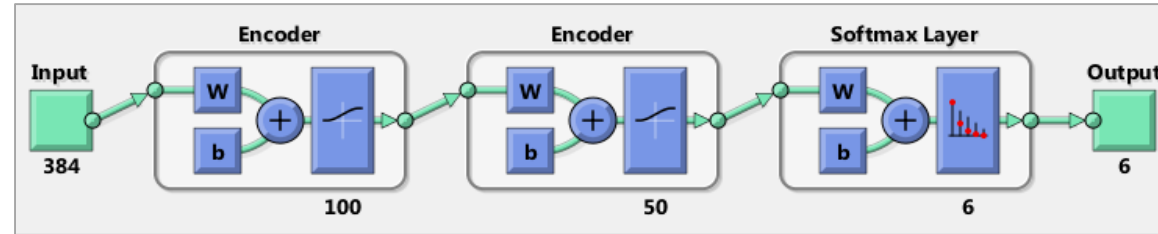
	ind2vec	1	0	0	0	1
1	→	0	0	1	0	0
3		0	1	0	0	0
2	←	0	0	0	1	0
4		0	0	0	1	0
1	vec2ind					

【例題】人の活動状態の識別

Stacked Autoencoder による人の活動状態の識別



部分時系列
(128 x 3 次元)



最初の2層のウェイトを Autoencoder により構成し最後に誤差逆伝搬法を用いて学習する



歩行

階段-上り

階段-下り

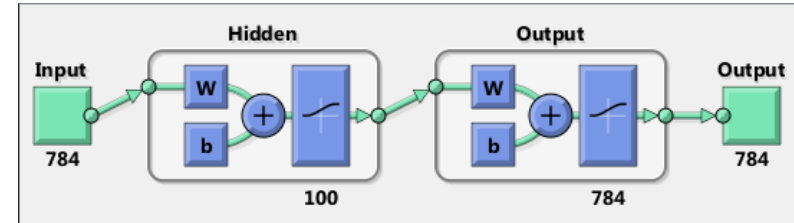


人の活動状態
(6-状態)

自己符号化器 (Autoencoder) とは？

入力したデータと同じものが出力側で再生されるように学習させるネットワーク

- 恒等写像となるようなウェイトを学習
- 主に DNN の事前学習に利用
- 主成分分析・スパース符号化とも関連

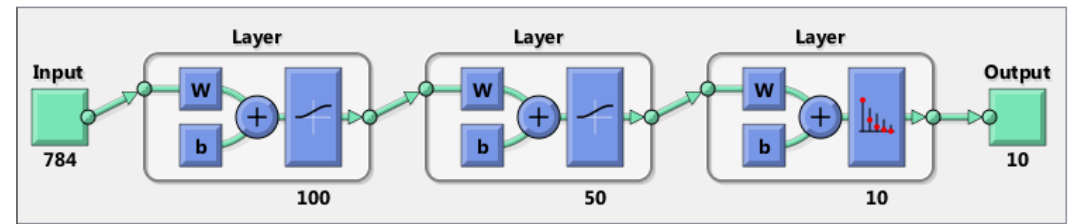


自己符号化器 (Autoencoder)

【コード例】 自己符号化器の学習

```
autoenc = trainAutoencoder(X, hiddenSize,...
    'L2WeightRegularization', 0.001, ...
    'SparsityRegularization', 4, ...
    'SparsityProportion', 0.05, ...
    'DecoderTransferFunction', 'purelin'
    'useGPU', true);
```

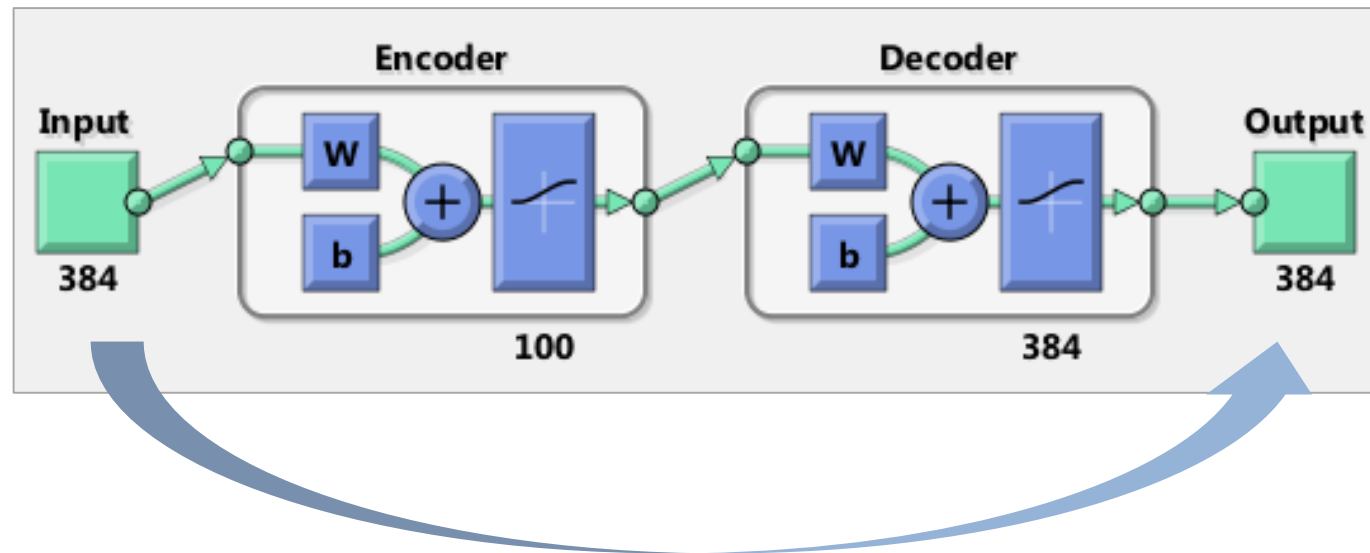
←..... GPUによる高速化



積層自己符号化器 (Stacked Autoencoder)

Step 1 : Autoencoder によるウェイトの学習

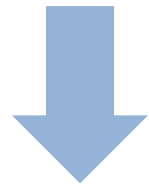
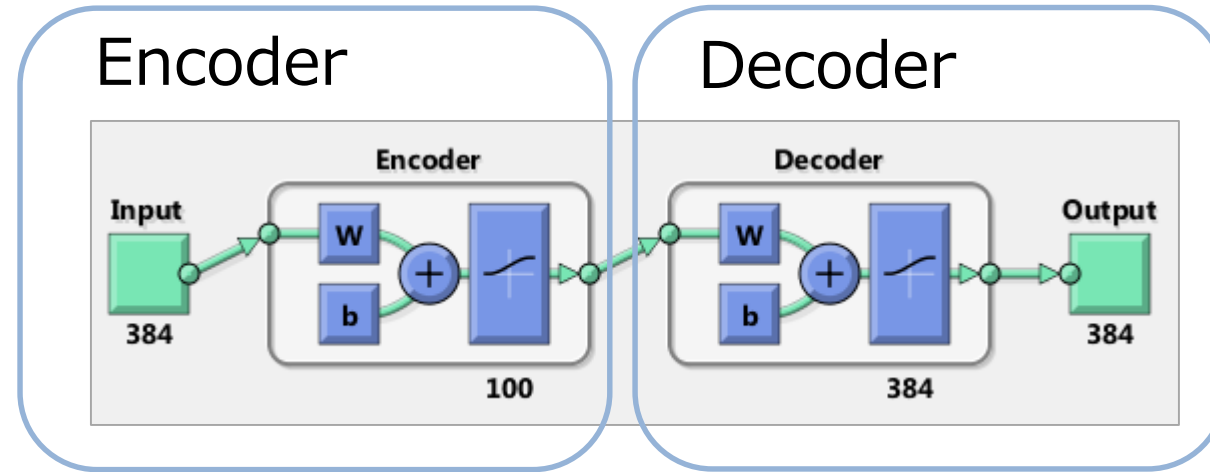
Autoencoder



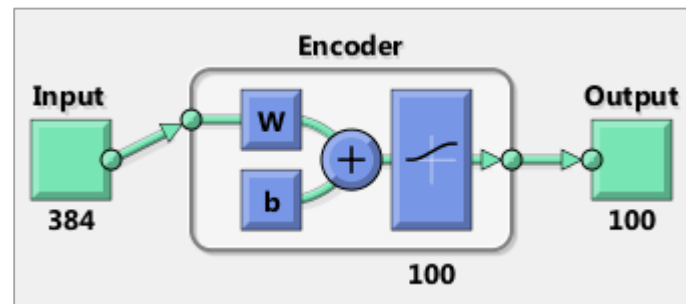
上のネットワークが恒等写像になるようにウェイトを学習する

※ 正確には、MATLAB の Autoencoder は Sparse Autoencoder と呼ばれる中間層の活性化に疎性を仮定したものになっている。詳細は後述の PDF を参照

Step 2 : Encoder 部分のみ残す

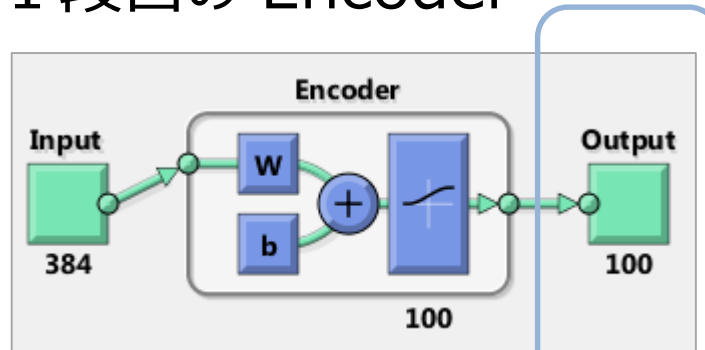


Encoder 部分のみ残す

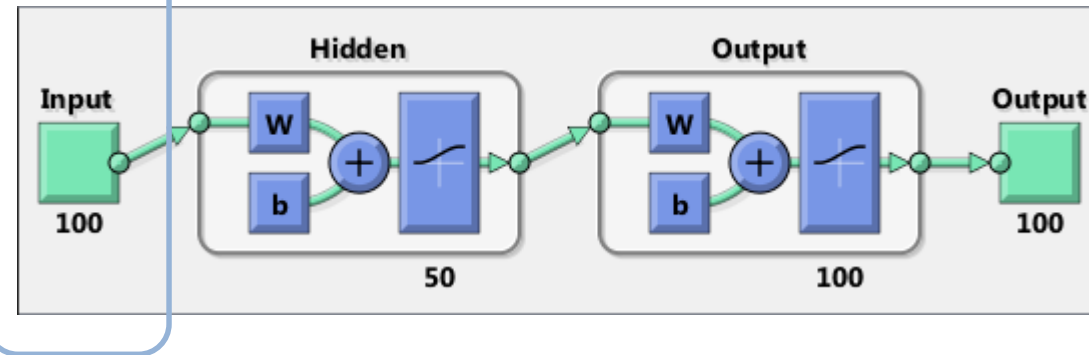


Step 3 : Autoencoder によるウェイトの学習

1 段目の Encoder



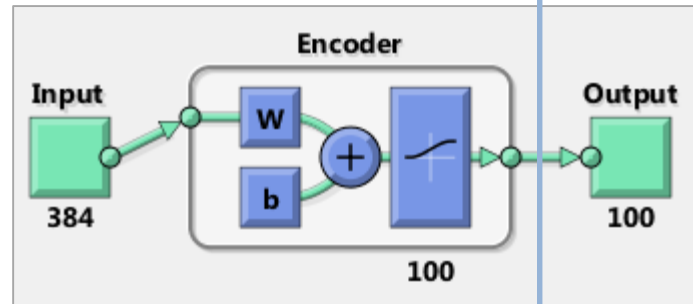
2 段目の Autoencoder



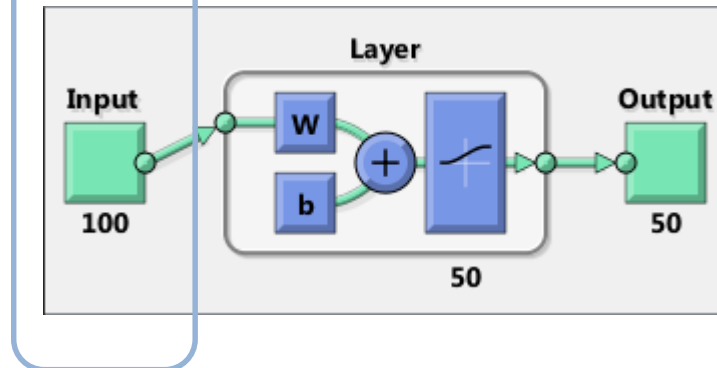
1 段目の Encoder 出力を使い、次の Autoencoder を学習させる

Step 4 : Encoder 部分のみ残す

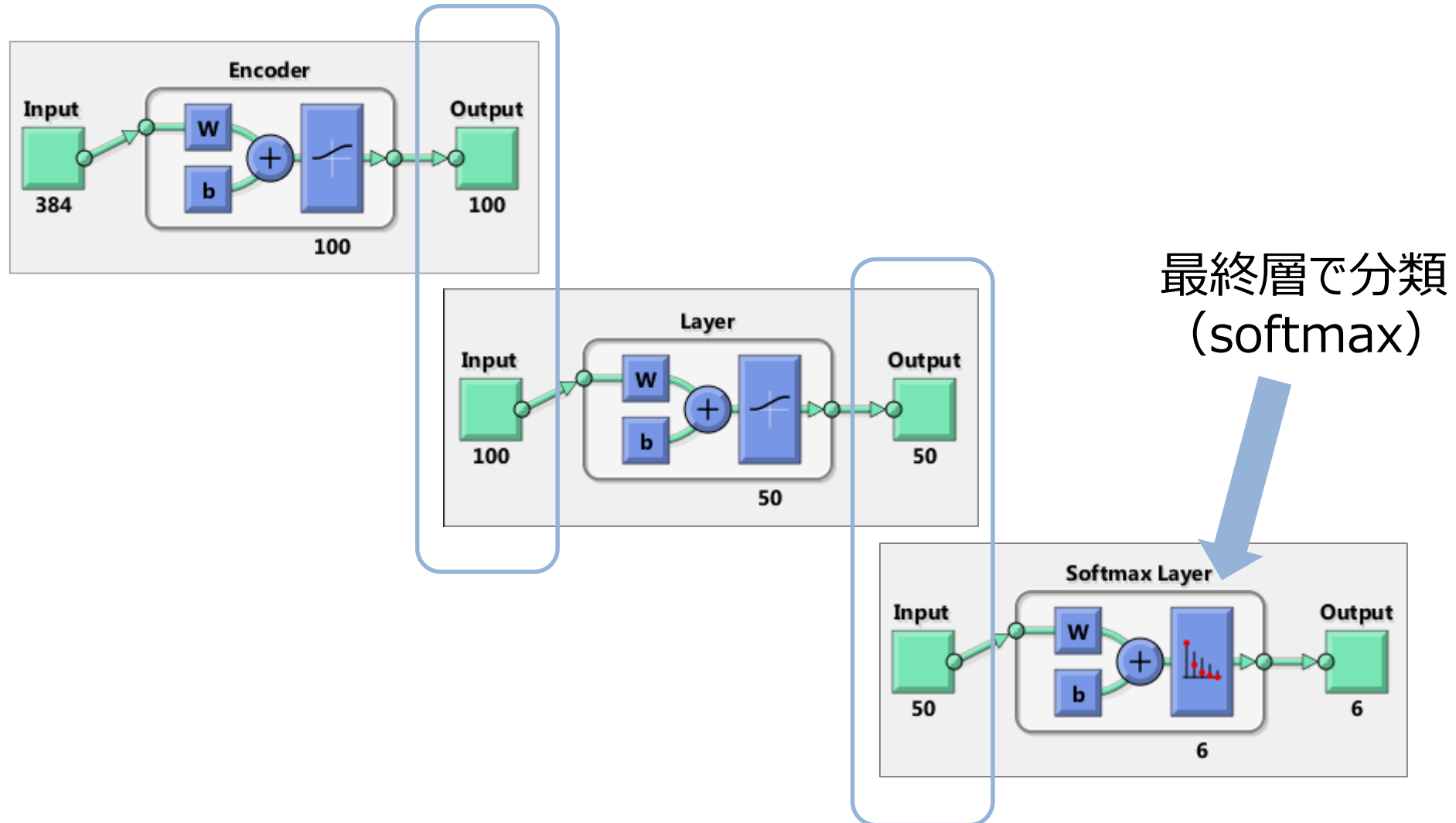
1 段目の Encoder



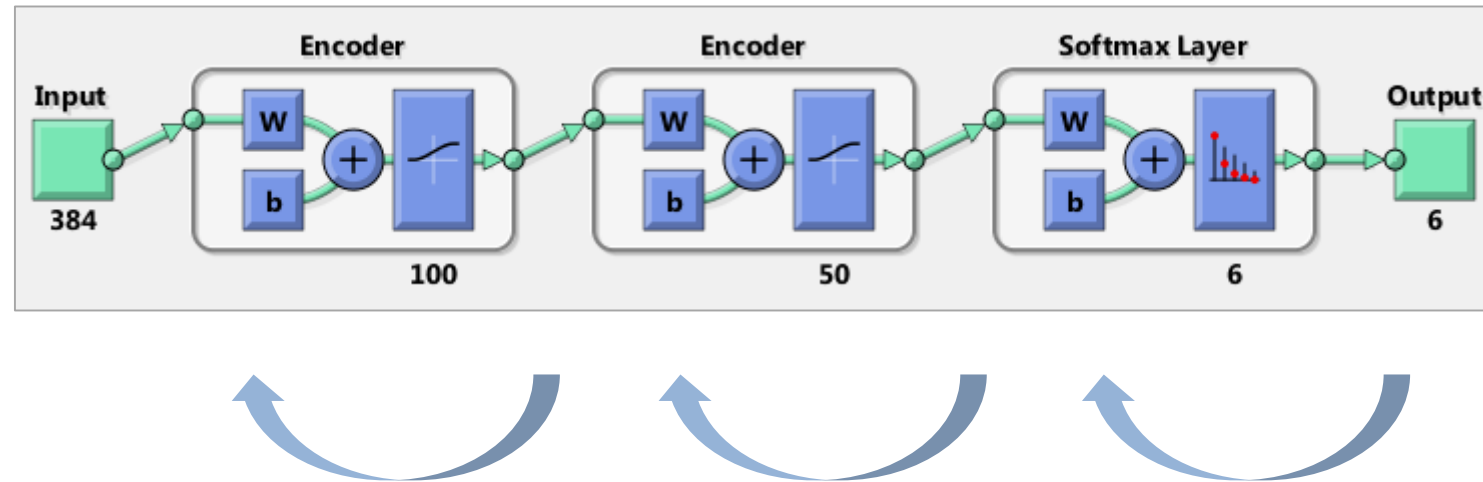
2 段目の Encoder



Step 5 : 分類のために最終層を追加



Step 6 : 微調整 (Fine Tuning) を行う



誤差逆伝搬法により、ウェイトの微調整 (Fine Tuning) を行う

【例題】人の活動状態の識別

積層自己符号化器として層を増やしたところ
分類性能が 3% 程度改善！

Accuracy : 0.81371

歩行	349	110	28	3	6	0
階段-上り	66	355	41	8	1	0
階段-下り	9	61	339	8	3	0
着席	1	23	0	360	107	0
直立	0	1	0	73	458	0
寝る	0	0	0	0	0	537
	歩行	階段-上り	階段-下り	着席	直立	寝る

混合行列 (浅いネットワークの場合)

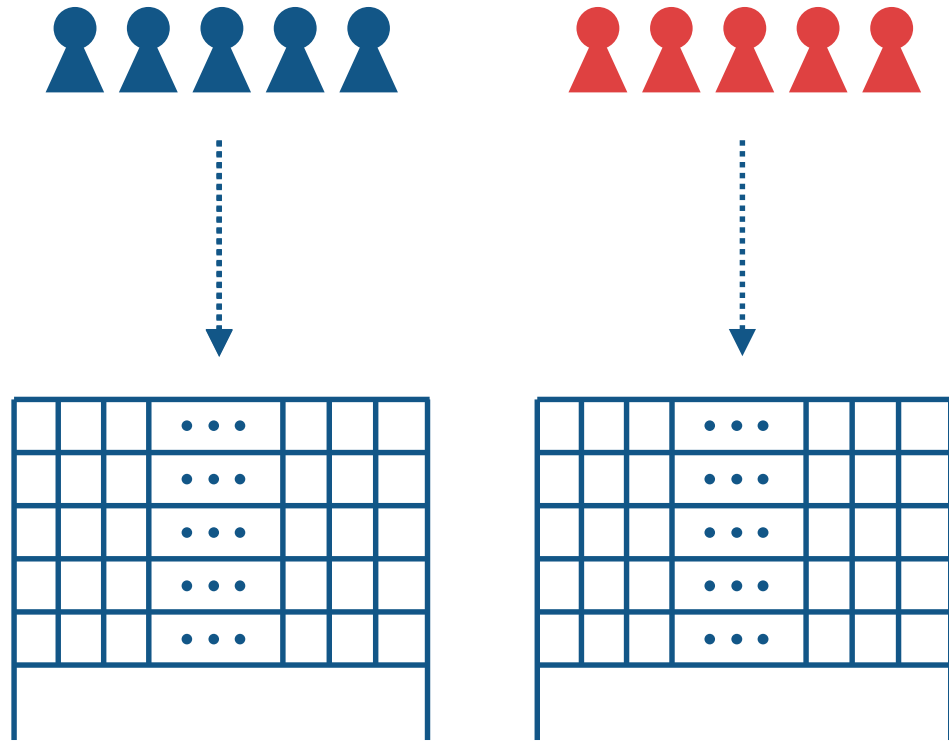
Accuracy : 0.84323

歩行	412	27	25	11	21	0
階段-上り	49	397	25	0	0	0
階段-下り	13	74	333	0	0	0
着席	1	17	0	382	86	5
直立	5	0	0	103	424	0
寝る	0	0	0	0	0	537
	歩行	階段-上り	階段-下り	着席	直立	寝る

混合行列 (深いネットワークの場合)

【例題】人の活動状態の識別

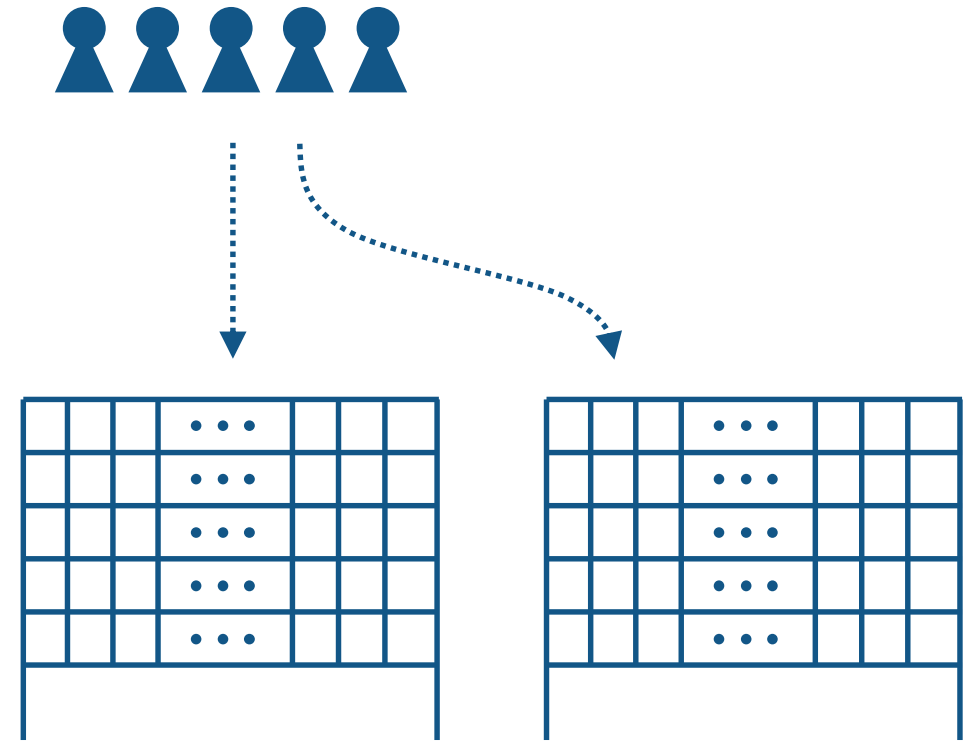
★ 学習データと評価データで被験者が異なる場合



学習用データセット

評価用データセット

★ 学習データと評価データで被験者が変わらない場合
(例えば、取得する時間だけを変えるなど)



学習用データセット

評価用データセット

【例題】人の活動状態の識別

精度評価時に同じ被験者のデータを使うと、更に性能が改善する

Accuracy : 0.89524

歩行	342	19	12	4	5	0
階段-上り	47	279	9	0	3	2
階段-下り	9	26	258	2	2	0
着席	0	0	0	362	32	0
直立	0	0	0	59	344	0
寝る	0	0	0	0	0	389
	歩行	階段-上り	階段-下り	着席	直立	寝る

混合行列（浅いネットワークの場合）

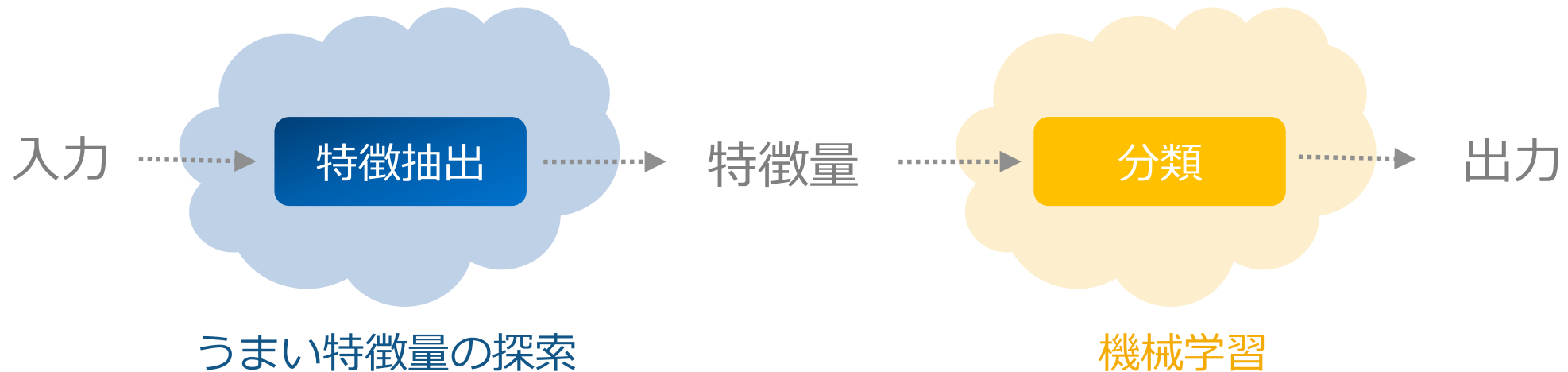
Accuracy : 0.93923

歩行	351	12	9	0	10	0
階段-上り	22	317	1	0	0	0
階段-下り	5	10	282	0	0	0
着席	1	0	0	364	29	0
直立	0	0	0	35	368	0
寝る	0	0	0	0	0	389
	歩行	階段-上り	階段-下り	着席	直立	寝る

混合行列（深いネットワークの場合）

特徴抽出と Deep Learning

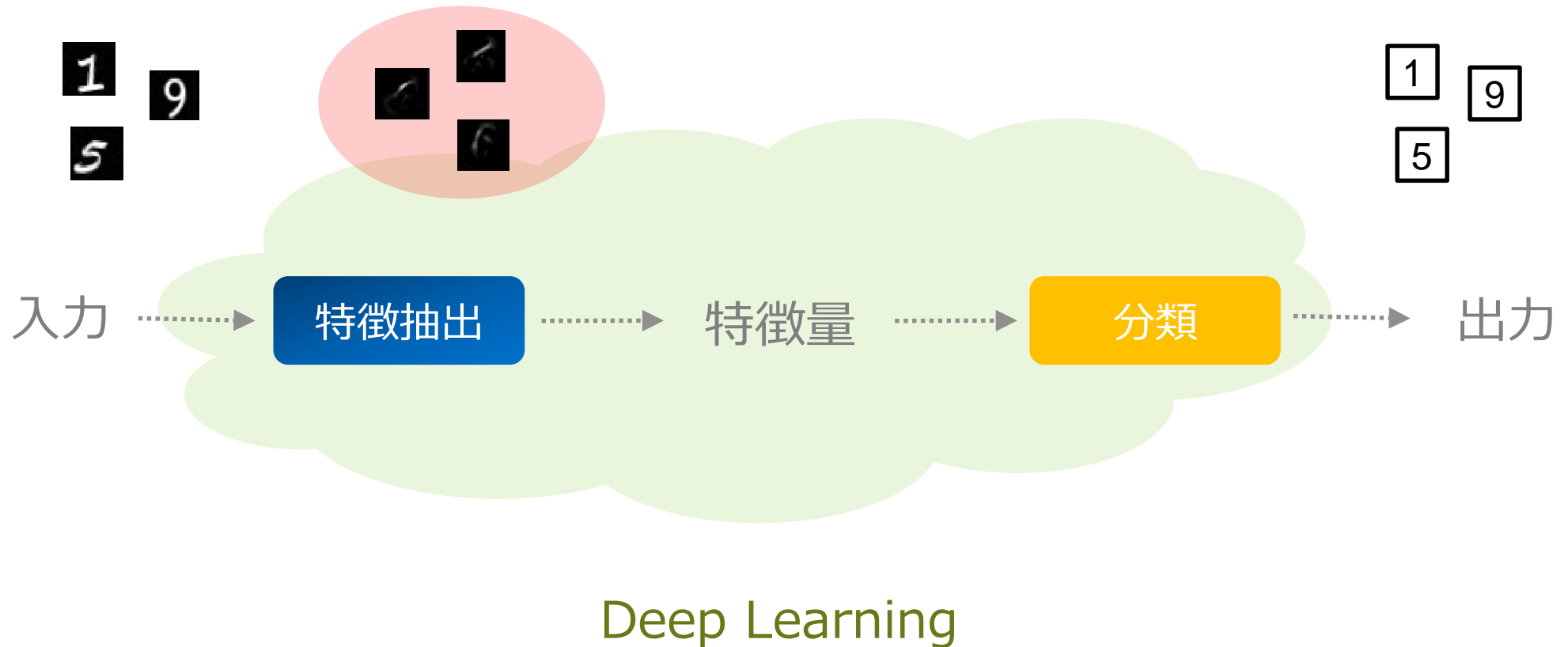
Deep Learning 以前は、特徴抽出の部分を試行錯誤で人間が探索していた



データに適した特徴量を見つけ出すことがよい性能を出すためのキーであった

特徴抽出と Deep Learning

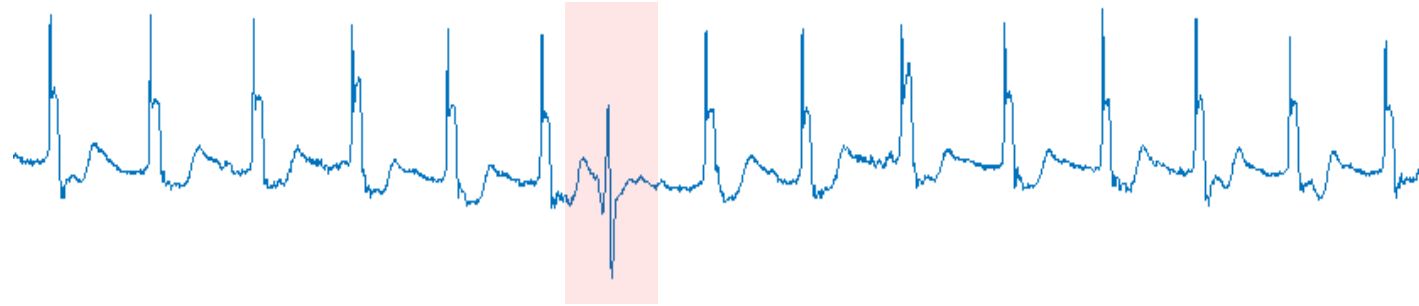
Deep Learning では、特徴抽出の部分もデータから学習させることができる



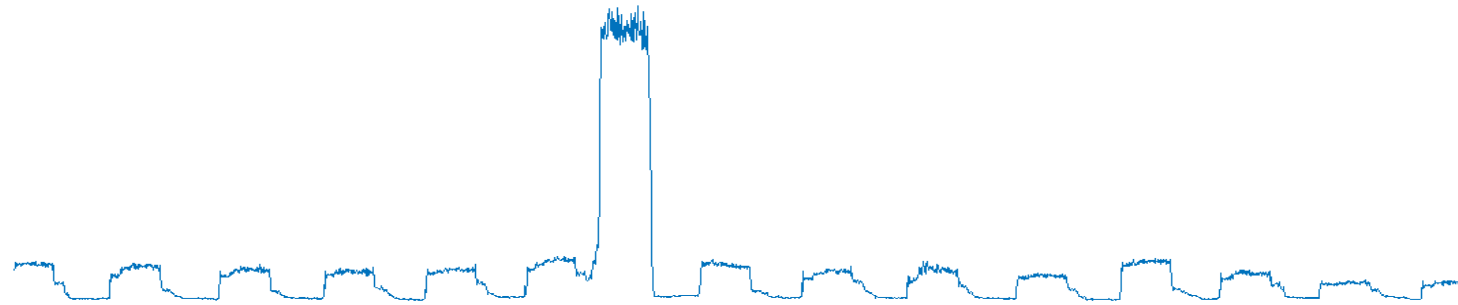
時系列データの異常検知

センサーデータからの異常検出

心電図



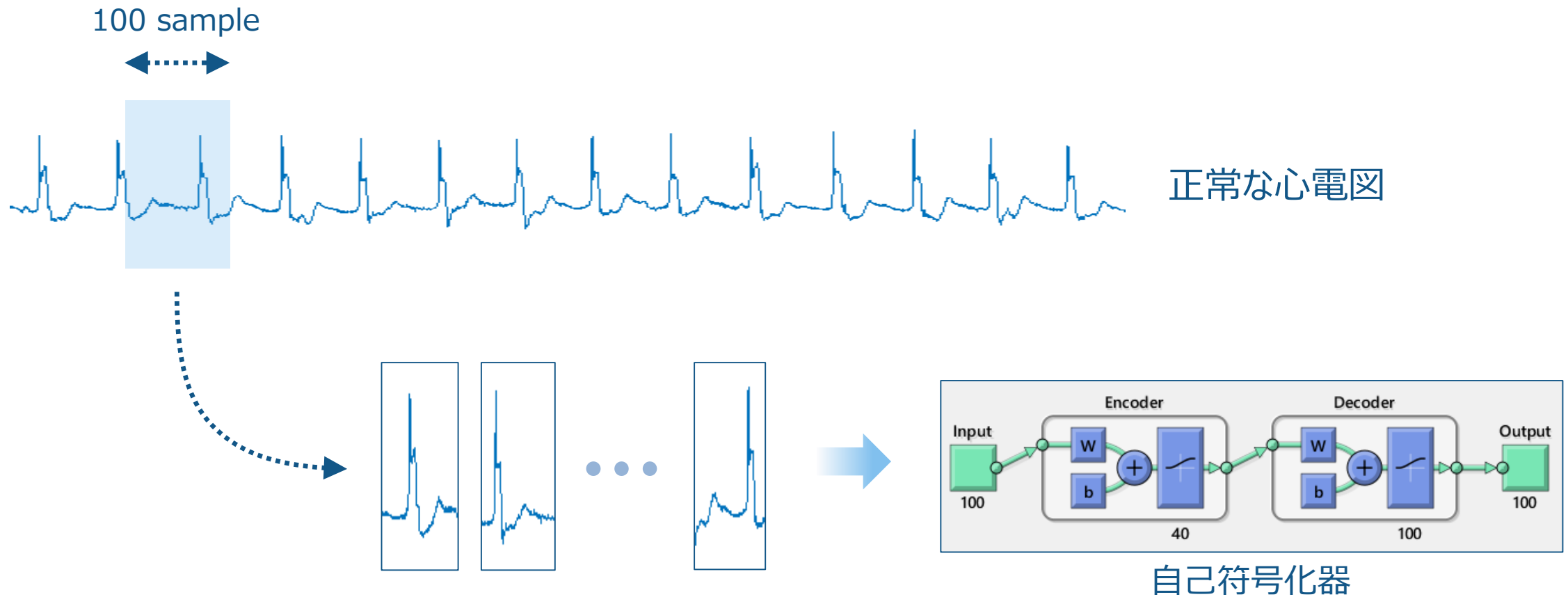
異常度



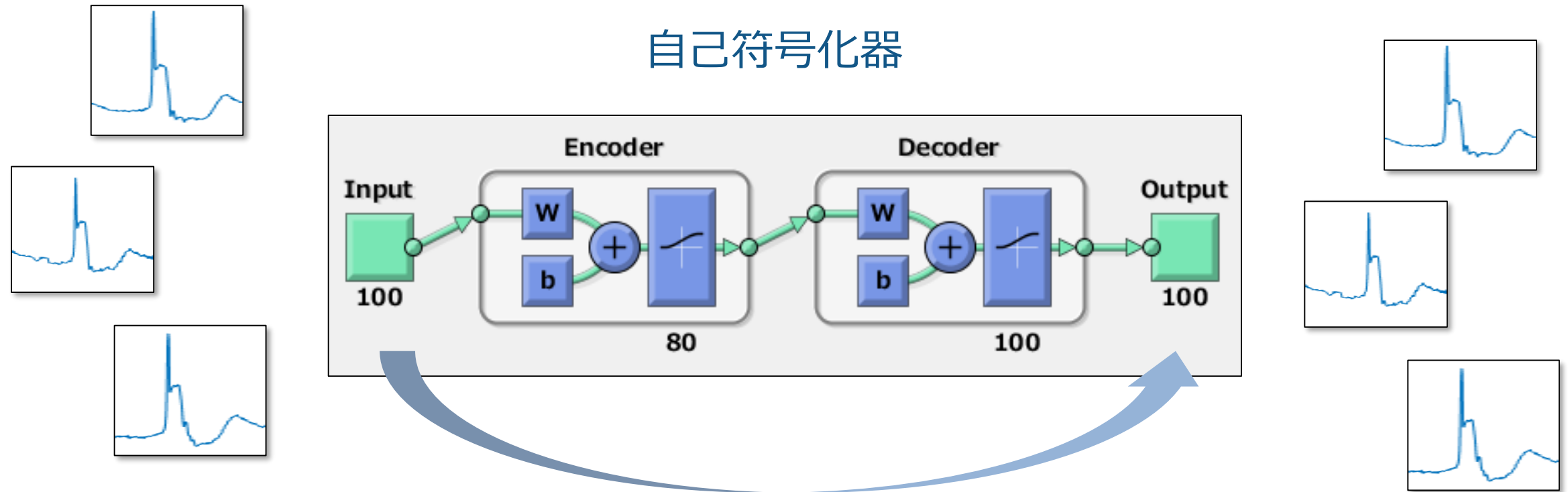
Keogh, E., Lin, J. and Fu, A. : HOT SAX : Efficiently Finding the Most Unusual Time Series Subsequence, in Proceedings of the Fifth IEEE International Conference on Data Mining, ICDM 05, pp.226-233

自己符号化器による特徴抽出

正常な心電図から部分時系列を生成して、自己符号化器に学習させる



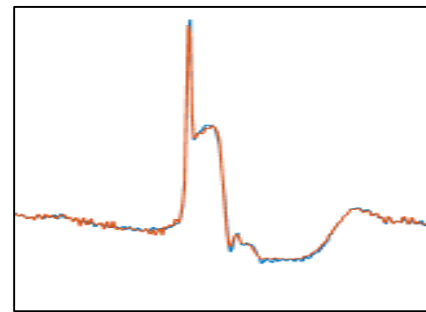
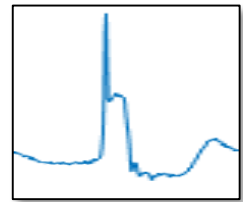
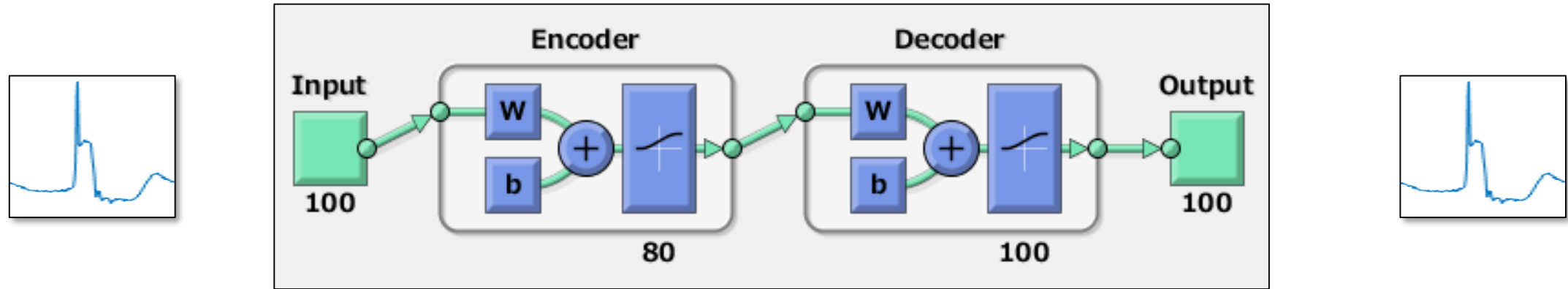
自己符号化器による異常検出



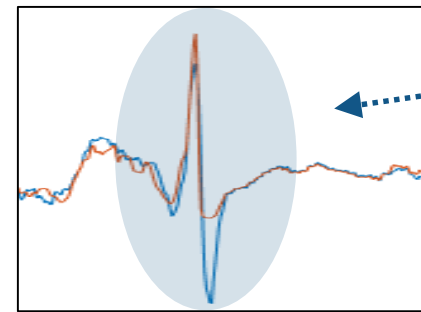
元の正常な波形を再生するようにウェイトを学習する

自己符号化器による異常検出

正常な波形の場合は、きれいに元の波形を再生するが、異常な波形の場合はうまく再生できない！



正常な波形の場合



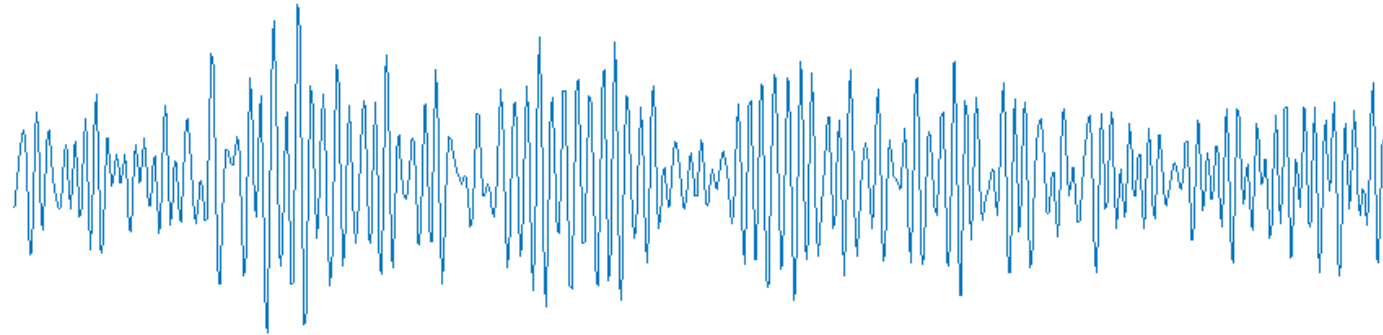
異常な波形の場合

異常な波形は正常な
波形に戻そうとする

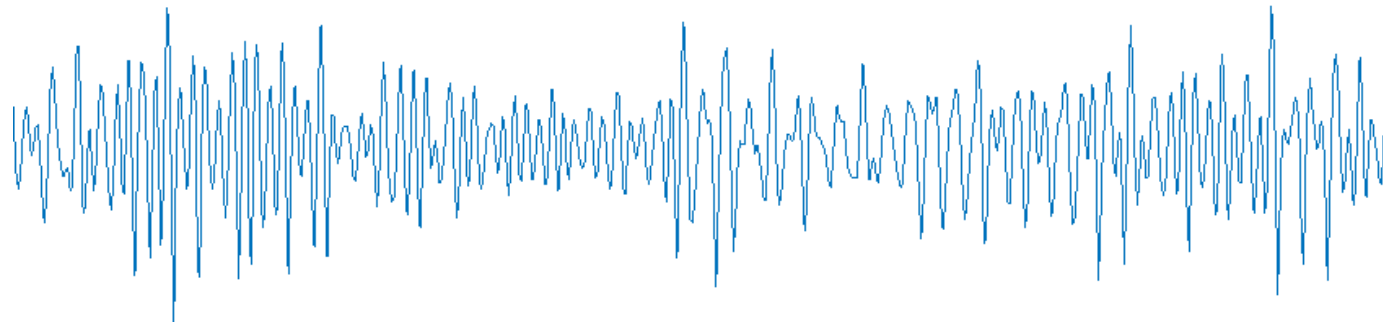
構造物の異常検出

ある構造物に取り付けられた振動データの比較

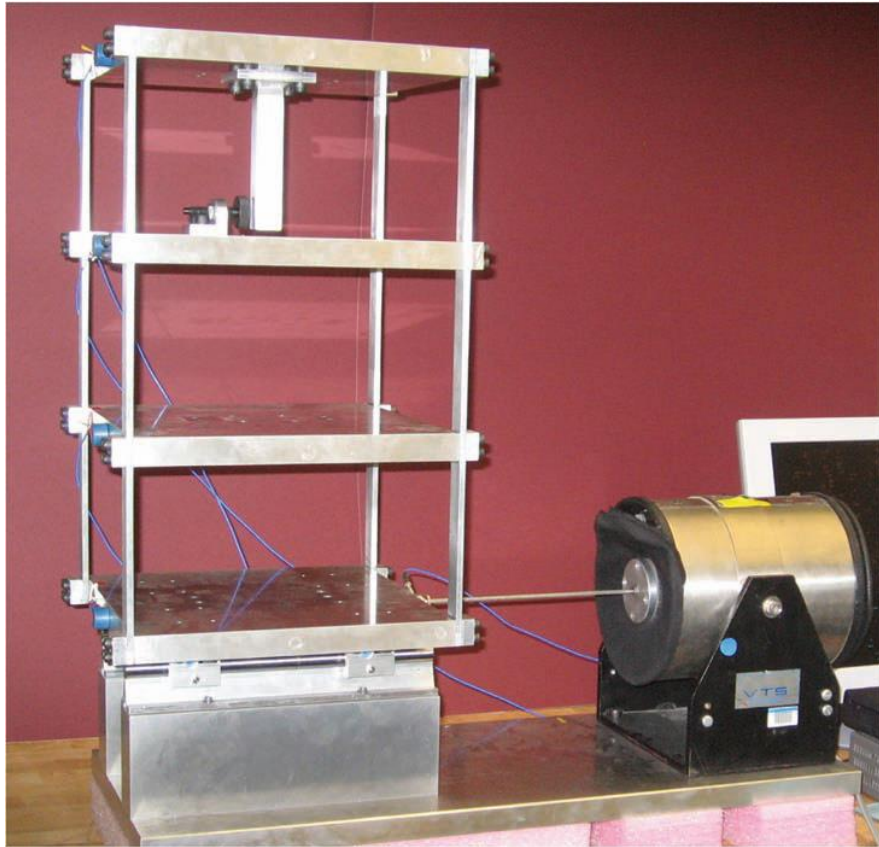
正常



異常



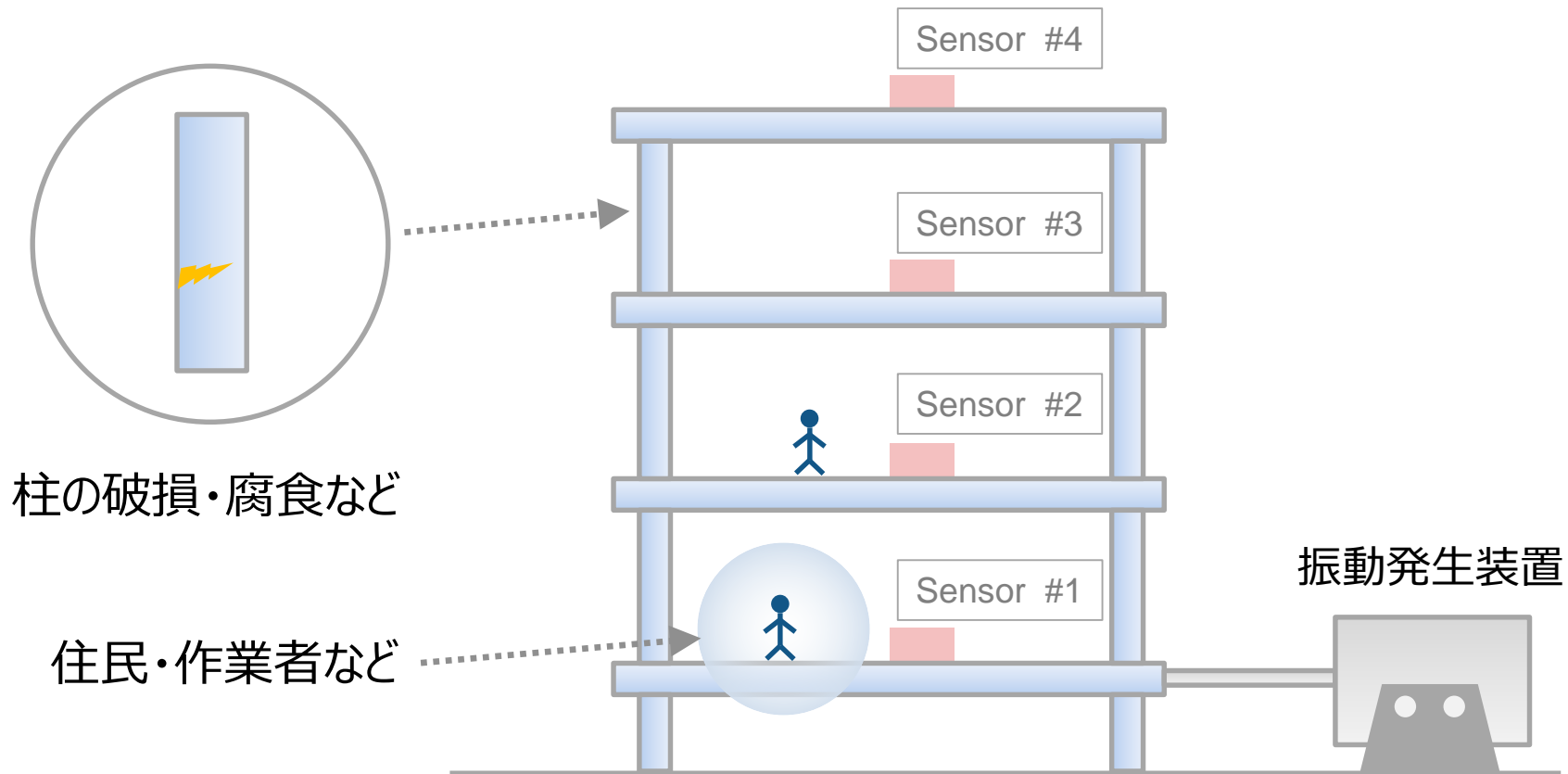
【例題】 構造ヘルスマニタリング



- 実験装置
 - 3階建ての建物の模型
 - 4ヶ所の加速度センサー
 - 1階部分に振動発生装置
 - 人工的な損傷状態の発生
- 実験目的
 - センサーによる異常検出

Reference: Figueiredo, E., Park, G., Figueiras, J., Farrar, C., & Worden, K. (2009). Structural Health Monitoring Algorithm Comparisons using Standard Data Sets. Los Alamos National Laboratory Report: LA-14393

【例題】 構造ヘルスマモニタリング



Reference: Figueiredo, E., Park, G., Figueiras, J., Farrar, C., & Worden, K. (2009). Structural Health Monitoring Algorithm Comparisons using Standard Data Sets. Los Alamos National Laboratory Report: LA-14393

【例題】 構造物ヘルスマモニタリング

構造物のモデルにおいて、次の条件を変化させて幾つかの条件を作り出した。

- 各階への重量 (Mass)
- 柱の硬さの低減 (Stiffness reduction)

State	Description
#1	Baseline condition
#2	Mass = 1.2 kg at the base
#3	Mass = 1.2 kg on the 1st floor
#4	87.5% stiffness reduction in column 1BD
#5	87.5% stiffness reduction in column 1AD and 1BD
#6	87.5% stiffness reduction in column 2BD
#7	87.5% stiffness reduction in column 2AD and 2BD
#8	87.5% stiffness reduction in column 3BD
#9	87.5% stiffness reduction in column 3AD and 3BD

上記 9 パターンのそれぞれで 10 回の実験を繰り返し、合計 90 回の実験を行ったものとします

【例題】 構造物ヘルスマモニタリング

State #1 を正常データとして学習させて、異常データが検知できるか調べたい

State	Description
#1	Baseline condition
#2	Mass = 1.2 kg at the base
#3	Mass = 1.2 kg on the 1st floor
#4	87.5% stiffness reduction in column 1BD
#5	87.5% stiffness reduction in column 1AD and 1BD
#6	87.5% stiffness reduction in column 2BD
#7	87.5% stiffness reduction in column 2AD and 2BD
#8	87.5% stiffness reduction in column 3BD
#9	87.5% stiffness reduction in column 3AD and 3BD

正常データ

異常データ

正常データを学習させる → 異常データを検出できるか？

SHMTools

Los Alamos National Laboratory

The screenshot shows the website for SHMTools at Los Alamos National Laboratory. The header includes the logos for Los Alamos National Laboratory and the National Security Education Center, along with a search bar. The main navigation menu has four items: EDUCATIONAL PROGRAMS, SHM DATA SETS AND SOFTWARE, RESEARCH PROJECTS, and PUBLICATIONS. The current page is titled "SHM Data Sets and Software" and includes a breadcrumb trail: NSEC » Engineering Institute » SHM Data Sets and Software. The page content describes SHMTools as a MATLAB package for structural health monitoring. It also features a "CONTACT" section for the Institute Director, Charles Farrar, and a "SHMTools Software (0.3.0 Beta)" section describing the package's functionality. A "Related Links" section on the right provides links to "Getting Started with SHMTools (pdf)" and "mFUSE Help Manual (pdf)".

Los Alamos National Laboratory National Security Education Center search site

Engineering Institute

Addressing national needs by fostering specialized recruiting and strategic partnerships

EDUCATIONAL PROGRAMS SHM DATA SETS AND SOFTWARE RESEARCH PROJECTS PUBLICATIONS

NSEC » [Engineering Institute](#) » [SHM Data Sets and Software](#)

SHM Data Sets and Software

SHMTools is a MATLAB package that facilitates the construction of structural health monitoring (SHM) processes.

CONTACT
Institute Director
Charles Farrar
(505) 665-0860
[Email](#)

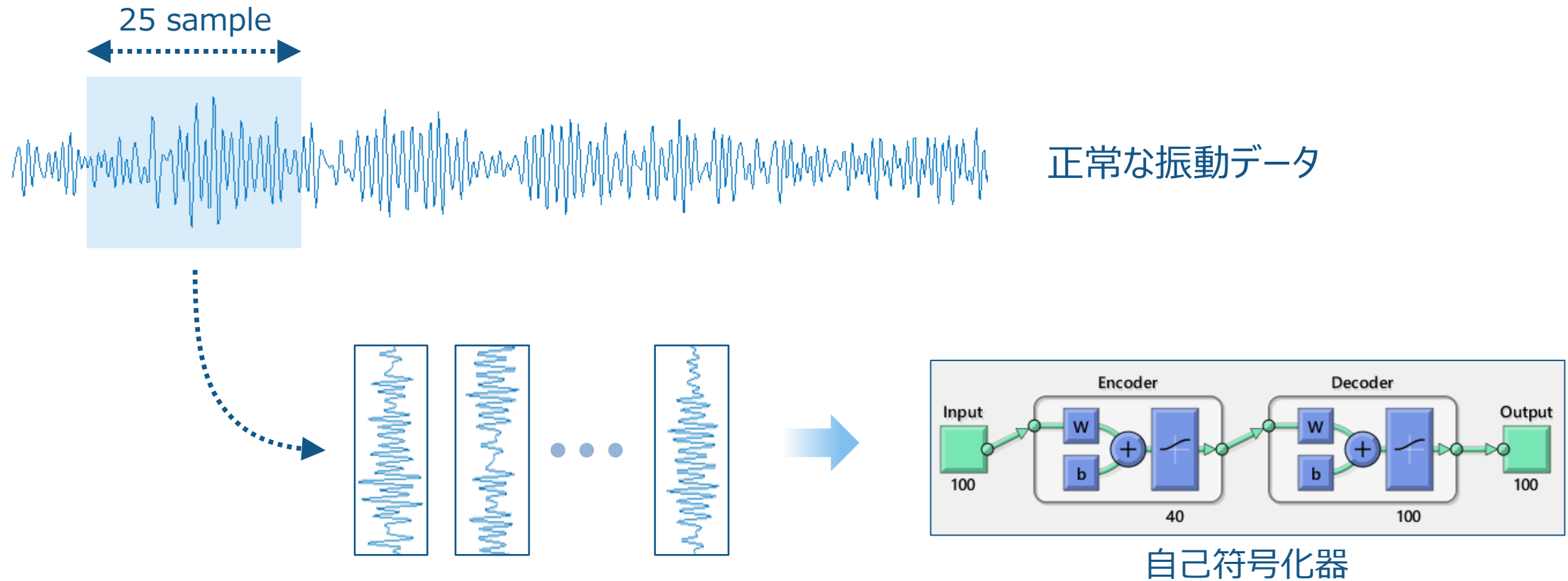
SHMTools Software (0.3.0 Beta)
The package provides a set of functions organized into modules according to the three primary stages of Structural Health Monitoring: Data Acquisition, Feature Extraction, and Feature Classification. A modular function design and set of standardized parameter formats make it easy to assemble and test customized SHM processes.

Related Links
[Getting Started with SHMTools \(pdf\)](#)
[mFUSE Help Manual \(pdf\)](#)

<http://www.lanl.gov/projects/national-security-education-center/engineering/software/shm-data-sets-and-software.php>

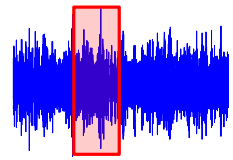
自己符号化器による異常検出

正常な振動データから部分時系列を生成して、自己符号化器に学習させる

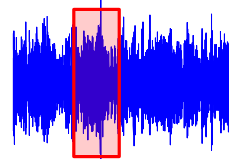


自己符号化器による異常検出

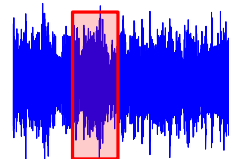
正常な場合のデータを自己符号化器に学習させる



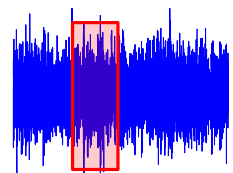
Sensor #1



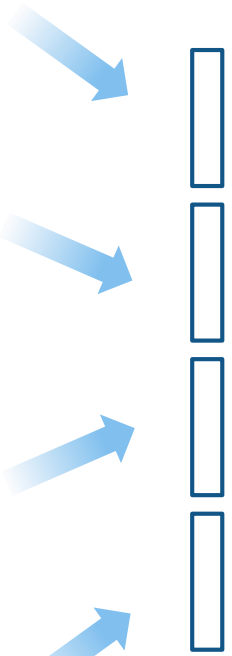
Sensor #2



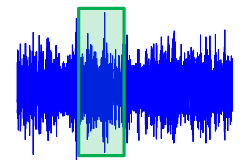
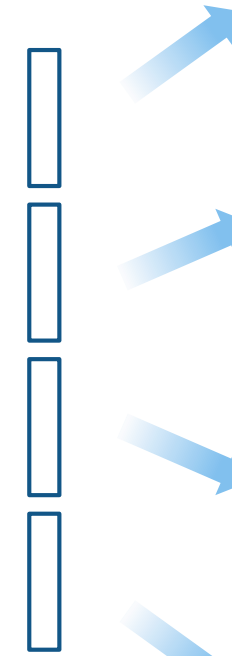
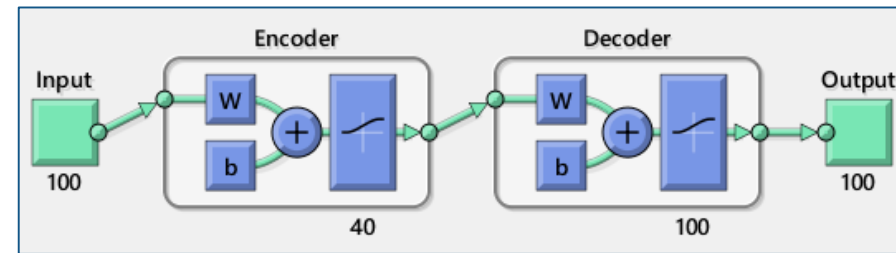
Sensor #3



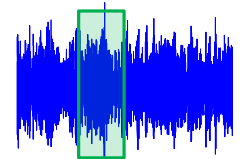
Sensor #4



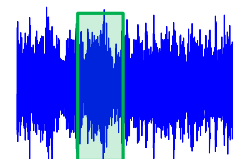
自己符号化器



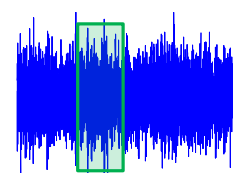
Sensor #1



Sensor #2



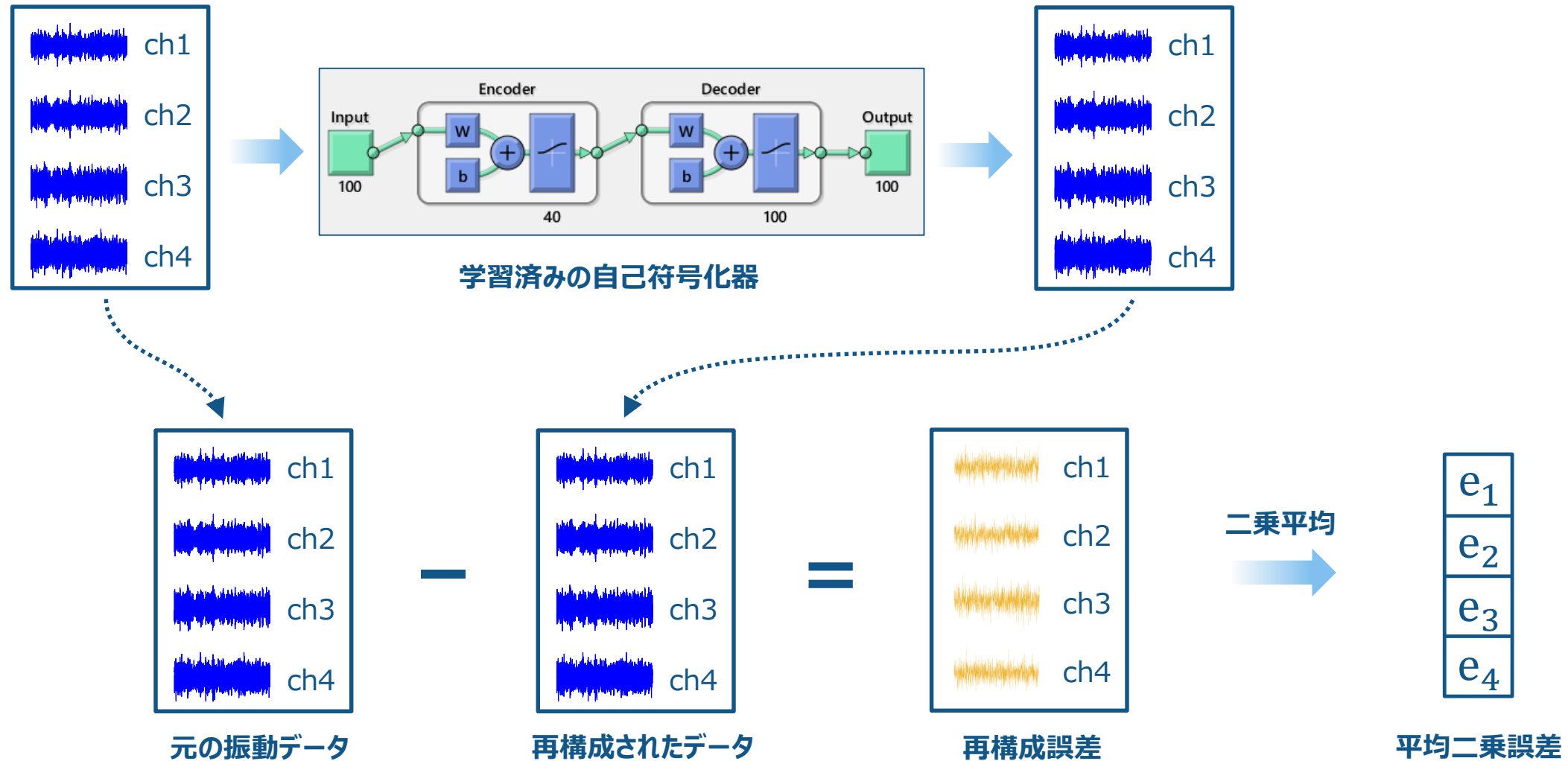
Sensor #3



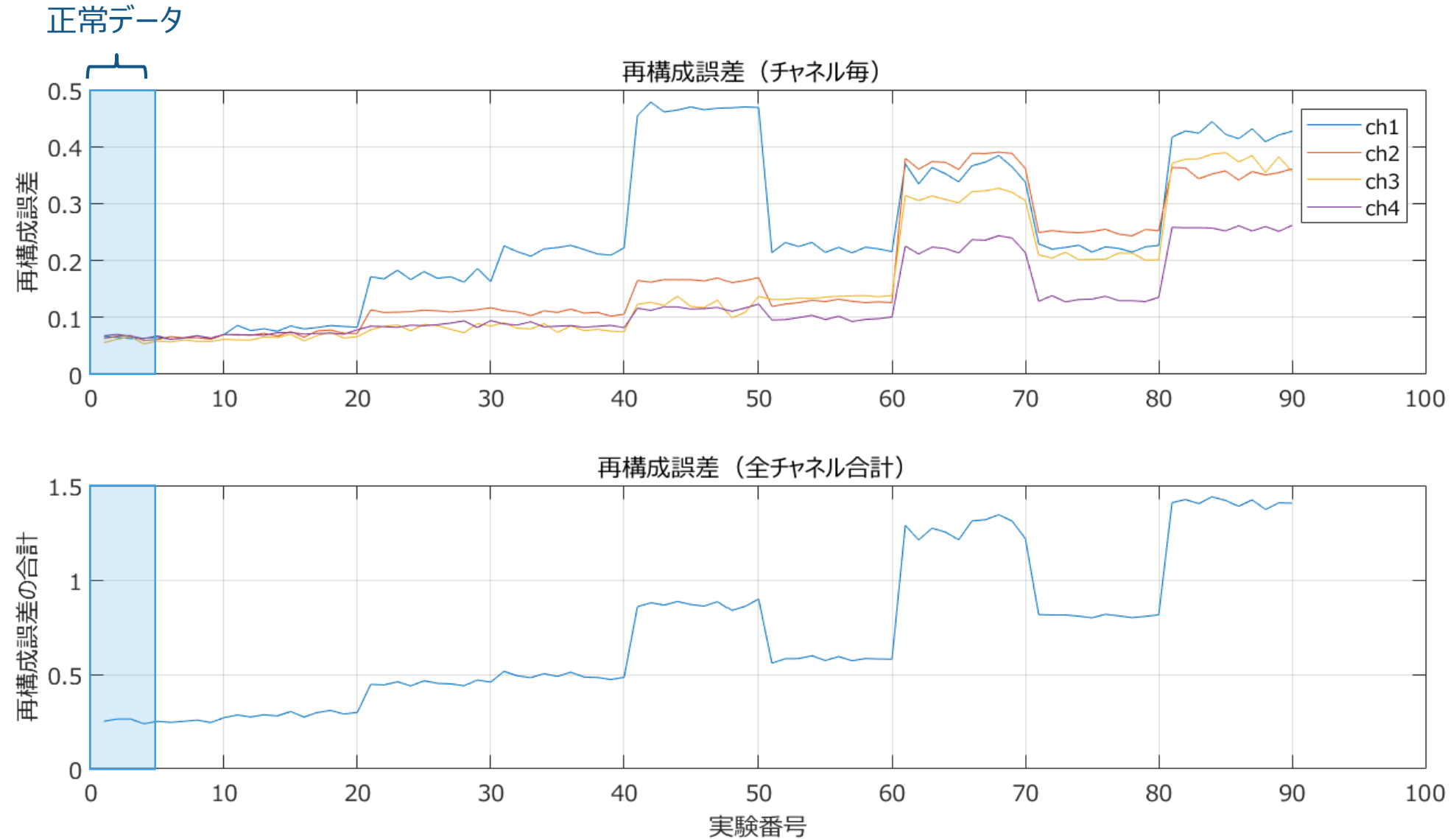
Sensor #4

自己符号化器による異常検出

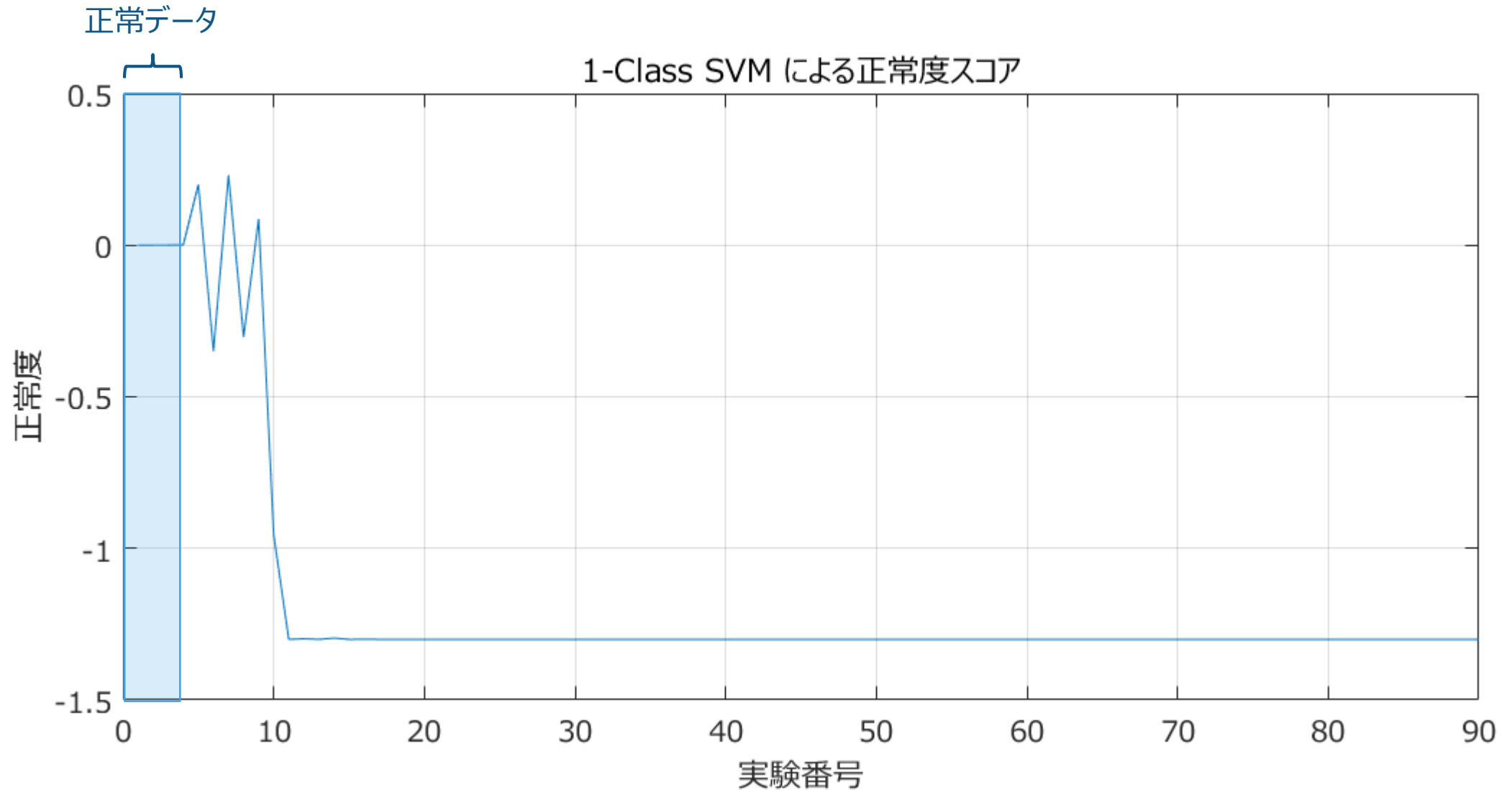
異常を含む可能性のあるデータを自己符号化器に通して、再構成誤差を算出する



自己符号化器による異常検出



1-Class SVM による異常検知

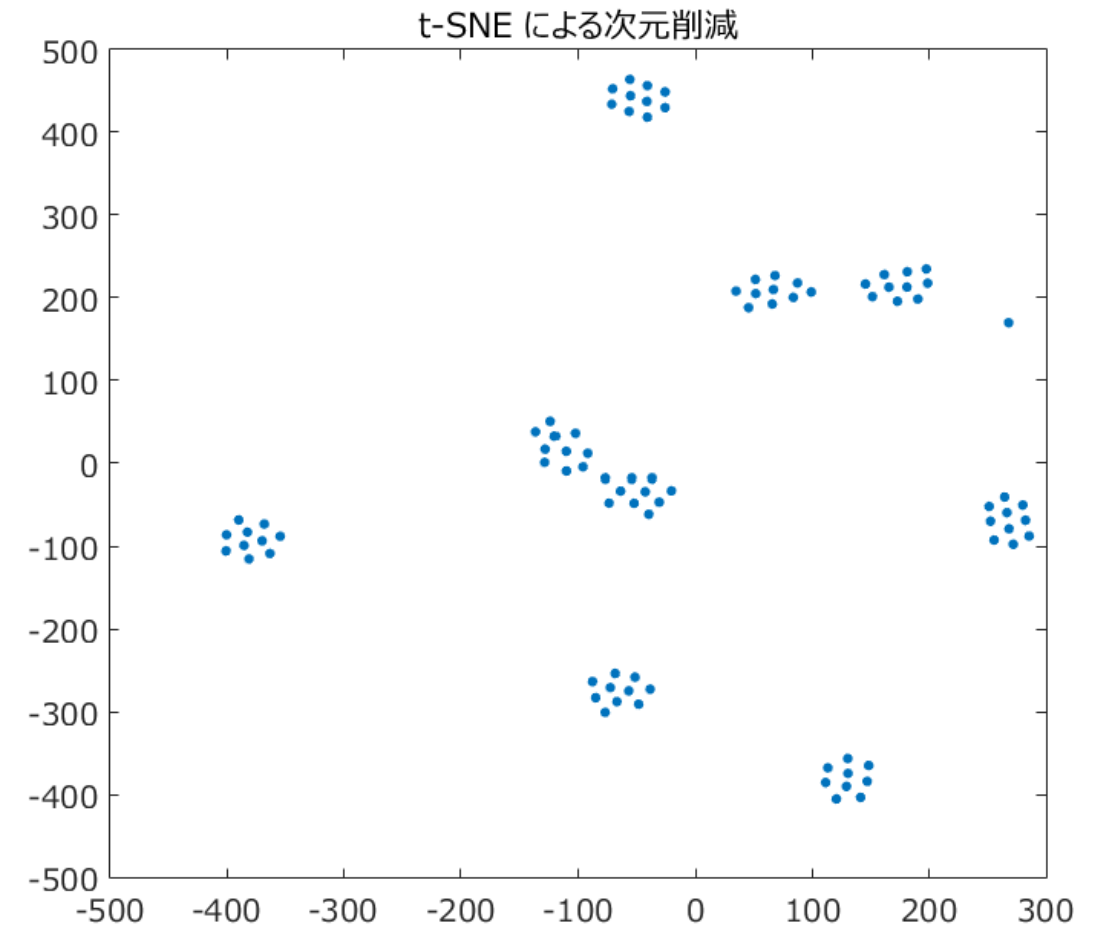
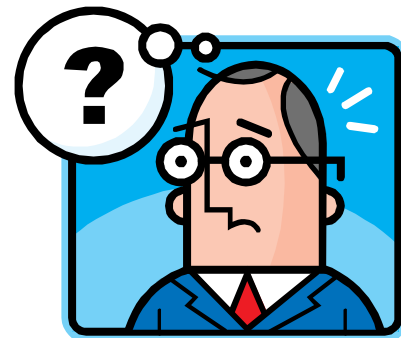


t-SNE による異常データのクラスタリング

検出された異常の種類を識別することはできるだろうか？

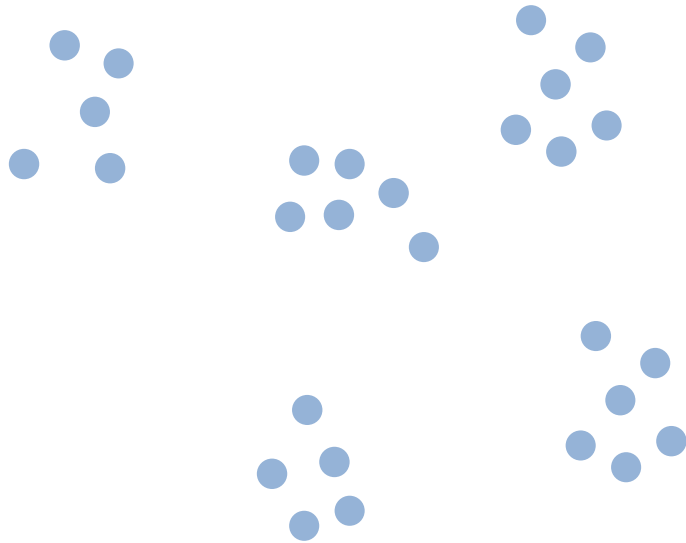
State	Description
#1	Baseline condition
#2	Mass = 1.2 kg at the base
#3	Mass = 1.2 kg on the 1st floor
#4	87.5% stiffness reduction in column 1BD
#5	87.5% stiffness reduction in column 1AD and 1BD
#6	87.5% stiffness reduction in column 2BD
#7	87.5% stiffness reduction in column 2AD and 2BD
#8	87.5% stiffness reduction in column 3BD
#9	87.5% stiffness reduction in column 3AD and 3BD

どのような異常の種類があるだろうか？



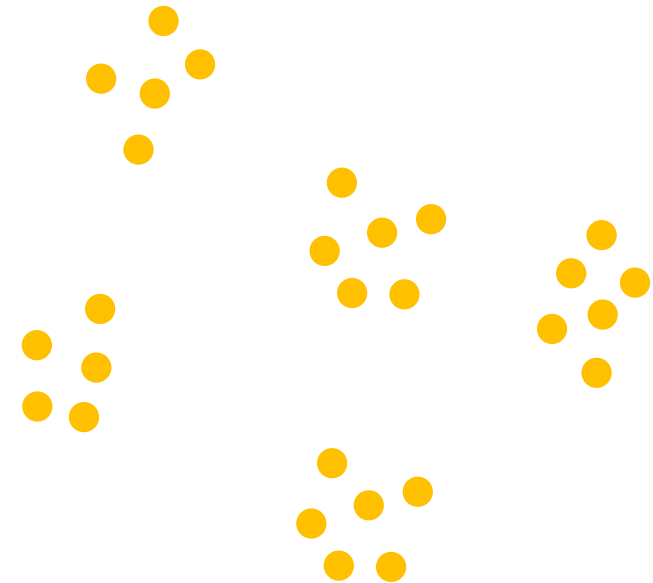
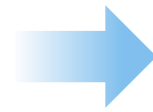
t-SNE とは？

直接目で見ることのできない高次元のデータを低次元にマップする手法（次元削減手法）



高次元空間（特徴量等）

t-SNE



低次元空間（2次元または3次元）

t-SNE とは？

「あるデータ点はその近傍（参照点）を選ぶ」という状況を考える（KNN分類器をイメージ）

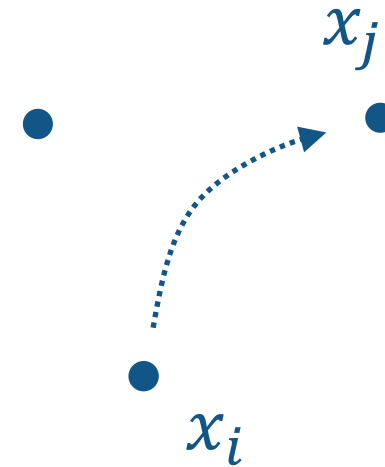
➡ KNN の場合は一番近い点を確定的に選ぶだけだが、もし「近傍」が確率的に選ばれるとすると…

$$P(j|i) \propto \frac{1}{\sqrt{2\sigma^2}} \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)$$

「近傍」を選択する確率が
ガウス分布に従っていると！

正規化

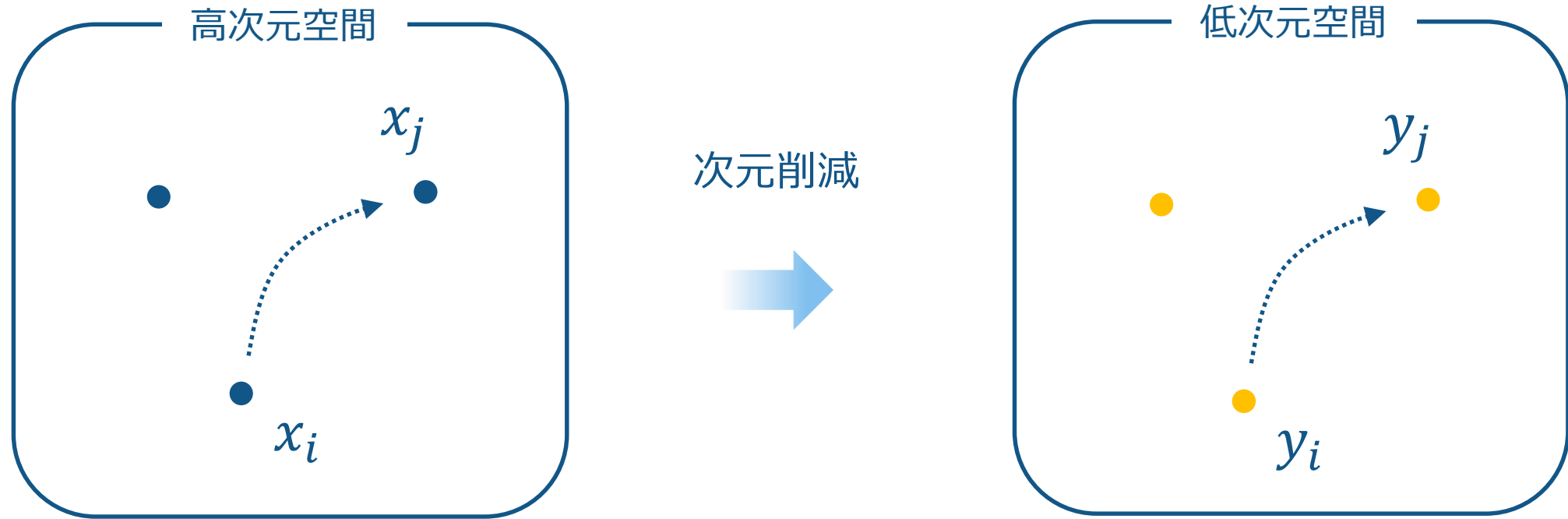
$$P(j|i) = \frac{\exp(-\|x_i - x_j\|^2/2\sigma^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma^2)}$$



この条件付き確率があまり壊れないように次元削減を行うのが SNE

t-SNE とは？

高次元のデータ点 x_1, x_2, x_3, \dots を低次元のデータ点 y_1, y_2, y_3, \dots へマッピングしたい



$$P_i(j) = P(j | i) = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma^2)}$$

$$Q_i(j) = Q(j | i) = \frac{\exp(-\|y_i - y_j\|^2 / 2\sigma^2)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2 / 2\sigma^2)}$$

t-SNE とは？

次元削減の前と後で、次の条件付き確率があまり変化しないように変換を行いたい

$$P_i(j) = P(j | i) = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma^2)}$$

$$Q_i(j) = Q(j | i) = \frac{\exp(-\|y_i - y_j\|^2 / 2\sigma^2)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2 / 2\sigma^2)}$$

確率分布の違いを
KL-ダイバージェンスで評価

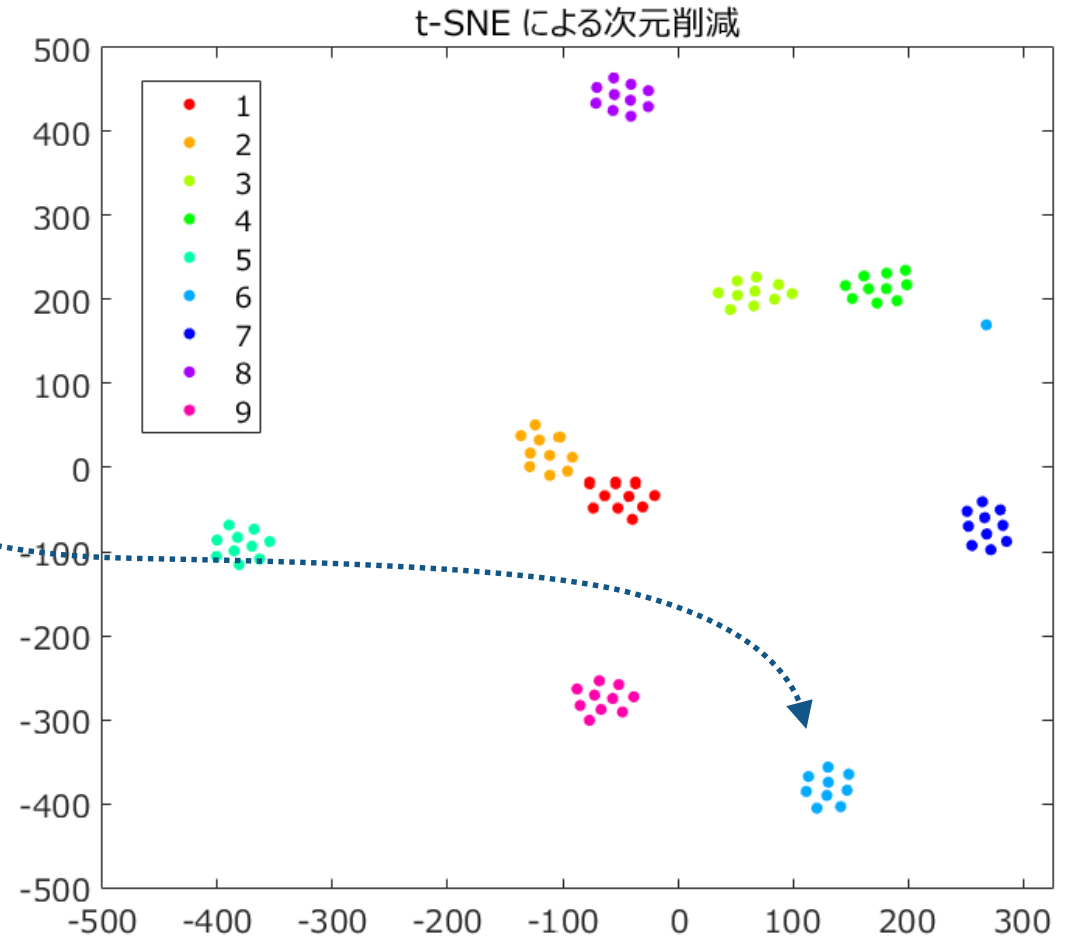
次の損失関数 C を最小化するように次元削減を行うのが SNE（改良版が t-SNE）

$$C = \sum_i KL(P_i | Q_i) \quad \leftarrow \dots \text{全ての } x_i \text{ に関する KL-ダイバージェンスの和}$$

t-SNE による異常データのクラスタリング

実際の異常の種類で色分けすると、次のような可視化を行うことができる

State	Description
#1	Baseline condition
#2	Mass = 1.2 kg at the base
#3	Mass = 1.2 kg on the 1st floor
#4	87.5% stiffness reduction in column 1BD
#5	87.5% stiffness reduction in column 1AD and 1BD
#6	87.5% stiffness reduction in column 2BD
#7	87.5% stiffness reduction in column 2AD and 2BD
#8	87.5% stiffness reduction in column 3BD
#9	87.5% stiffness reduction in column 3AD and 3BD



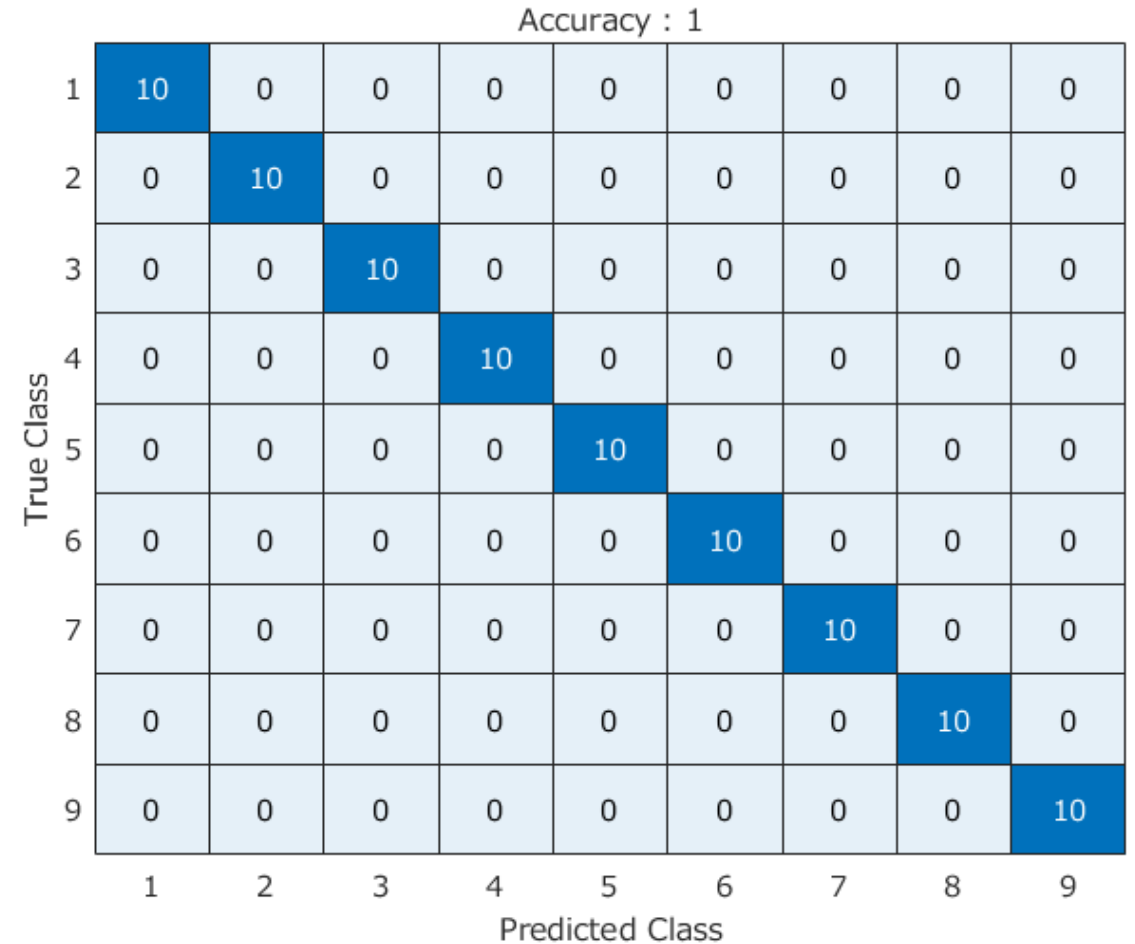
それぞれの異常がひとつの島に対応しているらしいな！



KNN による分類

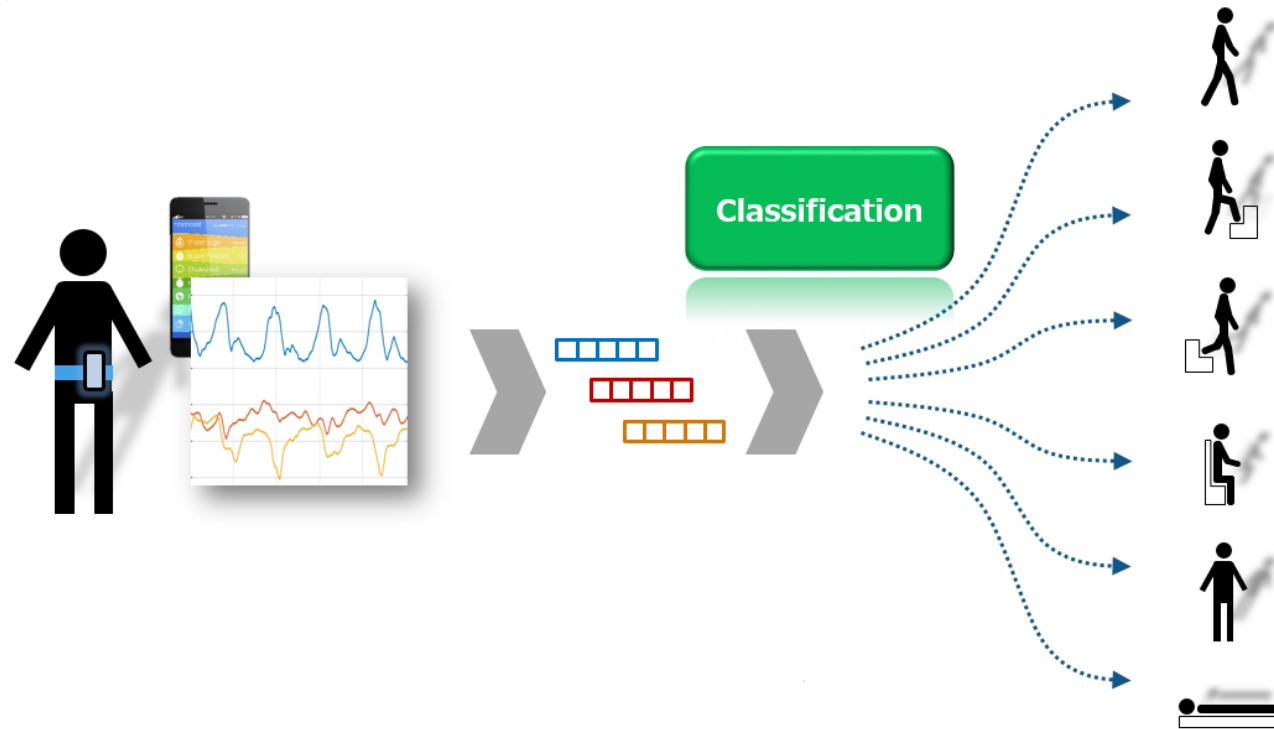
再構成誤差から生成した特徴量を KNN で分類してみると、次のような結果を得る

State	Description
#1	Baseline condition
#2	Mass = 1.2 kg at the base
#3	Mass = 1.2 kg on the 1st floor
#4	87.5% stiffness reduction in column 1BD
#5	87.5% stiffness reduction in column 1AD and 1BD
#6	87.5% stiffness reduction in column 2BD
#7	87.5% stiffness reduction in column 2AD and 2BD
#8	87.5% stiffness reduction in column 3BD
#9	87.5% stiffness reduction in column 3AD and 3BD

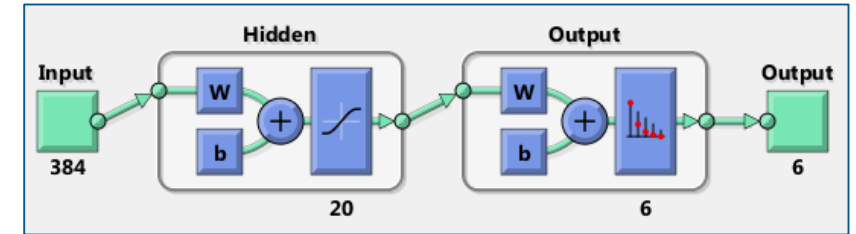


【例題】人の活動状態の識別

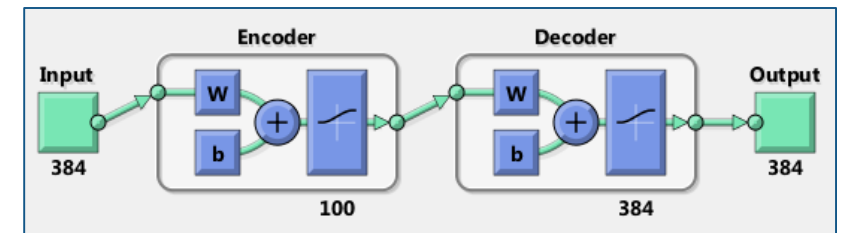
ニューラルネットを分類と次元削減の両方に利用



部分時系列からシンプルに分類



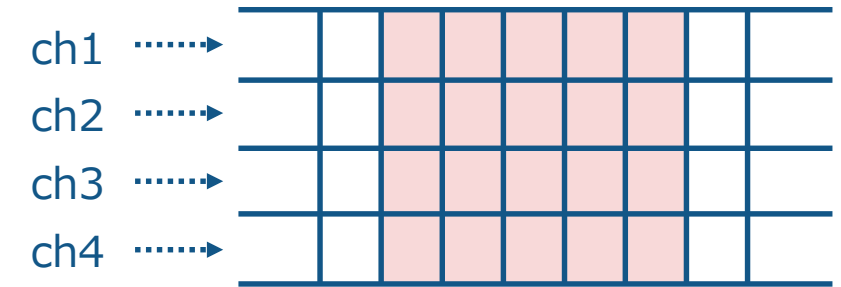
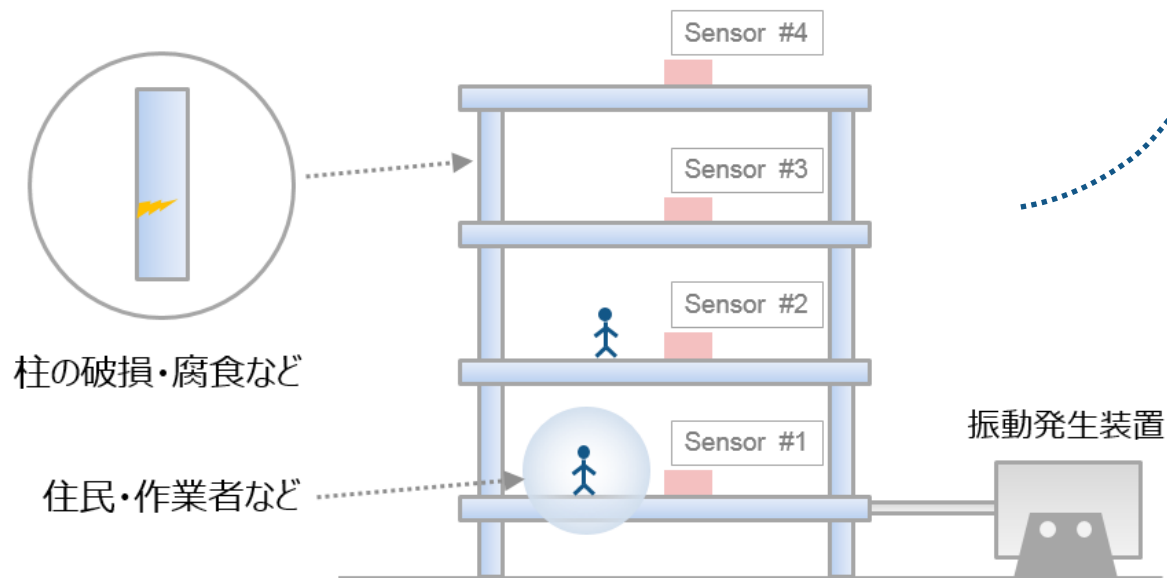
エンコーダ部分を次元削減に利用



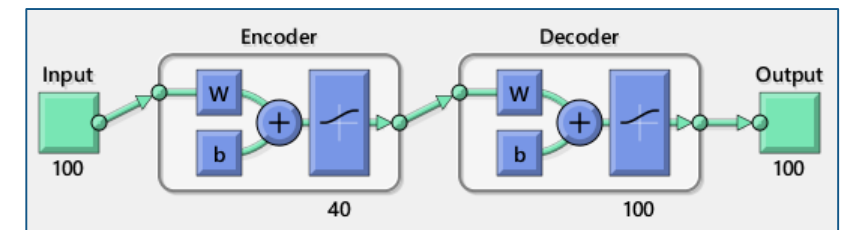
➡ 元々のデータで分類がうまくいくようなケースでは次元削減で性能向上が狙える可能性が！

【例題】構造ヘルスマニタリング

正常なセンサーデータが持つ「データの関係性」をモデル化



部分時系列として切り出される
データの関係性をモデル化する



➡ 「データの関係性」の崩れを「再構成誤差」として検出することが可能となる！

センサーデータ解析のためのニューラルネット

人の活動状態の識別

- 部分時系列をそのままニューラルネットワークへ
- 自己符号化器を非線形な次元削減に利用
- 積み上げることで積層自己符号化器を構築

構造ヘルスマニタリング

- 正常なセンサーデータを自己符号化器へ
- 再構成信号との誤差により異常を検出
- 異常の種別等も次元削減に解析可能に



Neural Network Toolbox

- パターンネット (分類)
- 積層自己符号化器 (分類)
- 自己符号化器 (異常検出)

Statistics and Machine Learning Toolbox

- t-SNE (次元削減)
- 1-class SVM (異常検出)
- KNN (分類器)

従来手法との関連性

従来の異常検知手法との関連性

主成分分析による異常検知

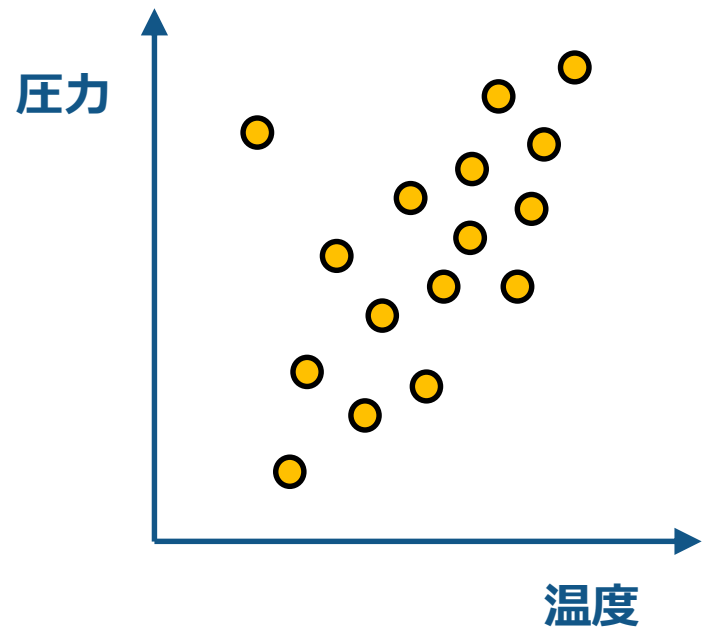
- 多変量統計的プロセス管理（MSPC）等の分野で有名
- 主成分空間からの距離を異常度として異常検出

自己回帰モデルによる異常検知

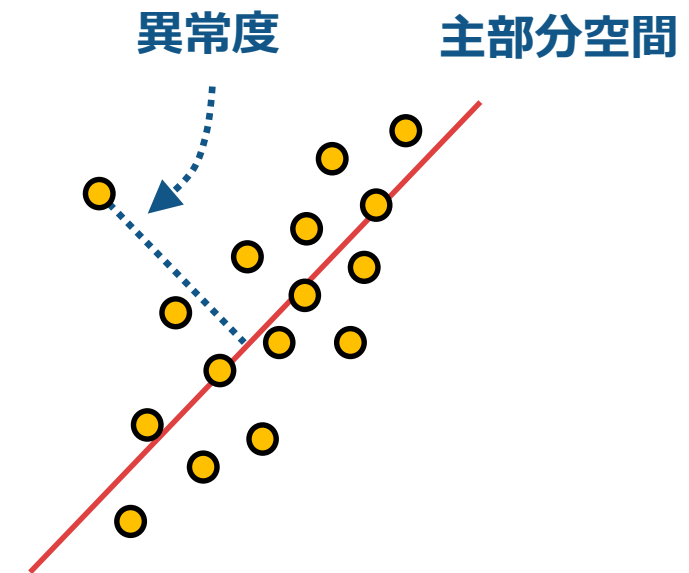
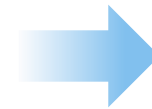
- 時系列信号の異常検出に利用される
- 自己回帰モデルにより1ステップ先を予測
- 予測値との差分により異常検出

主成分分析による異常検知

- 多変量統計的プロセス管理（MSPC）等の分野で有名
- 主成分空間からのズレを異常度として定義



PCA



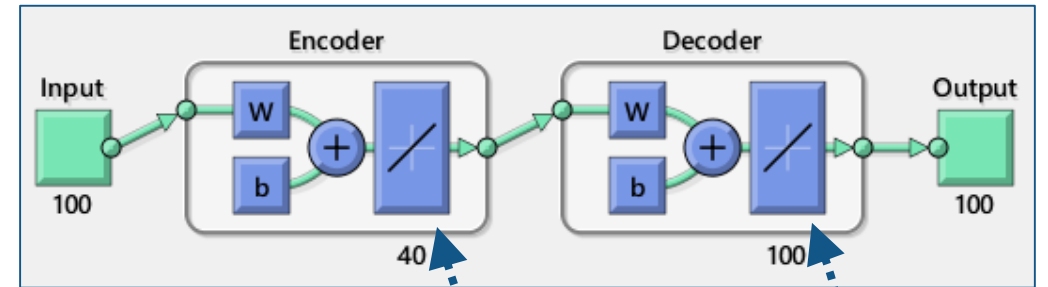
主成分分析による異常検知

主部分空間からの距離

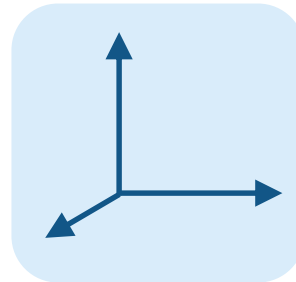
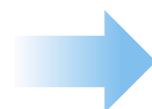
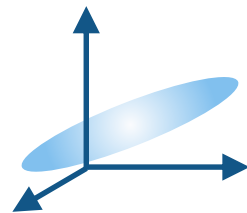
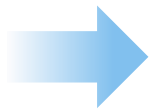
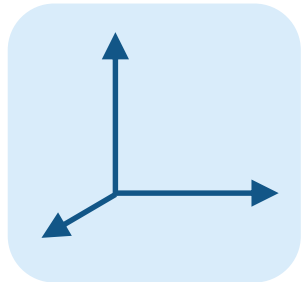
=

再構成信号と元信号の距離

- 線形な Autoencoder は主成分分析に一致する
- Encoder 部分 → 主部分空間への射影
- Decoder 部分 → 元の空間への埋め込み



主部分空間への射影



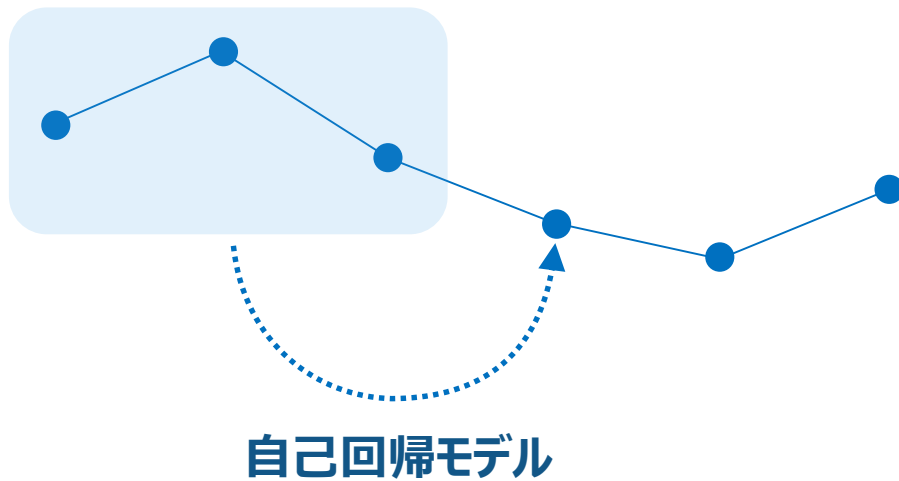
元の空間への埋め込み

活性化関数を線形な関数へ置き換える

ARモデルによる異常検知との関連

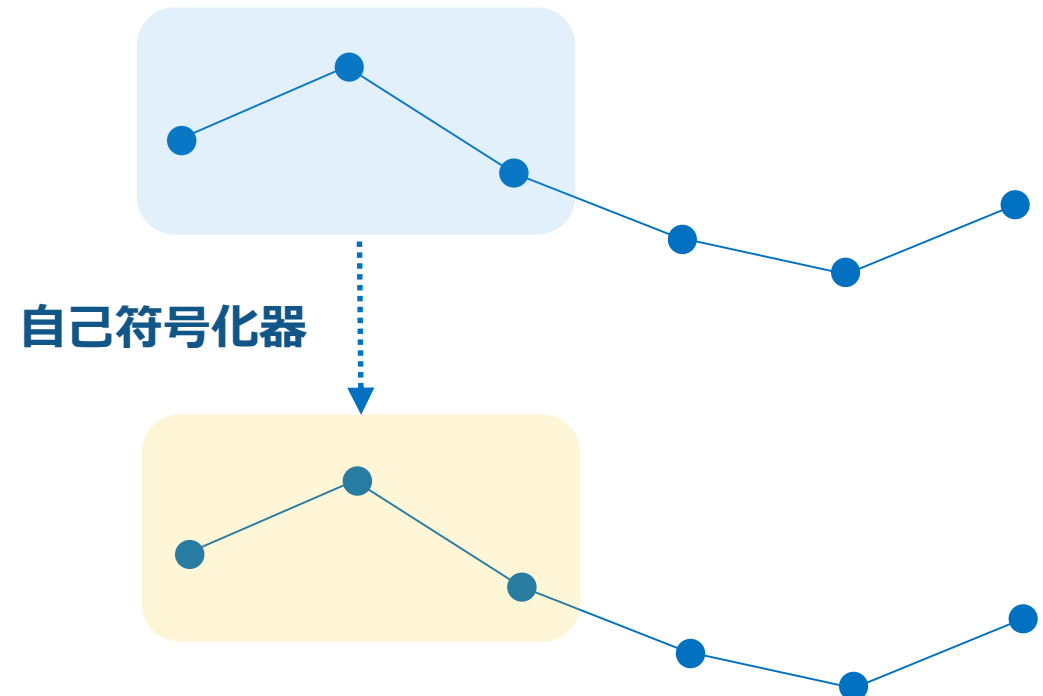
自己回帰モデルによる異常検知

- ARモデルにより1ステップ先を予測
- 予測との誤差を使って異常検出



自己符号化器による異常検知

- 自己符号化器により時系列データを再構成
- 再構成したものとの誤差を使って異常検出



システムへの展開・Cコード生成

センサーデータ解析システムにおける課題

■ 端末の問題

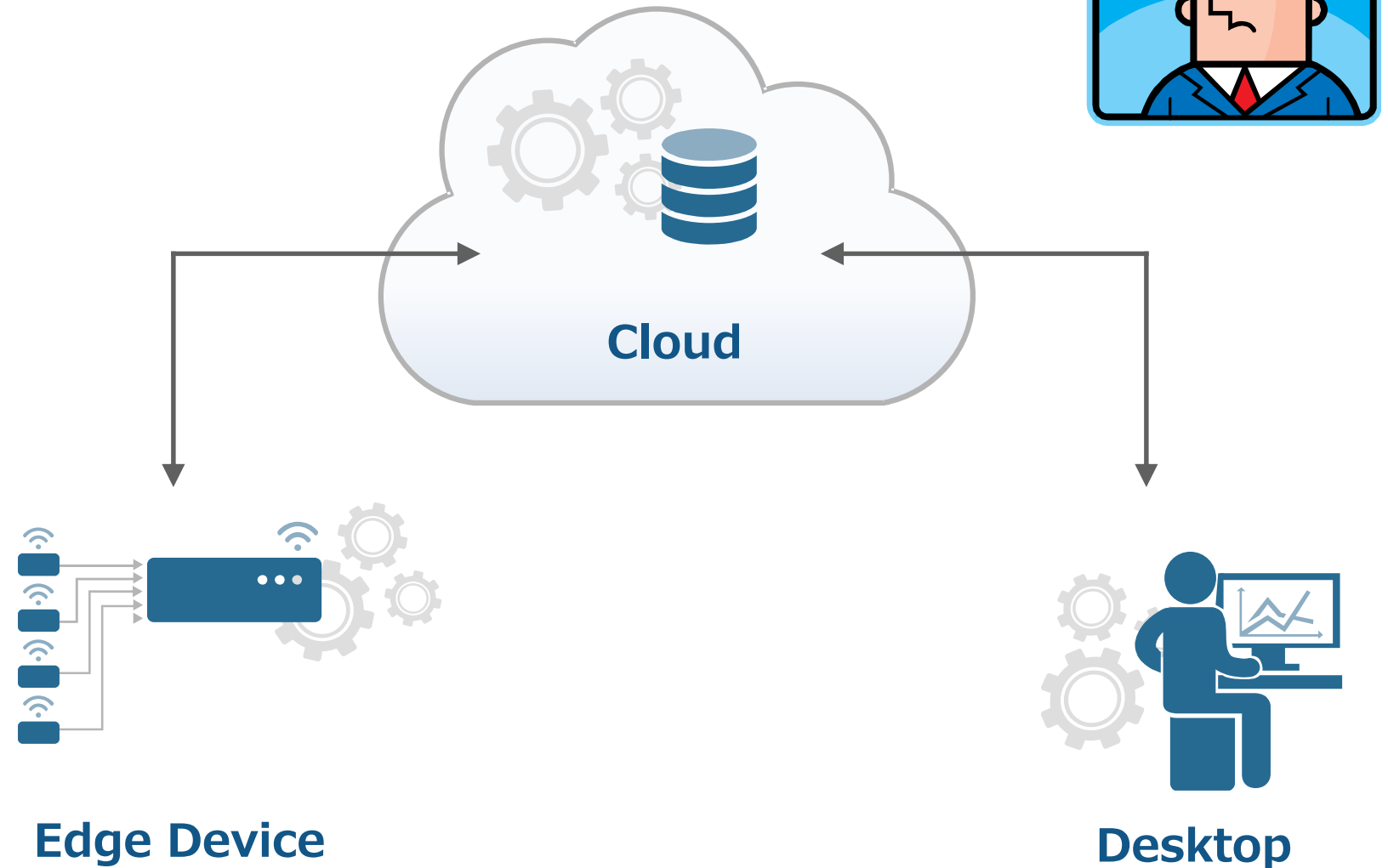
- ・ データを全部送るか？
- ・ 特徴量のみを送るか？
- ・ 異常なデータのみ送るか？
- ・ 通信モジュールの消費電力は？
- ・ 演算速度と消費電力は？

■ インフラの問題

- ・ 必要な帯域は確保されているか？
- ・ ネットワークは安定しているか？
- ・ 電力供給は安定しているか？
- ・ 通信量に対するコストは？

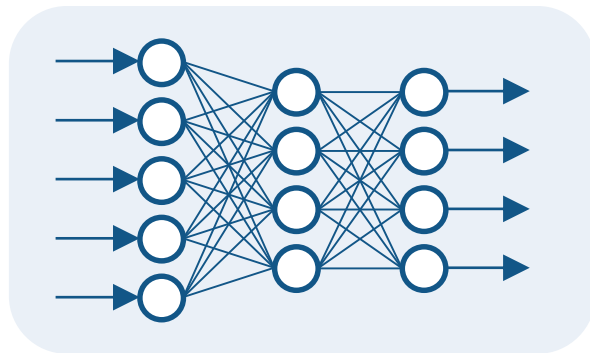
■ 開発側の問題

- ・ アルゴリズムの検証し易さ
- ・ 組み込みコードの開発コスト
- ・ サーバ側コードの開発コスト



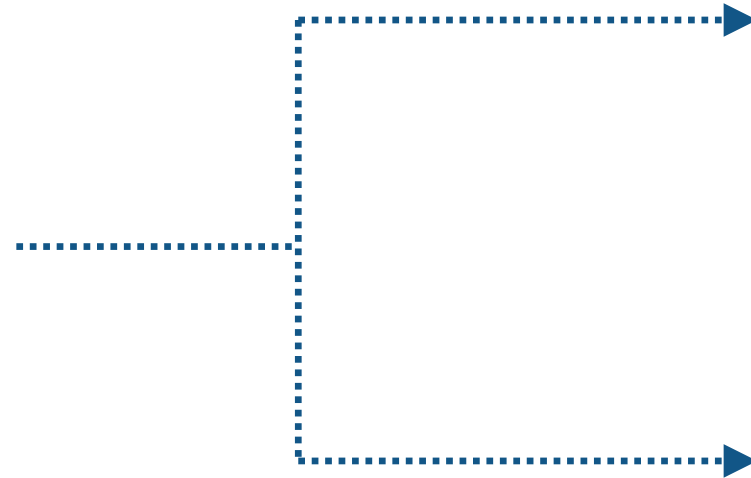
学習済み自己符号化器の活用

- Simulink ブロック への変換
- MATLAB 関数への変換

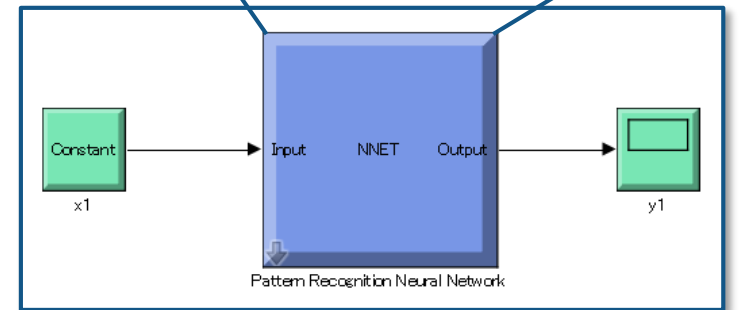
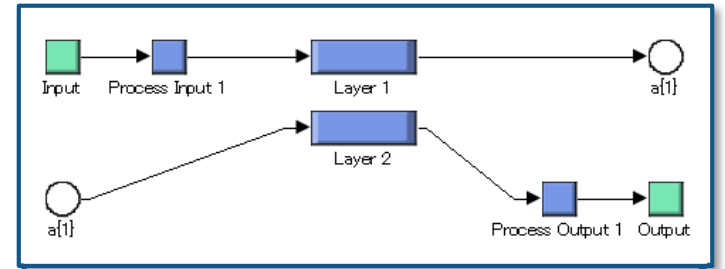


学習済み自己符号化器

Simulinkブロック生成
(関数 : generateSimulink)



MATLABコード生成
(関数 : generateFunction)



Simulinkブロック

```

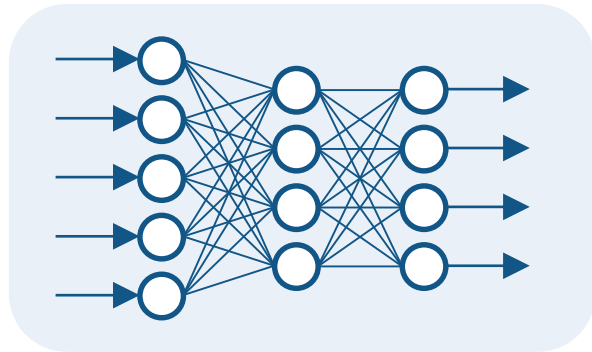
1 function [Y,Xf,Af] = neural_function(X,~,~)
2 %NEURAL_FUNCTION neural network simulation function.
3 %
4 % Generated by Neural Network Toolbox function genFunction, 08-Jul-2016 15:06:58.
5 %
6 % [Y] = neural_function(X,~,~) takes these arguments:
7 %
8 % X = 1xTS cell, 1 inputs over TS timesteps
9 % Each X{1,ts} = 4xQ matrix, input #1 at timestep ts.
10 %
11 % and returns:
12 % Y = 1xTS cell of 1 outputs over TS timesteps.
13 % Each Y{1,ts} = 3xQ matrix, output #1 at timestep ts.

```

MATLAB関数

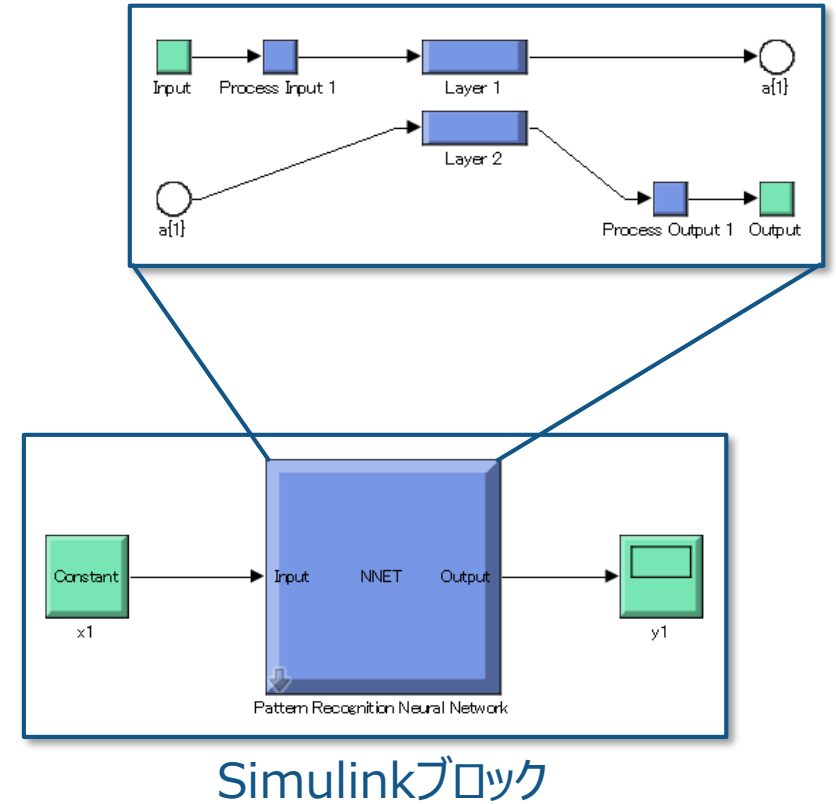
学習済みニューラルネットの活用

- Simulink ブロック への変換
- MATLAB 関数への変換



学習済みニューラルネット

Simulinkブロック生成
(関数 : gensim)



MATLABコード生成
(関数 : genFunction)

```

1 function [Y,Xf,Af] = neural_function(X,~,~)
2 %NEURAL_FUNCTION neural network simulation function.
3 %
4 % Generated by Neural Network Toolbox function genFunction, 08-Jul-2016 15:06:58.
5 %
6 % [Y] = neural_function(X,~,~) takes these arguments:
7 %
8 % X = 1xTS cell, 1 inputs over TS timesteps
9 % Each X{1,ts} = 4xQ matrix, input #1 at timestep ts.
10 %
11 % and returns:
12 % Y = 1xTS cell of 1 outputs over TS timesteps.
13 % Each Y{1,ts} = 3xQ matrix, output #1 at timestep ts.
    
```

MATLAB関数

C コード生成対応（学習済みモデルによる予測部分）

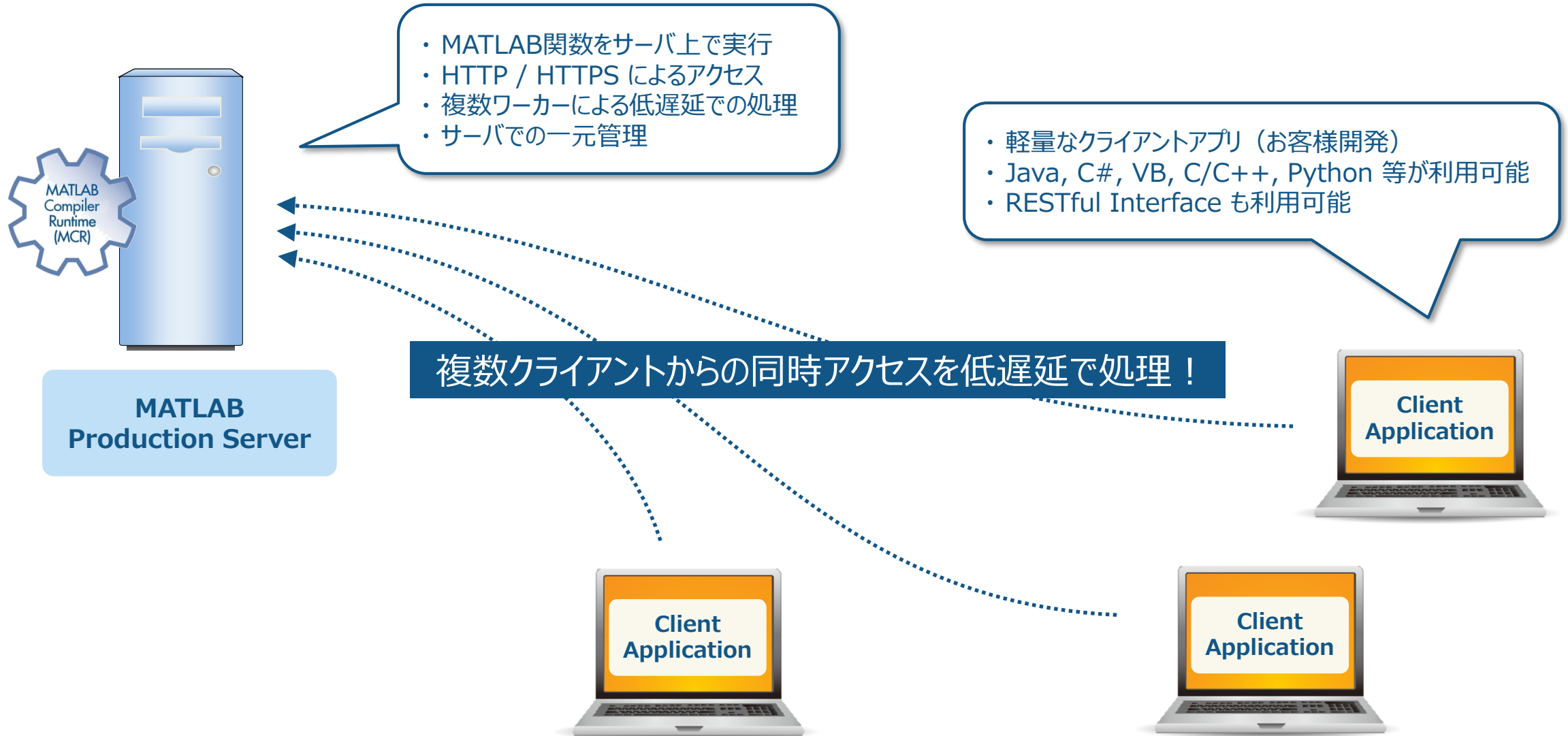
R2016b

- サポートベクターマシンによる分類 (fitcsvm)
- 線形モデルによる 2 クラス分類 (fitclinear)
- 線形モデルによる多クラス分類 (fitcecoc)

R2017a

- 線形回帰 (fitlm)
- 一般化線形モデルによる回帰 (fitglm, glmval)
- 決定木による分類・回帰 (fitctree, fitrtree)
- アンサンブル学習器（決定木）による分類 (fitcensemble)

MATLAB Production Server™

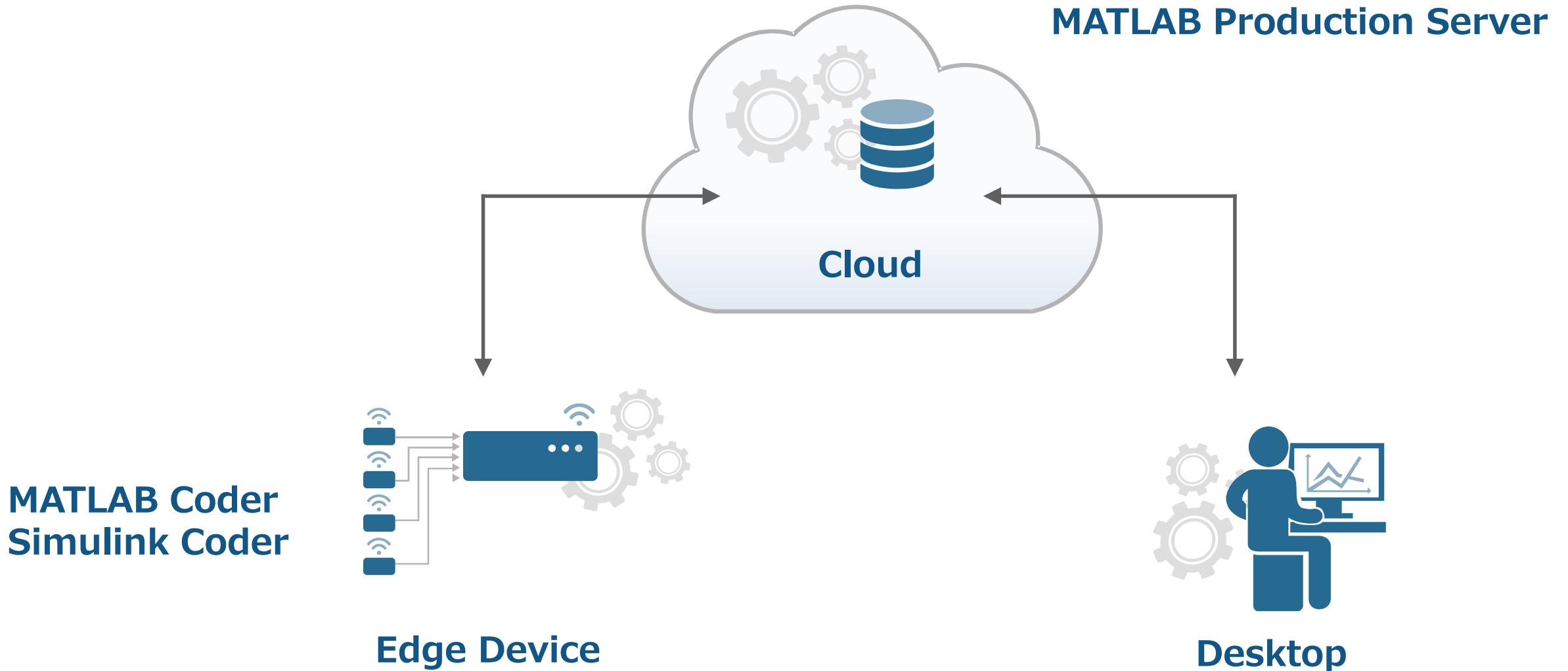


- MATLAB関数をサーバ上で実行
- HTTP / HTTPS によるアクセス
- 複数ワーカーによる低遅延での処理
- サーバでの一元管理

- 軽量なクライアントアプリ（お客様開発）
- Java, C#, VB, C/C++, Python 等が利用可能
- RESTful Interface も利用可能

複数クライアントからの同時アクセスを低遅延で処理！

MATLAB/Simulink によるセンサーデータ解析システムの構築



Questions?