埼玉大学産学連携協議会
第4回 データサイエンス技術研究会

時系列データの解析と産業応用

NTTコミュニケーションズ株式会社

2023-12-15

© Copyright NTT Communications Corporation All Rights Reserved.

準備-環境構築

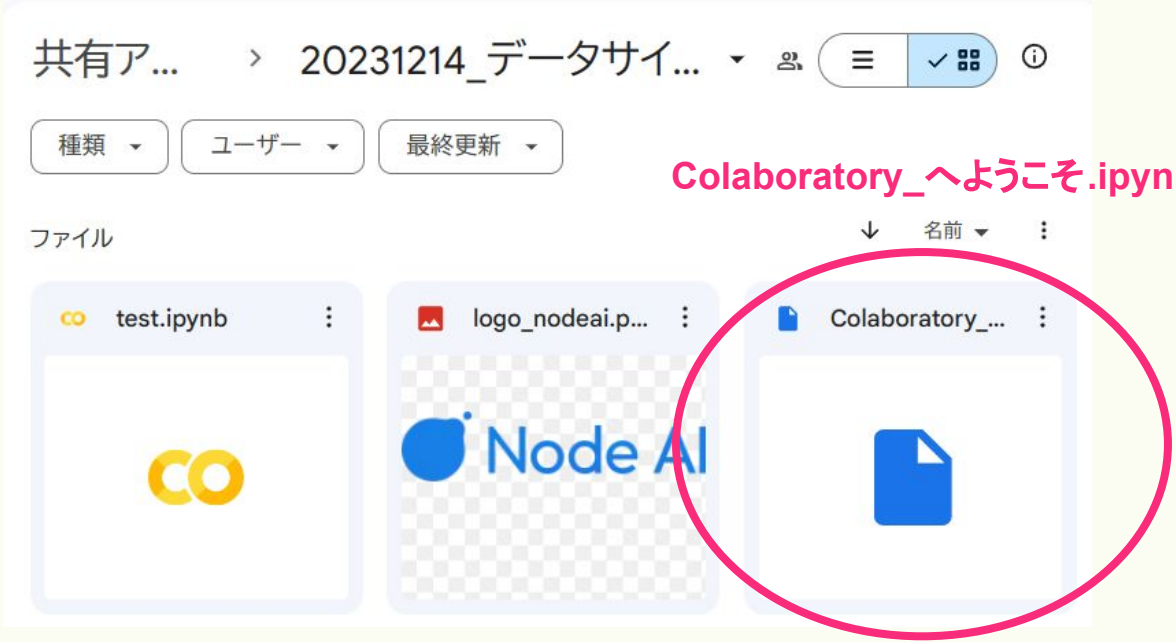
© Copyright NTT Communications Corporation All Rights Reserved.

資料やコードなど

- Google Drive
  - リンク:
    - https://drive.google.com/drive/folders/19TUFBINNeyVi45aUpe6yIsbdCoRkbvk5
- コードの実行環境: Google Colaboratory
  - Googleの提供するPython（JupyterNotebook環境）
  - 無料である程度使える。
  - 途中保存したファイルなどはしばらくたつと削除されるので注意

© Copyright NTT Communications Corporation All Rights Reserved.

Colaboratoryの動作確認



© Copyright NTT Communications Corporation All Rights Reserved.

Colaboratoryの動作確認

ご自身のドライブにコピー
コピーしなくても動作しますが、編集履歴は残りませんので、コピーしての編集をお勧めします。

© Copyright NTT Communications Corporation All Rights Reserved.

Colaboratoryの動作確認

計算実行

一括実行

© Copyright NTT Communications Corporation All Rights Reserved.

GPUの使用設定

- Colaboratory　デフォルト設定ではCPUを使うようになっている
- ディープラーニングの実行の際は、**GPUを有効**にしないと計算が終わらない

© Cop

AI・データ分析関連事業紹介

© Copyright NTT Communications Corporation All Rights Reserved.

NTTコミュニケーションズとは

NTT

総合ICT事業 docomo / NTT Communications / NTTコムウェア

- 携帯電話事業
- **県間通信サービス・国際通信事業**
- **ソリューション事業**
- **システム開発事業**

NTTコミュニケーションズの事業は**太字**
主に**企業・ビジネス向け**のソリューション・
サービスを担当

地域通信事業 NTT東日本 / NTT西日本

- 県内通信サービス

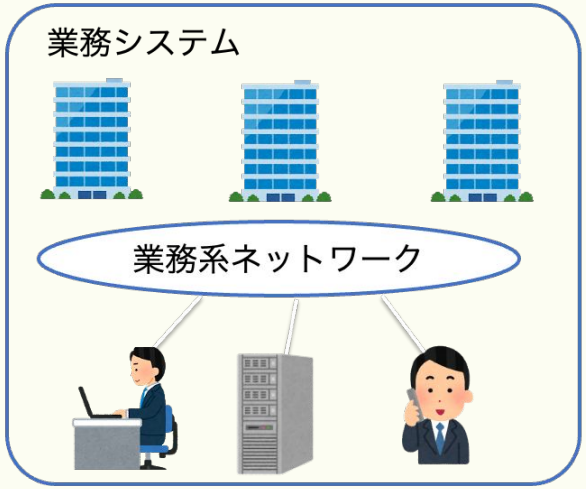
グローバル・ソリューション事業 NTT DATA

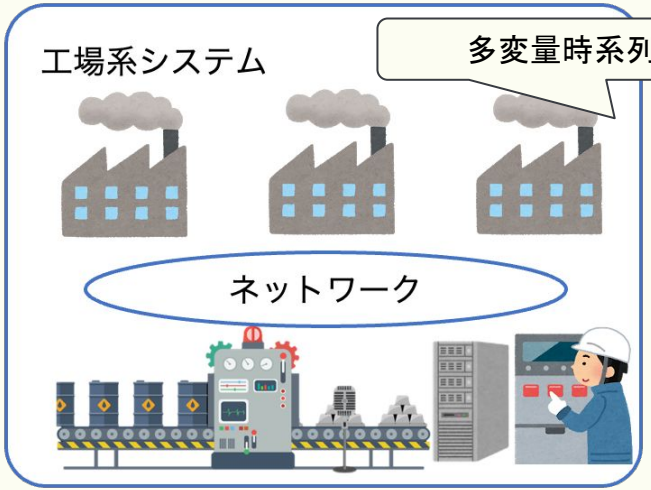
- コンサルティング
- システム・ソフトウェア開発
- ネットワークシステム・クラウド・グローバルデータセンター

© Copyright NTT Communications Corporation All Rights Reserved.

なぜデータ分析？なぜ時系列分析？

工場（ライン）におけるデータ活用需要の増大→AI・データ分析への期待





もともとは…
- 電話サービス
- ネットワークサービス
- システム構築

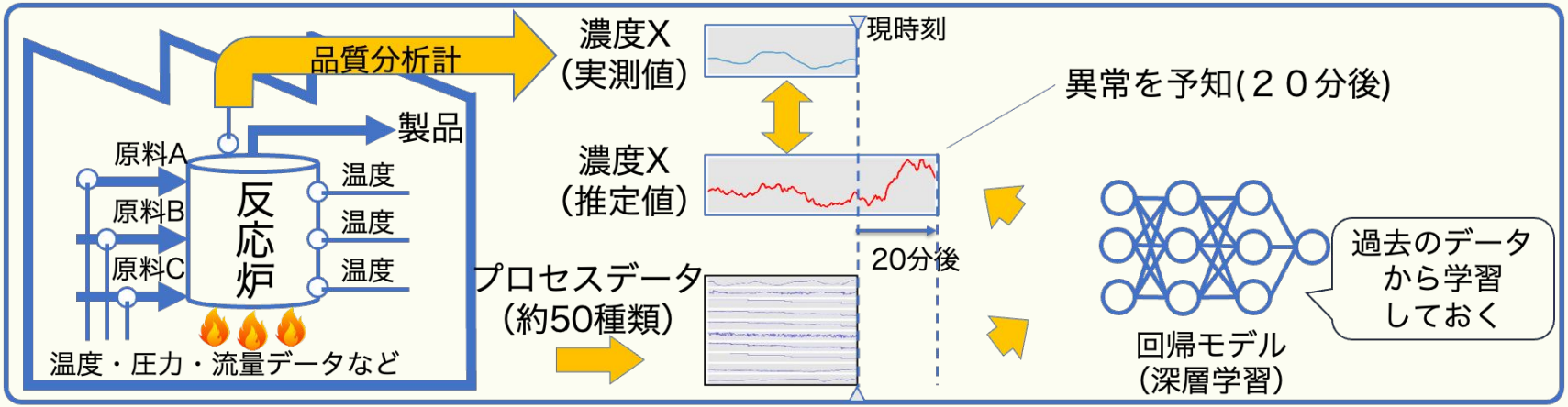
- 5G無線NW
- 複数拠点の統合
- データの有効活用（監視・分析・予測・制御）

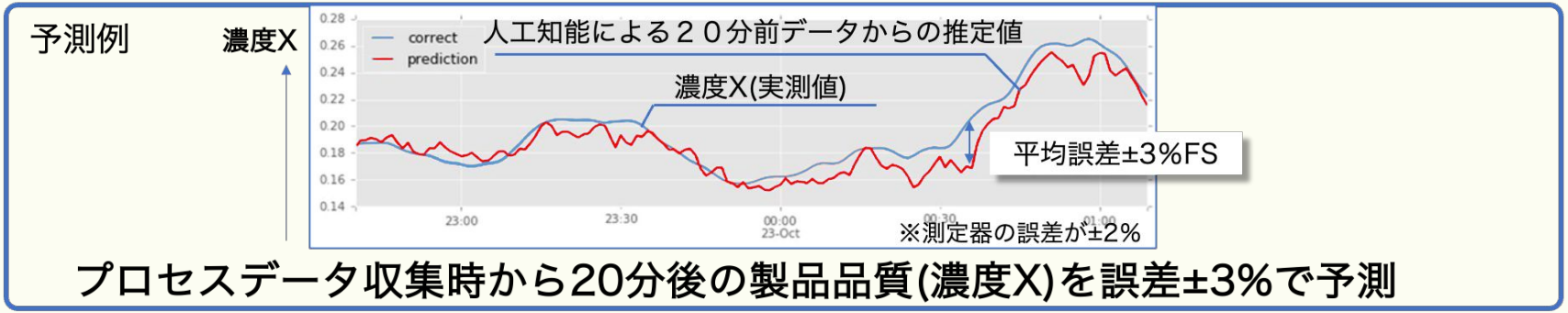
AI・データ分析への期待の高まり

Smart Data Platform（SDPF：docomo business サービス） https://www.ntt.com/business/sdpf.html

© Copyright NTT Communications Corporation All Rights Reserved.

事例1：化学プラントの品質予測



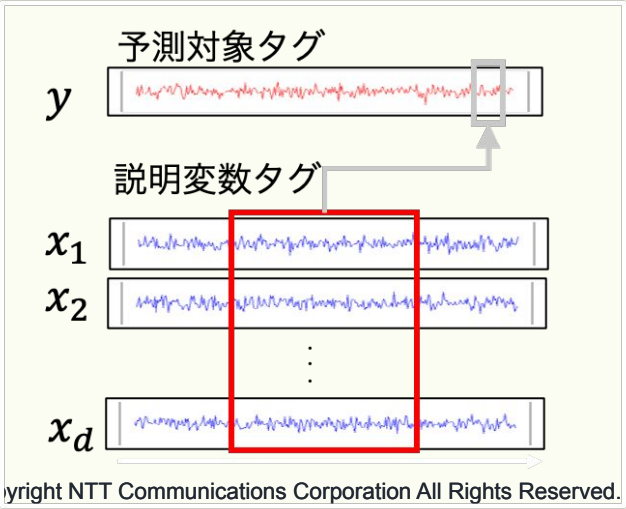


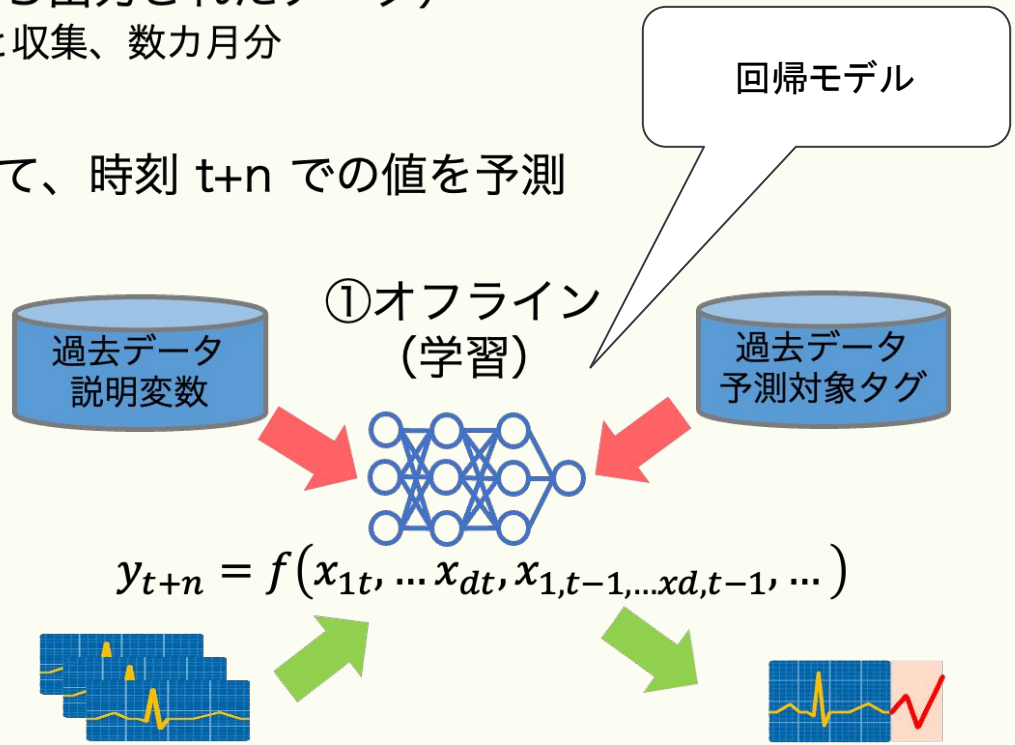
プロセスデータ収集時から20分後の製品品質(濃度X)を誤差±3%で予測

© Copyright NTT Communications Corporation All Rights Reserved.

事例1の分析手法（回帰分析を利用）

- データ
  - 多変量時系列データ（DCSから出力されたデータ）
    - 数十～数百タグ、1秒～1分ごと収集、数カ月分
- タスク
  - 時刻 t までのタグ情報を用いて、時刻 t+n での値を予測

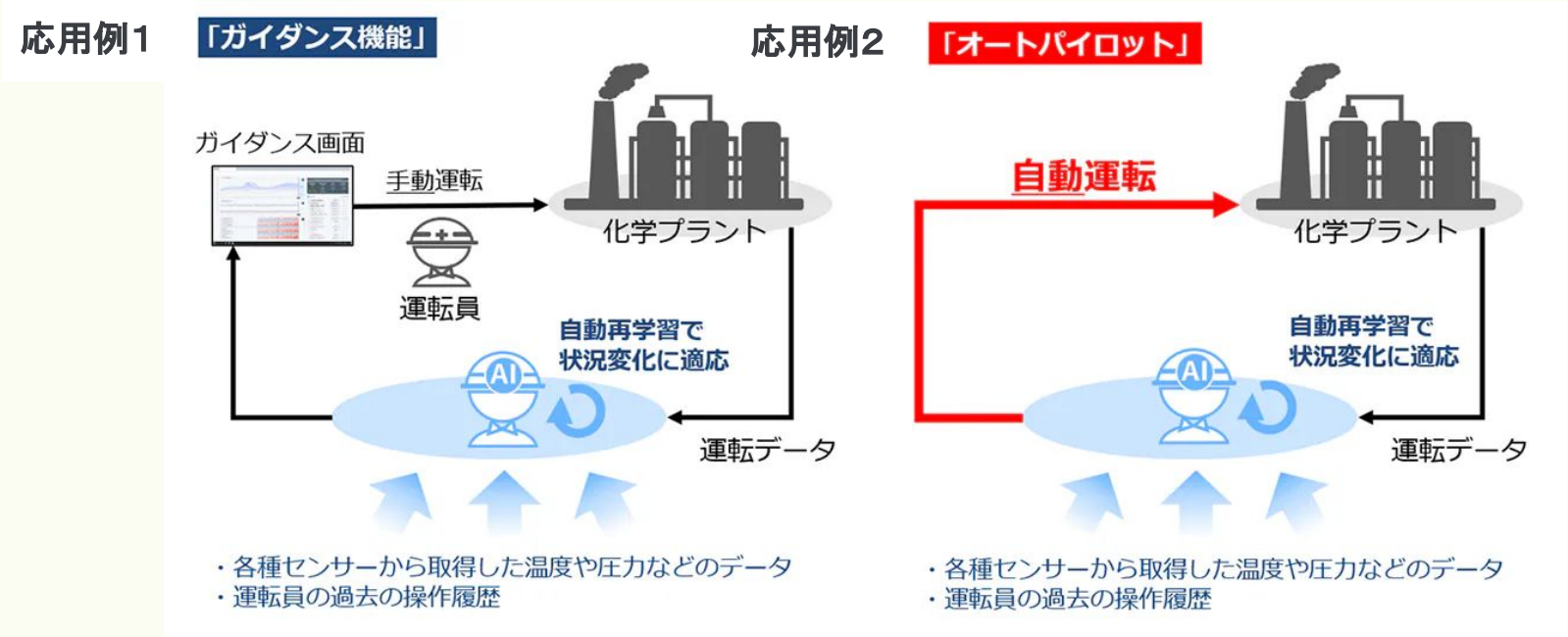




© Copyright NTT Communications Corporation All Rights Reserved.

事例2：人間の操作・運転の模倣

- 回帰分析を利用し、センサではなく、データから人間の操作を学習する
- 運用時には、回帰モデルを使って人間を模倣する操作値を算出する。



AIプラント運転支援ソリューション： https://www.ntt.com/business/dx/smart/factory/ai-plant.html

© Copyright NTT Communications Corporation All Rights Reserved.

ごちきか（本日の目玉）

課題　製造業の時系列に着目した時系列データ分析の標準手法・最新研究をまとめた書籍・Webサイトの欠乏

われわれの取り組み　ナレッジベース「ごちきか」の制作・公開

URL: https://gochikika.ntt.com/

コンテンツ

- **分析:** 主に製造業の時系列データを対象として、前処理からモデリングまで一連の基本的な分析手法をPythonコード付きで解説しています。
- **特集記事:** 比較的新しめであったり難易度の高い手法や、私たちの取り組みを知ってもらうための学会発表資料が掲載されます。また一部未分類なコンテンツが格納されています。

© Copyright NTT Communications Corporation All Rights Reserved.

ノーコードAIモデル開発ツール

Node AI https://nodeai.io/

データ処理を**カード化**、**連結**することで
分析フローを表現できる

コメントなどを
Markdown形式で
記述できる

時系列データの分析に強い
（予測、異常検知、要因・因果分析、など）

分析用キャンバスを
ネット上で共有して**コラボ**できる

© Copyright NTT Communications Corporation All Rights Reserved.

本日の講師陣

木村大地(きむら だいち)

- 専門: 時系列予測, 制御
- 趣味: 自転車, オタク

藤原大悟(ふじわら だいご)

- 専門: 因果探索/時系列分析/物理モデリング
- 趣味:エレキギター

泉谷知範(いずみたに とものり)

- 専門:データ分析一般・ビジネス検討・マネジメント
- 趣味:クラシックギター演奏

大川内智海(おおかわち ともみ)

- 専門: 数理統計・不確実性定量化・ブラックボックス最適化
- 趣味: 旅行・古楽・PCゲーム

石山隼(いしやま はやと)

- 専門: 因果推論・計量経済学
- 趣味: ボードゲーム

ごちきか執筆・編集メンバーです！

© Copyright NTT Communications Corporation All Rights Reserved.

背景

© Copyright NTT Communications Corporation All Rights Reserved.

はじめに

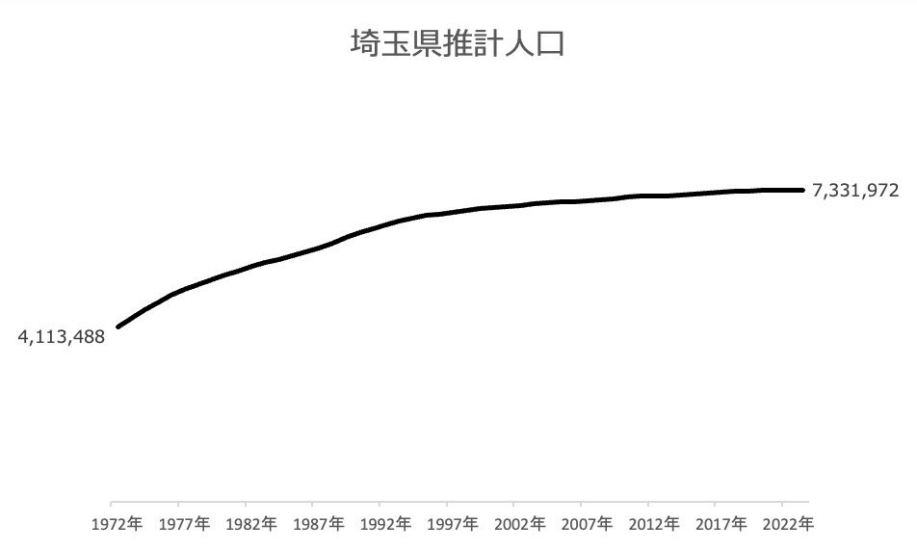
- 成長領域はどこか？(↔下降トレンドにある領域は？)
- 3か月後の売上は上がりそうか？下がりそうか？
- どのくらい発注するべきか？いつどんなときに需要があがるのか？
- ウェブサイトの訪問数は時間帯によってどのように変化する？
- 工場のセンサをもとによりよい操作をするには？

こういった疑問に、時系列データをうまく使うことができれば、答えることができます

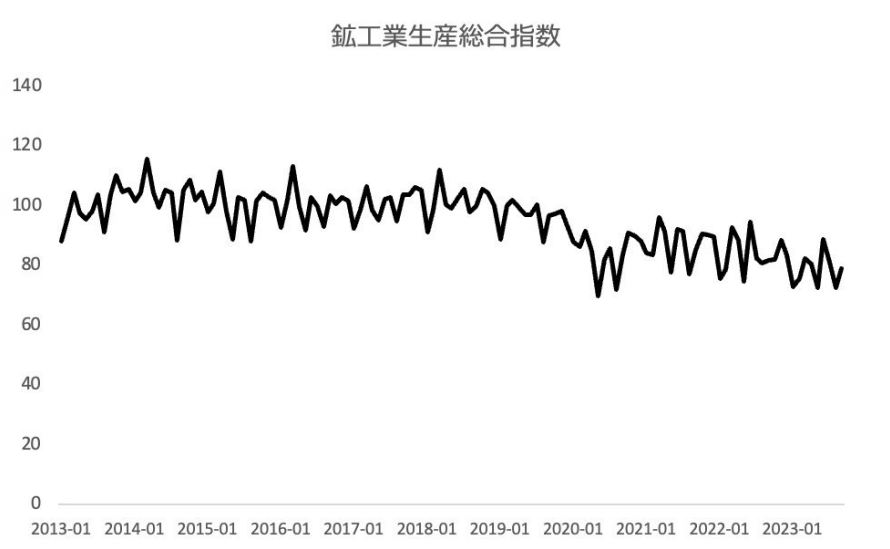
© Copyright NTT Communications Corporation All Rights Reserved.

時系列データとはなにか？

- 時系列データとは、一定間隔で記録されたデータの集まりのこと
- 折れ線グラフで表現されることが多い



埼玉県HPより作成：https://www.pref.saitama.lg.jp/a0206/03suikei/index.html



彩の国統計情報館ページ
（https://www.pref.saitama.lg.jp/a0206/a099/index.html）取得データから作成

© Copyright NTT Communications Corporation All Rights Reserved.

時系列データの具体例

時系列データそのものは、身近にたくさんあり、多くの人や企業が利用している

- 例えば
  - 工場：圧力、温度、濃度、流量、振動など
  - 営業：売上金額、販売数、受注数など
  - 医療：心拍数、血圧など
  - 気象：気温、降水量など

© Copyright NTT Communications Corporation All Rights Reserved.

時系列データ分析で、できること

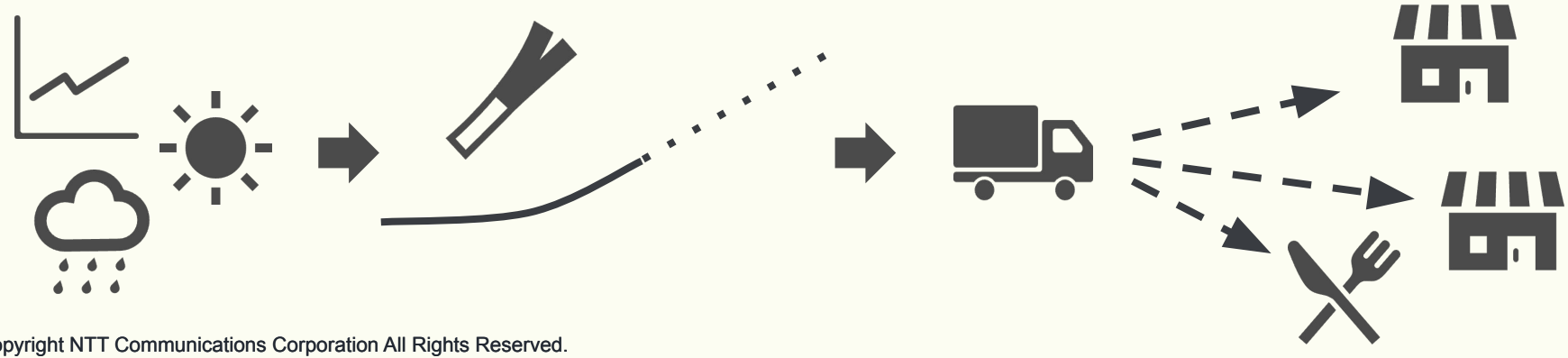
- **予測**
  - 将来的なデータの動きやパターンを予測する
  - 計画立案や意思決定に活かす
- **長期的な傾向の把握**
  - 季節性やトレンド、周期性をとらえる
  - 系列同士の関係を見る
  - 現象の理解に活かす

ほかにも、異常検知や因果関係の推定などができますが、今回の講義ではその基礎となる手法について話します

© Copyright NTT Communications Corporation All Rights Reserved.

時系列データ分析の具体例

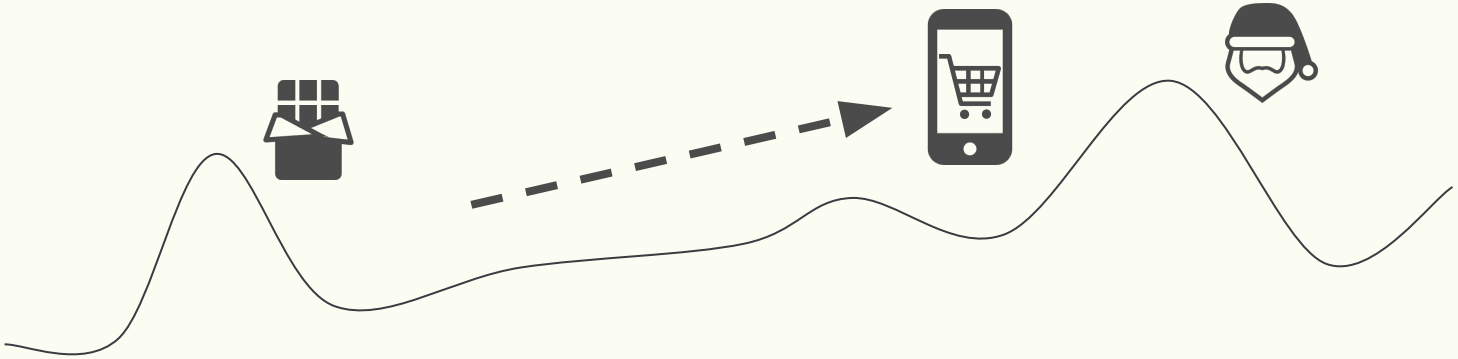
- **予測の例**
- 農業分野における作物の収穫予測
  - 過去の気温、降水量、日照時間をもとにモデルを作り、収穫量を予測する
  - 予測した収穫量に基づいて価格設定や流通計画を決定する



© Copyright NTT Communications Corporation All Rights Reserved.

時系列データ分析の具体例

- **長期的な傾向の把握の例**
- 小売業における商品の需要増加の背景理解
  - 季節性：クリスマスやバレンタインデーによる需要増加
  - トレンド：オンラインショッピングの増加
  - 周期性：高景気による需要増加



© Copyright NTT Communications Corporation All Rights Reserved.

この講義で行うこと

次の章からは、実際に時系列データの分析を行う方法をPythonコードとともに学びます

- 可視化と探索的データ解析/前処理～データ分析をする前に～
- 線形モデリング
- Deep Learningによる時系列予測

古典的な手法からはじめ、最終的にDeep Learningによる時系列モデルの概観とその課題をお話します

© Copyright NTT Communications Corporation All Rights Reserved.

この講義で行うこと

古典的な手法からはじめる理由

- 必ずしもDeep Learningを利用した方法が常にいい性能とは限らない
  - データの質と量
  - 予測までの時間
- 人の知見や能力を活かしたほうがいい場合がある
  - ドメイン知識によるモデリング
  - 可視化
- ベースラインとして利用する
- 手法の性質がよく知られている

© Copyright NTT Communications Corporation All Rights Reserved.

まとめ

- 時系列データとは、一定間隔で記録されたデータの集まりのこと
- 折れ線グラフで表現されることが多い
- 時系列データは、予測や長期的な傾向の把握に役立ちます
- 本講義では、古典的な手法からDeep Learningによる時系列モデルまでを話します

© Copyright NTT Communications Corporation All Rights Reserved.

可視化と探索的データ解析/前処理
～データ分析をする前に～

© Copyright NTT Communications Corporation All Rights Reserved.

可視化と探索的データ解析/前処理

この章では、本格的な分析の前段階で重要な技術である

- 可視化と探索的データ解析
- データの前処理

について説明します。

© Copyright NTT Communications Corporation All Rights Reserved.

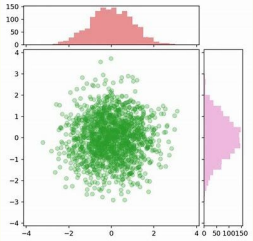
可視化と探索的データ解析

可視化とは？

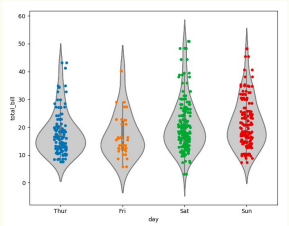
文字通りデータをプロットしたりすること。
具体的には、散布図やヒストグラム、折れ線グラフ、バイオリンプロット、
ヒートマップetc...といった様々な可視化方法がある。

探索的データ解析(Exploratory Data Analysis, EDA)

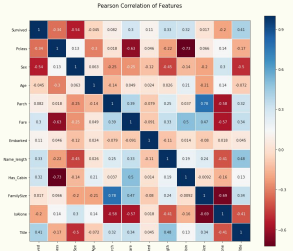
本格的なデータ分析を行う前に分析の仮説を立てること。
可視化や、平均分散、相関などを確認し、データの性質を見積もる。
これを事前に行わないと、見当違いの分析をしてしまうリスクが上がる。



ヒストグラム付き散布図



バイオリンプロット



相関行列 (ヒートマップ)

出典: [matplotlib] 21. ヒストグラム付き散布図 – サボテンパイソン,
stack overflow: Seaborn stripplot with violin plot bars in front of points,
Seabornで相関係数行列を書こう #Python - Qiita

© Copyright NTT Communications Corporation All Rights Reserved.

データの前処理

前処理とは？

データを分析しやすい形に整形すること。
実務上で出会うデータは、理想的なデータとは異なり、様々な誤差や特殊な仕様を持つ。具体的には例えば以下のようなものに対処する。

- 欠損値：何らかの理由で値が測定されていない例：12月15日の値だけ取れていない。
- 外れ値：何らかの異常な値が計測されている例：12月15日の値だけ異常に大きい値が入っている
- カテゴリカル変数：(大きさに意味のある)数値で表せない変数がある。例：血液型
- ノイズ：測定誤差などが乗っている 例：温度センサーの精度上、±1℃ズレた値が計測されている

© Copyright NTT Communications Corporation All Rights Reserved.

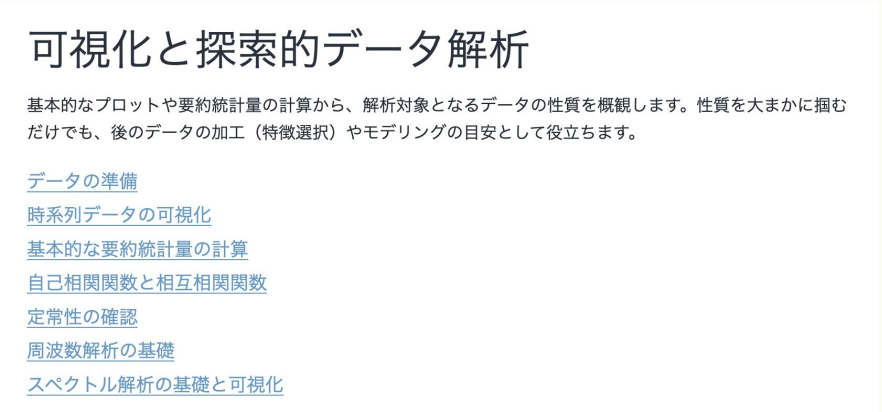
具体的な手法～可視化やEDA編～

可視化やEDAの前に**前処理**が必要になる場合もある。
(平均を計算する前に欠損値を取り除く、など)
実務上はこれらは合わせて実施していきましょう。(逆に、可視化して初めて欠損値に気づくことも多い)

可視化やEDAはごちきかでは右下のような内容で説明されている。(リンク)
今回は

- 時系列データの可視化
- 基本的な要約統計量の計算
- 自己相関関数と相互相関関数

に絞って説明する



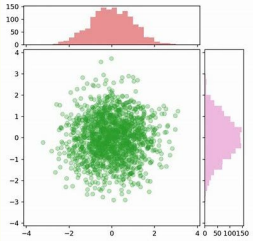
© Copyright NTT Communications Corporation All Rights Reserved.

時系列データの可視化

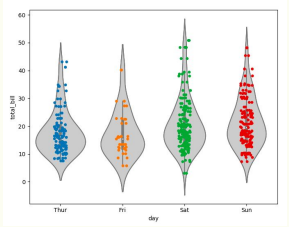
モジュールとしては、Pythonでは[Matplotlib](), [Plotly]()などが便利。

先ほどの図もMatplotlibやその派生の[seaborn]()で容易にプロット可。

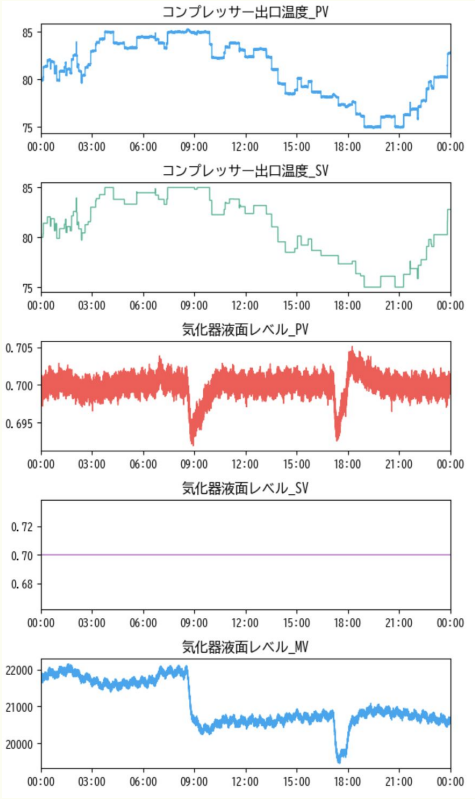
特に時系列データでは、複数の系列の時刻を揃えて折れ線グラフを描画したりする。
系列同士の相関を視覚的に把握することができる。



ヒストグラム付き散布図



バイオリンプロット



複数系列のプロット

出典：[matplotlib] 21. ヒストグラム付き散布図 – サボテンパイソン , stack overflow: Seaborn stripplot with violin plot bars in front of points,

© Copyright NTT Communications Corporation All Rights Reserved.

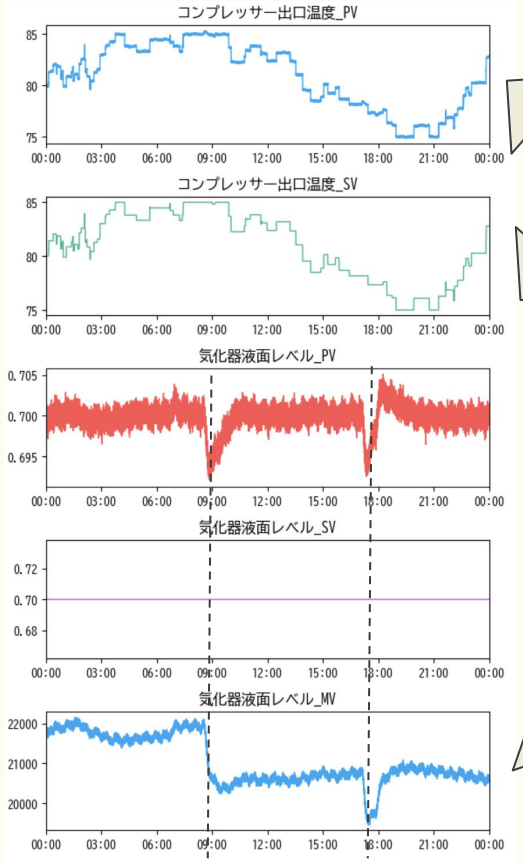
時系列データの可視化 (具体例)

例えば予測対象が一番下の「気化器液面レベルMV」の場合、

- 上二つが予測に有用そう
  - **多重共線性**があり片方で十分
- 「気化液面レベルPV」もインパルス的な影響を与えている

ことなどが予想できる

例えばこれらの知見を予測モデル構築時変数選択に生かせる



© Copyright NTT Communications Corporation All Rights Reserved.

基本的な要約統計量の計算

要約統計量とは、データ分布の特徴を大まかに表す指標群。
例えば平均、分散、中央値、最頻値、相関係数など、観測データから計算される。

PythonではPandasという表データ処理ライブラリを使うことで、いくつかの指標を一度に確認できる。

- `count`：そのカラムの（欠損値を除く）カウント数。
- `mean`：標本平均。
- `std`：標本標準偏差（ここでは不偏分散の平方根のこと）。
- `min`：最小値。
- `25%`：第一四分位点。
- `50%`：中央値。
- `75%`：第三四分位点。
- `max`：最大値。

- `unique`：そのカラムの（欠損値を除く）カテゴリ数。
- `top`：最頻値。
- `freq`：最頻値の頻度。

```
df.describe(include='all')
```

	気化器圧力_PV	気化器圧力_SV	気化器圧力_MV	気化器圧力_MODE	気化器液面レベル_PV	気化器液面レベル_SV	気化器ベ
count	86401.000000	86401.000000	86401.000000	86401	86401.000000	86401.0	86401.0
unique	NaN	NaN	NaN	1	NaN	NaN	
top	NaN	NaN	NaN	AUT	NaN	NaN	
freq	NaN	NaN	NaN	86401	NaN	NaN	
mean	128.003572	128.003523	18.739939	NaN	0.699817	0.7	21042.8
std	0.730975	0.528019	0.271741	NaN	0.001456	0.0	595.5
min	124.867305	126.117504	17.630283	NaN	0.691872	0.7	19450.3
25%	127.516310	127.638645	18.556992	NaN	0.699350	0.7	20619.9
50%	128.019870	128.032549	18.740014	NaN	0.699983	0.7	20782.9
75%	128.514749	128.406929	18.923368	NaN	0.700589	0.7	21668.3
max	130.558771	129.371837	19.971127	NaN	0.705129	0.7	22159.4

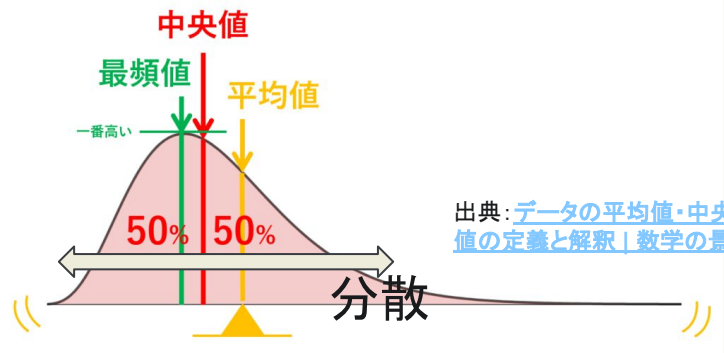
11 rows × 117 columns

© Copyright NTT Communications Corporation All Rights Reserved.

基本的な要約統計量の計算 (具体例)

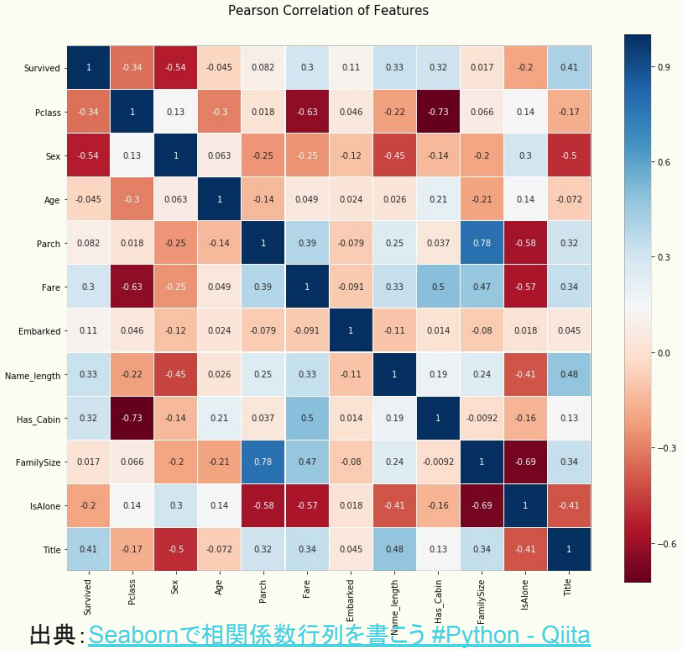
例えば年収の分布のように平均と最頻値、中央値が一般に一致しない分布の場合、**各種統計量は分布の形状の判断に非常に有用**。分散は、分布全体の幅を表します。

最大値、最小値で外れ値に、countなどで欠損値の存在に気づくこともできる。



出典：データの平均値・中央値・最頻値の定義と解釈 | 数学の景色

また、特に変数間の相関関係を定量的に測定できる**相関係数(相関行列)**は、変数選択などにあたって便利



出典：Seabornで相関係数行列を書こう #Python - Qiita

© Copyright NTT Communications Corporation All Rights Reserved.

自己相関関数と相互相関関数

時系列の場合、相関は容易に測ることができない。

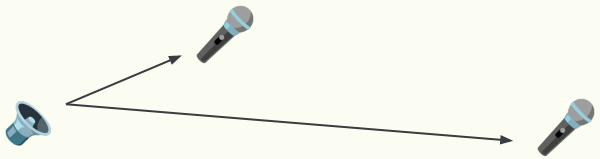
例えば、こんな極端な例ではどうだろうか？

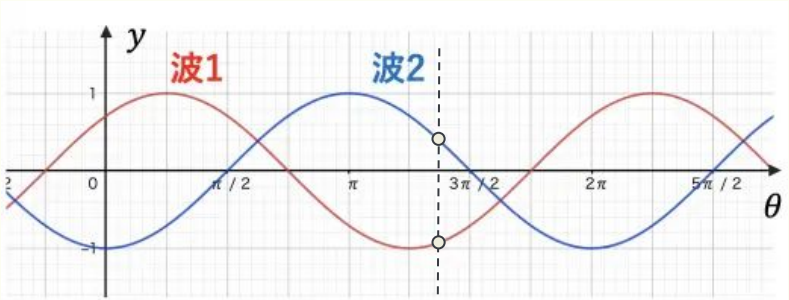
- 二地点で同じサイン音波を拾う
- ちょうど半周期分距離の差がある

$$r = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}\ \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

参考：相関係数の計算式
出典：https://office-hack.com/excel/correlation-coefficient/

波1と波2は”相関”があるのに、各時刻で区切って相関係数を計算するとゼロに近くなる





出典：https://manabitimes.jp/physics/1824

© Copyright NTT Communications Corporation All Rights Reserved.

自己相関関数と相互相関関数

時系列の場合、相関は容易に測ることができない

解決法：系列を少し(k秒)ずらして相関係数rを計算する

相関係数rはkごとに値が異なる関数となる
→相関関数:r(k)

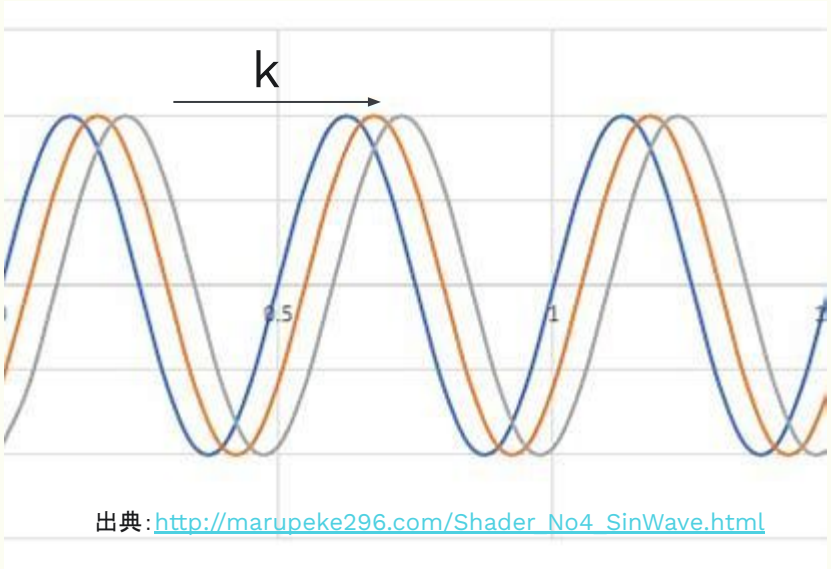
・他の系列に対して計算すれば相互相関関数
・自分自身に対して計算すれば自己相関関数

自己相関は、
- **どれくらいの期間、過去の影響を引きずるか**
- **周期性の検出**

などの指標になる

$$\frac{E[(x(t)-\mu_x)(y(t-k)-\mu_y)]}{\sqrt{\sigma_x^2}\sqrt{\sigma_y^2}}$$

参考：相関関数の計算式



出典：http://marupeke296.com/Shader_No4_SinWave.html

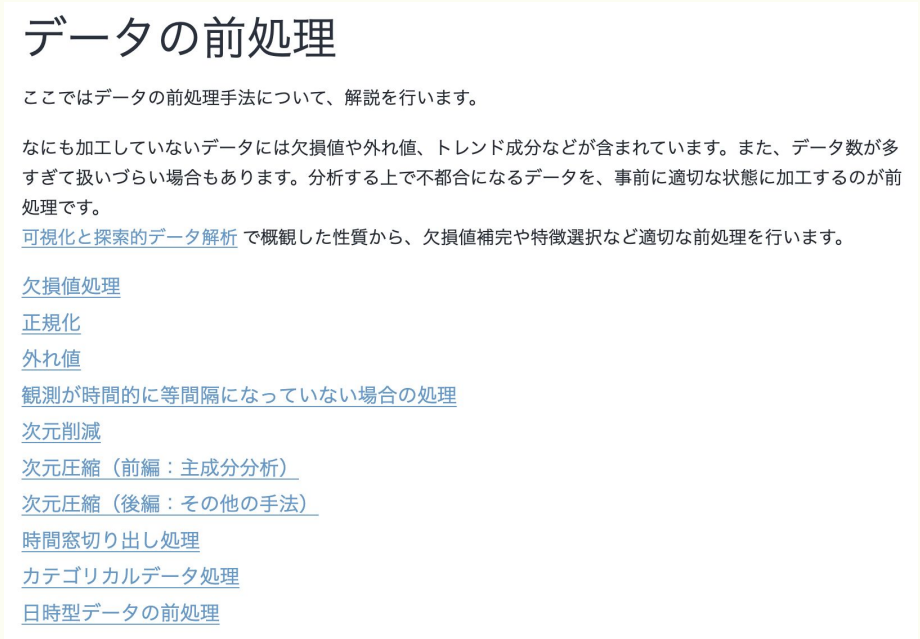
© Copyright NTT Communications Corporation All Rights Reserved.

具体的な手法～前処理編～

前処理はごちきかでは右下のような内容で説明されている。(リンク)
今回は

- 欠損値処理と外れ値
- 時間窓切出し処理

に絞って説明する



データの前処理

ここではデータの前処理手法について、解説を行います。

なにも加工していないデータには欠損値や外れ値、トレンド成分などが含まれています。また、データ数が多すぎて扱いづらい場合もあります。分析する上で不都合になるデータを、事前に適切な状態に加工するのが前処理です。
可視化と探索的データ解析 で概観した性質から、欠損値補完や特徴選択など適切な前処理を行います。

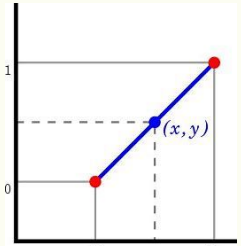
欠損値処理
正規化
外れ値
観測が時間的に等間隔になっていない場合の処理
次元削減
次元圧縮（前編：主成分分析）
次元圧縮（後編：その他の手法）
時間窓切り出し処理
カテゴリカルデータ処理
日時型データの前処理

© Copyright NTT Communications Corporation All Rights Reserved.

欠損値処理

欠損値とは？

データが欠けてしまっている状態。例えば特定の日付、時間のレコードだけない場合。特定のカラムだけが欠損してる場合も含む。



出典：Qiita「2点間の線形補間を計算する」

対処法

時系列データの場合、何らかの値で補間(置き換え)することが多い。

※時間的な連続性の仮定を利用しているとも言える

※時系列データでない場合、1つでも欠損を含む列は除去してしまうことも多い。

- 直前時刻を使う：欠損に一番近い**過去の観測**データで埋める。
- 線形補完：欠損から見て**未来の観測**も使い直線で繋いだ中間値で埋める。
  - 実運用時、欠損データが来ても対処できない欠点がある

© Copyright NTT Communications Corporation All Rights Reserved.

外れ値

外れ値とは？

何か異常な値が混入している状態。データをモデリングするにあたり、性質の異なる異常値が混ざっていると邪魔になってしまう。例えば、観光客数をモデリングするときにコロナの期間は除きたい。発注数モデリングにおいて、入力ミスによる桁間違いなども。

対処法

- 明確に期間が分かる場合はそれを除去。
- その他の外れ値：
  - まず、**外れ値とは何かを定義**。どこからが異常？(次で紹介)
  - **外れ値を除去**。or **欠損値と同様に置き換え**。
    - まとまった期間ならゴソっと除去。1点程度なら補間することが多い。

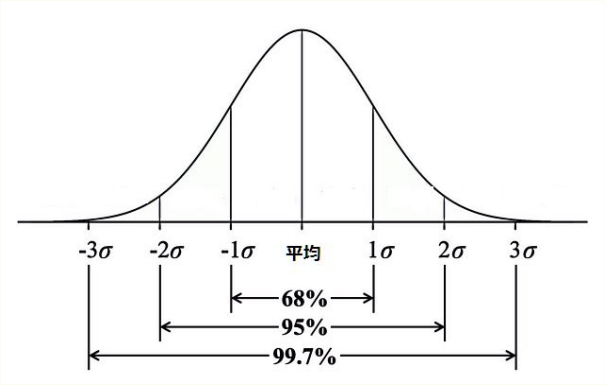
© Copyright NTT Communications Corporation All Rights Reserved.

外れ値 (具体例)

外れ値の定義は？

データには自然な変動も含まれており、“極端な”値を除去するには閾値を設定する必要がある。

対処法:
よく使われる方法として、データの標準偏差σに対して「平均±3σ」を超える範囲にある点を外れ値とする方法など。
データの出現確率をベースとした、仮説検定的な考え方をよく用いる。



出典:pythonで正規分布（ガウス分布）を作る |Tech Press | テックプレス

© Copyright NTT Communications Corporation All Rights Reserved.

時間窓切出し処理

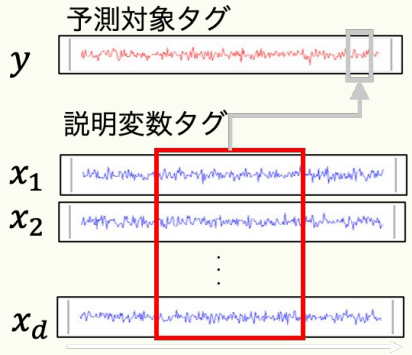
時間窓切出し処理とは？:

時系列回帰の際に重要になる、特徴量作成方法。「窓枠処理」「時間窓切り出し」「Time delay embedding」「Lagged feature」などとも一般に呼ばれる。

具体例:

例えば、多変量時系列(系列が複数ある時系列)に関して回帰タスクを考える。

- 複数の系列：x_i(t) (i=0,1,2,...D) があり、
- ある系列：y(t)=x_0(t) の、
- 未来の値：y(t+n) (n=1,2,...,N) を予測したいとする。



この時回帰モデルへの入力として、

- 直前の時刻の値：x_i(t) (i=0,1,2,..,D) だけでなく、
- m時刻前の値：x_i(t-m) (i=0,1,2,...D) (m=1,2,...,M) も用いた方が正確な予測が可能である。

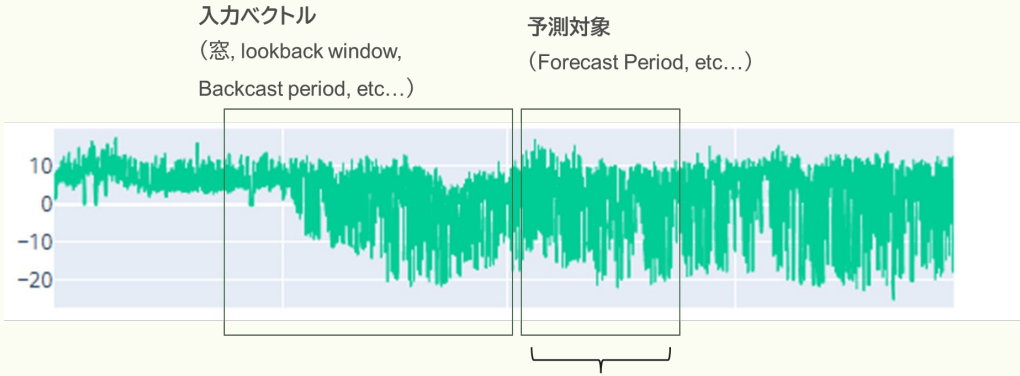
© Copyright NTT Communications Corporation All Rights Reserved.

時間窓切出し処理

図にすると右のようなイメージ。

- 単変量データの場合、
  作成された窓枠入力は、1次元のベクトル：
  $\mathbf{x}(t)=(x(t), x(t-1), x(t-2),\ldots,x(t-M))^T$
  となる。
- 多変量データの場合、
  作成された窓枠入力は2次元配列：
  $X(t)=(\mathbf{x}_1(t), \mathbf{x}_2(t),\ldots,\mathbf{x}_D(t))$
  となる。

※実際には、このtの値を変えて窓をスライドさせて入出力関係(x,y)を表す学習データを複数作成するため、入力データは単変量では2次元配列、多変量では3次元配列となる。(右図は多変量の場合)



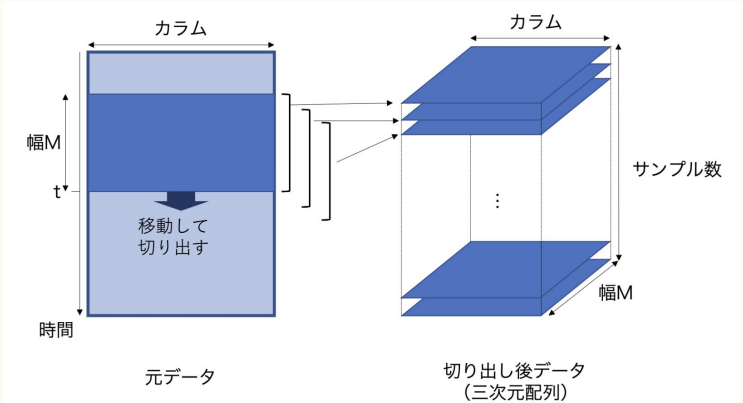


図１：時間窓切り出し処理（説明変数）

© Copyright NTT Communications Corporation All Rights Reserved.

線形モデリング

© Copyright NTT Communications Corporation All Rights Reserved.

注意事項

このセクションでは、主に

- 機械学習
- システム同定

の分野におけるモデリングの文脈にそって話を進めます。

（特に、統計的時系列解析の文脈におけるいくつかの重要事項（※）についてはほとんど触れません）

（※）例えば、単位根の問題、共和分の問題、残差の自己相関の問題など。

© Copyright NTT Communications Corporation All Rights Reserved.

時系列回帰モデリングとは？

ある時刻 t での興味がある変数 y の値を、
その時刻 t よりも過去の時刻（※）での
　　その変数 y 自身の値や関連する変数 x_k たちの値を用いて
うまく表現できるような関係性 f を見つけること；

　　何かの関係性 f：
　　（変数 y の t より過去の値，変数 x_k たちの t より過去の値）
　　↦ 変数 y の時刻 t での値

（※）ただし、関連する変数 x_k たちについては同じ時刻 t の値を用いてよい場合もあります。

© Copyright NTT Communications Corporation All Rights Reserved.

時系列回帰モデリングの目的

目的 (A): 予測

新しい時刻、未知の時刻のデータを与えられたときに
対応する目的変数 y の値を正確に予測すること。

目的 (B): 表現

主に既存・既知の時刻のデータについて
目的変数 y や各特徴量 x_k の時間的な関係性を正確に表現できること。

→ 機械学習では、このうち目的 (A): 予測がしばしば重視される。

© Copyright NTT Communications Corporation All Rights Reserved.

記号の準備

- t: 時刻（t = 1, 2, ...）
- y[t]: 時刻 t での目的変数 y の値
- **y** = (y[1], y[2], ..., y[T]): 目的変数の値全体のベクトル
- x_k[t]: 時刻 t での k 番目の特徴量 x_k の値
- **x**[t] = (x_1[t], x_2[t], ..., x_K[t]): 時刻 t での特徴量の値のベクトル
- X = (**x**[1]', **x**[2]', ..., **x**[T]')': 特徴量の値全体の行列 ('は転置記号)
- D = (X, **y**): 利用できる時系列データの全体

→ 時系列回帰では、データ D から y[t] を表現する関係性を見出したい。

© Copyright NTT Communications Corporation All Rights Reserved.

時系列回帰モデリングの定式化

時刻 t での目的変数 y の値 y[t] を
過去の目的変数 y 自身および各特徴量 x_k の値から表現する式

$$y[t] = f(\mathbf{x}[t-1], \mathbf{x}[t-2], \ldots, y[t-1], y[t-2], \ldots) + \varepsilon[t]$$

を発見すること（※）。
（ただし ε[t] は時刻tにおけるそれぞれの式の誤差項）

→ 特に、関数 f を特定のクラス（線形関数など）に限定して探索したい。

（※）正確には、この課題は時系列回帰のうち「1ステップ先予測」などと呼ばれる種類のものです。

© Copyright NTT Communications Corporation All Rights Reserved.

線形の時系列回帰モデル

自己回帰モデル（ARモデル）:

$$y[t] = c + a[1]\, y[t-1] + \ldots + a[n]\, y[t-n] + \varepsilon[t]$$

線形回帰モデル:

$$y[t] = c + b_1[1]\, x_1[t-1] + \ldots + b_K[n]\, x_K[t-n] + \varepsilon[t]$$

ARXモデル:

$$y[t] = c + a[1]\, y[t-1] + \ldots + a[n]\, y[t-n] + b_1[1]\, x_1[t-1] + \ldots + b_K[n]\, x_K[t-n] + \varepsilon[t]$$

© Copyright NTT Communications Corporation All Rights Reserved.

線形回帰モデル

線形回帰モデル（※）:

$$y[t] = c + b_1[1]\, x_1[t-1] + \dots + b_K[1]\, x_K[t-1] + b_1[2]\, x_1[t-2] + \dots + b_K[2]\, x_K[t-2] + \dots + b_1[n]\, x_K[t-n] + \dots + b_K[n]\, x_K[t-n] + \varepsilon[t]$$

c, (b_k[s]): モデルのパラメータ（切片・係数）

→ 過去 n 時刻分の特徴量の値のみを利用し、線形結合

（※）計量経済学では「分布ラグモデル」などとも呼ばれる。

© Copyright NTT Communications Corporation All Rights Reserved.

ARXモデル

ARXモデル（※）:

$$y[t] = c + a[1]\, y[t-1] + \dots + a[n]\, y[t-n] + b_1[1]\, x_1[t-1] + \dots + b_K[1]\, x_K[t-1] + b_1[2]\, x_1[t-2] + \dots + b_K[2]\, x_K[t-2] + \dots + b_1[n]\, x_K[t-n] + \dots + b_K[n]\, x_K[t-n] + \varepsilon[t]$$

c, (a[s]), (b_k[s]): モデルのパラメータ（切片・係数）

→ 過去 n 時刻分の目的変数の値と特徴量の値を用い、線形結合

（※）計量経済学では「自己回帰分布ラグモデル」などとも呼ばれる。時系列解析の「ARXモデル」とは別物。

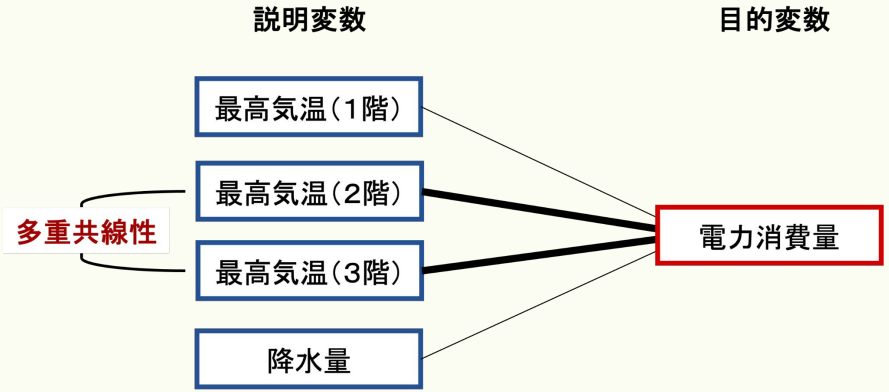
© Copyright NTT Communications Corporation All Rights Reserved.

線形モデルにおける多重共線性の問題

線形モデルでは、学習データの説明変数間に高い相関があるときなどに
未知データに対する予測性能が著しく低下する問題が発生
→「多重共線性の問題」

主な対処法：

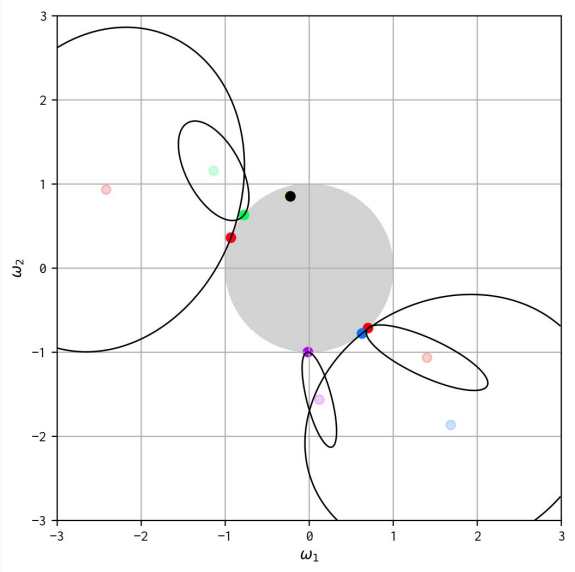
- 係数の正則化による縮小推定
- 学習データの次元削減
  （前処理 or モデリング）

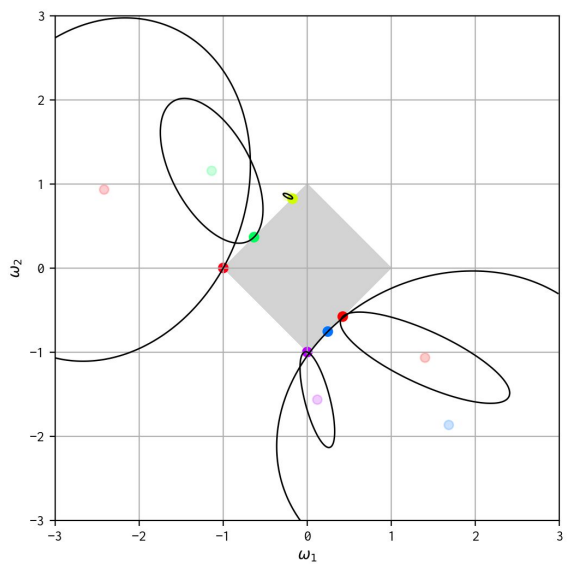


© Copyright NTT Communications Corporation All Rights Reserved.

係数の正則化による縮小推定

L2正則化 (Ridge) / L1正則化（LASSO）



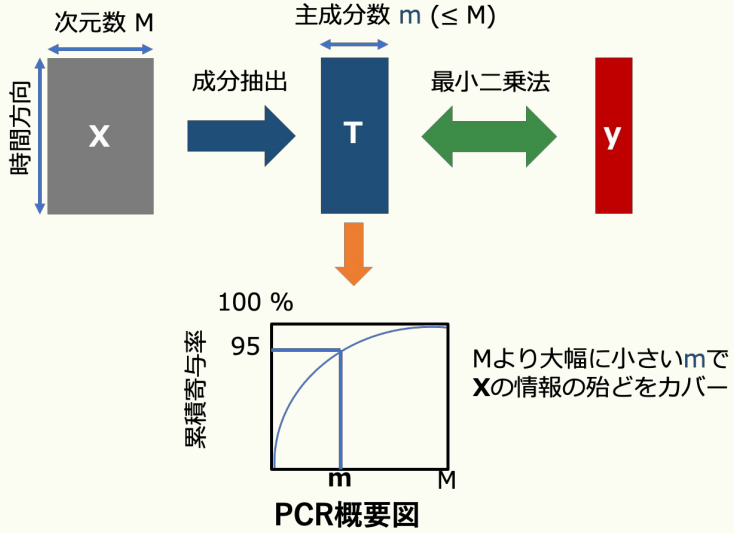


(左) L2正則化による影響
(右) L1正則化による影響

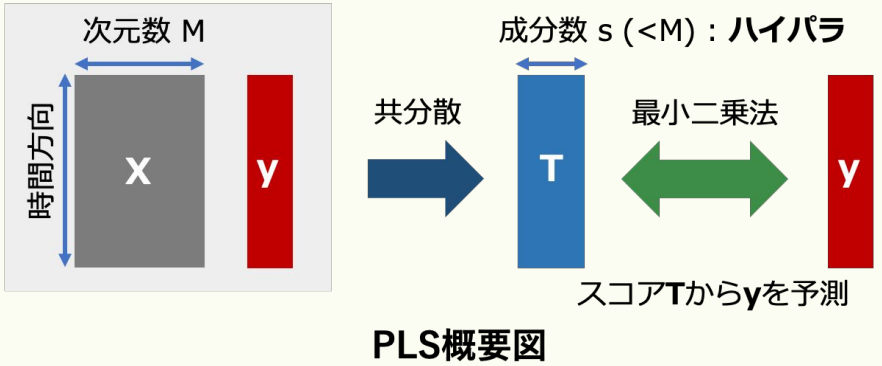
© Copyright NTT Communications Corporation All Rights Reserved.

次元削減を伴うモデリング

PCR（主成分回帰）モデル / PLS（部分最小二乗法）回帰モデル



PCR概要図



PLS概要図

(左) PCRモデルの概要図
(右) PLSモデルの概要図

© Copyright NTT Communications Corporation All Rights Reserved.

学習データの分割

機械学習モデルの検討段階においては、
利用できる時系列データの全体 D = (X, **y**) を分割して

- 訓練データ D_train
- 試験データ D_test

とし、このうち訓練データ D_train だけを用いてモデルの学習を行い、
その予測結果を試験データ D_test に基づいて評価することが一般的。

→ D_test での誤差を未知データに対する誤差の指標として考える。

© Copyright NTT Communications Corporation All Rights Reserved.

パラメータの選択とvalidation

更に、モデルや学習プロセスの中に
事前に決めるべきパラメータ（ハイパーパラメータ）がある場合、

- 訓練データ D_train から更に検証データ D_val を分割
- 訓練データ内で交差検証（cross-validation, CV）を行う（※）

などの方法で、試験データ D_test を使わずにパラメータを決定する。

例：時間窓の幅 n 、L1正則化の正則化係数、PLS回帰の成分数

（※）ただし、時系列回帰に限っては素朴なK-fold CVを行うとデータ漏洩を引き起こすので厳に注意が必要。

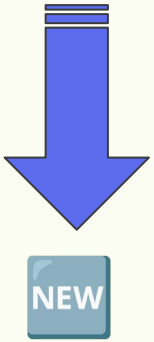
© Copyright NTT Communications Corporation All Rights Reserved.

Deep Learningによる時系列予測

© Copyright NTT Communications Corporation All Rights Reserved.

Deep Learningベースの時系列予測モデル

主に4種類 ※



- MLP (Multi-Layer Perceptron)
- RNN (Recurrent Neural Network)
- CNN (Convolutional Neural Network)
- Transformer

※ Graph Neural Network もあるが少しマイナー

次のスライドで概要を簡単に解説

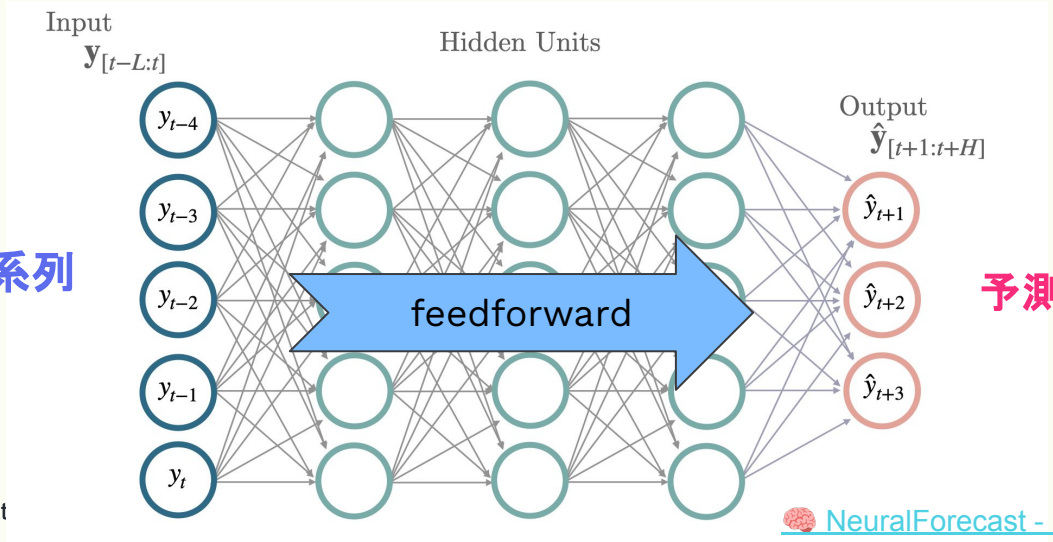
© Copyright NTT Communications Corporation All Rights Reserved.

MLP (Multi-Layer Perceptron)

- 一番ベーシック。基本的なモジュールとしても使われる。
- 各ノード（Unit）で非線形変換を行い、組み合わせることで高い表現能力

活性化関数（非線形変換）　　学習パラメーター

$$\mathbf{x} \leftarrow f(\mathbf{W}\mathbf{x} + \mathbf{b})$$



過去系列

予測系列

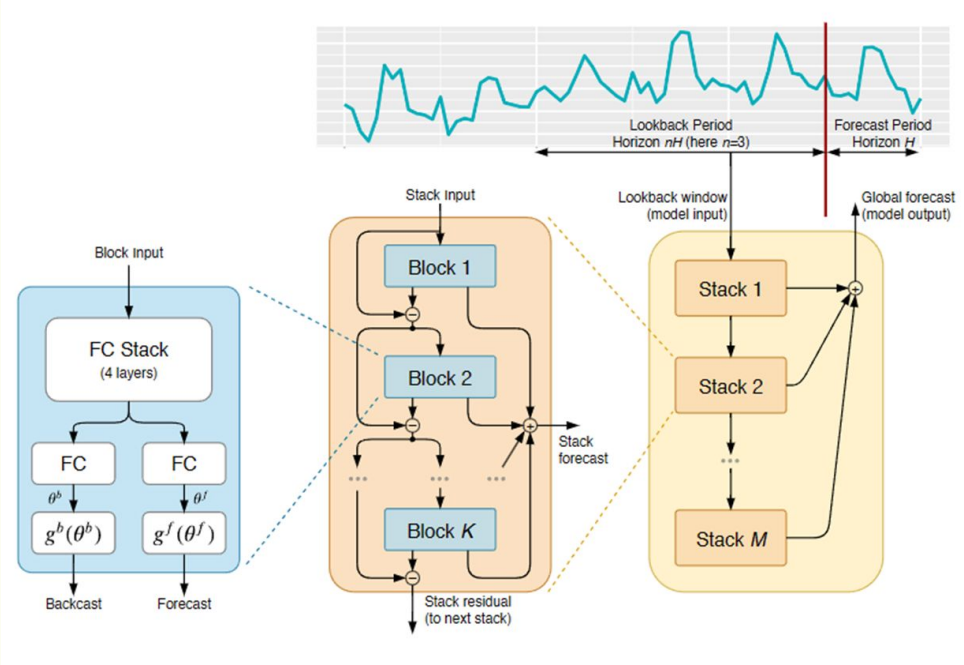
© Copyright NTT Communicat

NeuralForecast - Nixtla

最近のMLPベース

N-BEATS (2019), N-HiTS(2023) など

- 階層状にブロックを積み重ねる
- 各ブロックは時系列の
  季節性やトレンドを捉えるように工夫



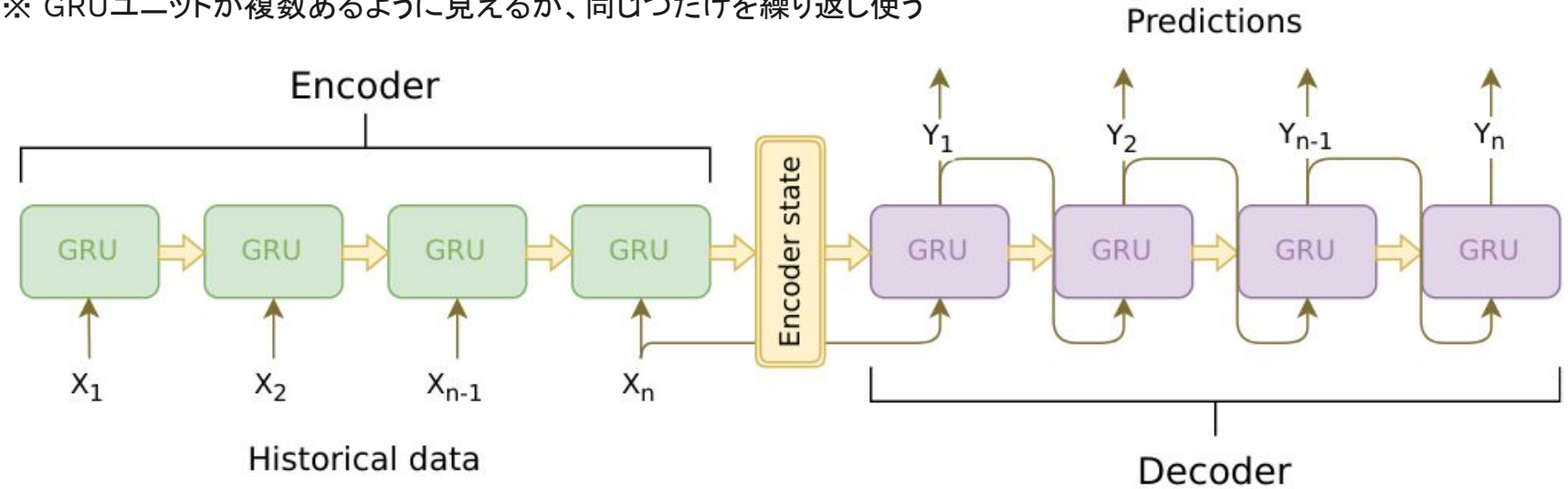
Oreshkin, Boris N., et al. "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting." *arXiv preprint arXiv:1905.10437* (2019).

© Copyright NTT Communications Corporation All Rights Reserved.

RNN, Recurrent Neural Network

- 時系列性を明示的に取り込む。もともとは **自然言語処理** で発達
  - 遠い過去の情報を保持しながら、次のステップの予測を繰り返し行う
- **代表例:** LSTM (1997), GRU (2014), DeepAR(2019)

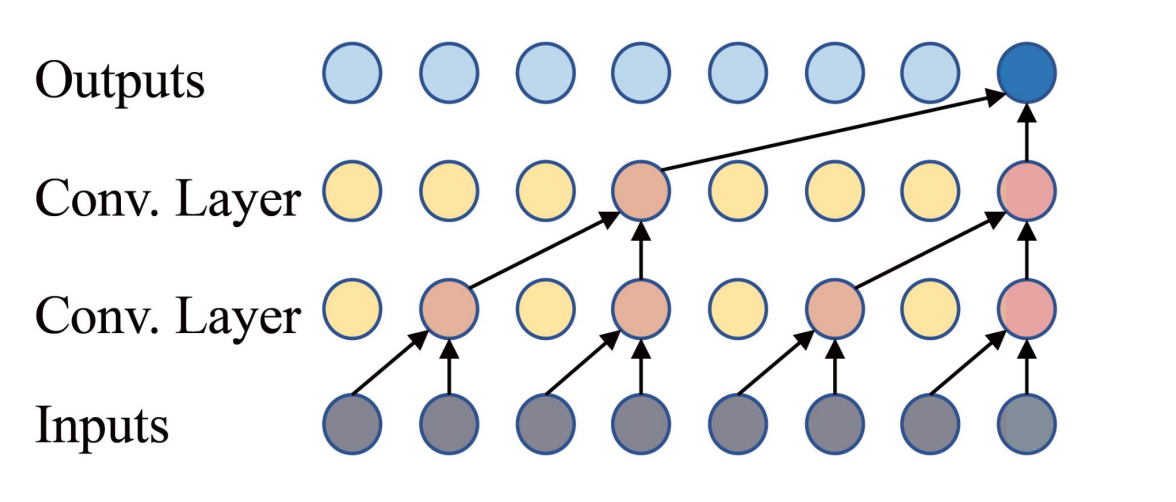
※ GRUユニットが複数あるように見えるが、同じつだけを繰り返し使う



sooftware/seq2seq: PyTorch implementation of the RNN-based sequence-to-sequence architecture. (github.com)

CNN, Convolutional Neural Network

- より遠く過去の情報を取り込む。もともとは**画像処理**で発達
  - 入力を広くとり、畳み込み層を通るたびに情報集約。
- 代表例: WaveNet (2016), SCINet (2021)



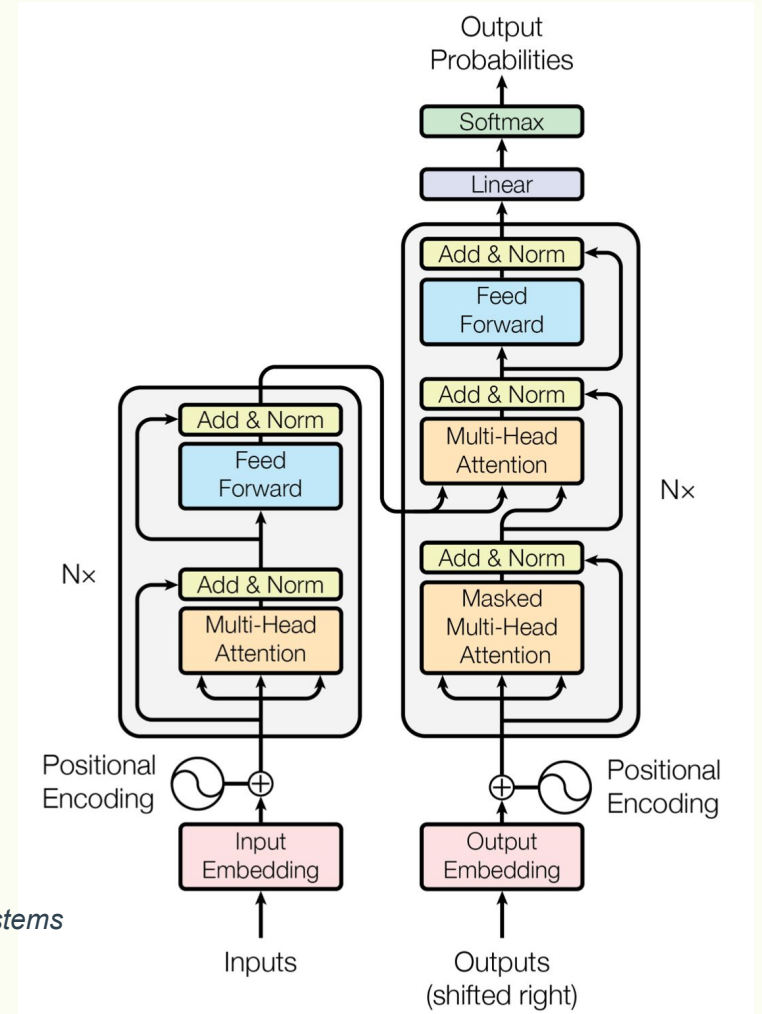
Liu, Minhao et al. “SCINet: Time Series Modeling and Forecasting with Sample Convolution and Interaction.” *Neural Information Processing Systems* (2021).

© Copyright NTT Communications Corporation All Rights Reserved.

Transformer

Attention Is All You Need

- Attentionを積み重ねた構造
  - 後ほど解説
- もともとは**自然言語処理**
  - RNNより長期記憶に特化
- **ChatGPT**に代表される生成系AIの基本構造
- 今回の講義ではこの最新手法を扱う

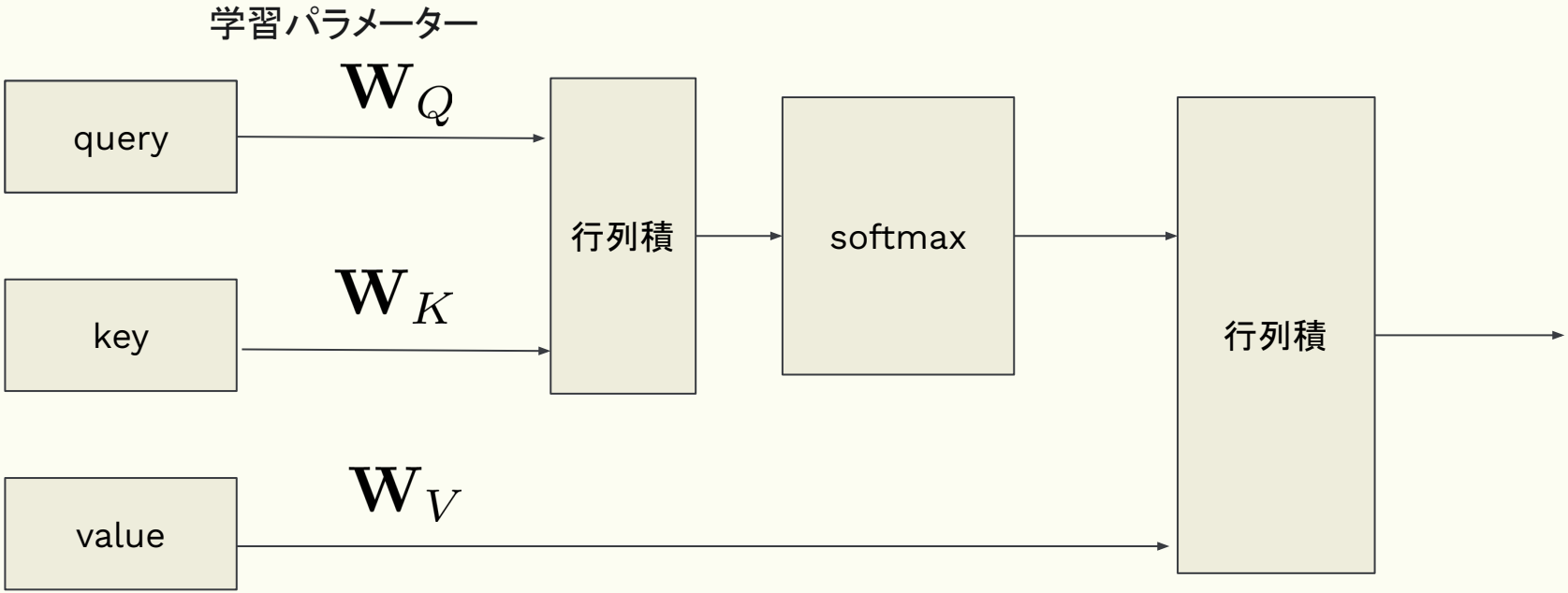


Vaswani, Ashish et al. “Attention is All you Need.” *Neural Information Processing Systems* (2017).

© Copyright NTT Communications Corporation All Rights Reserved.

Attention　全体像

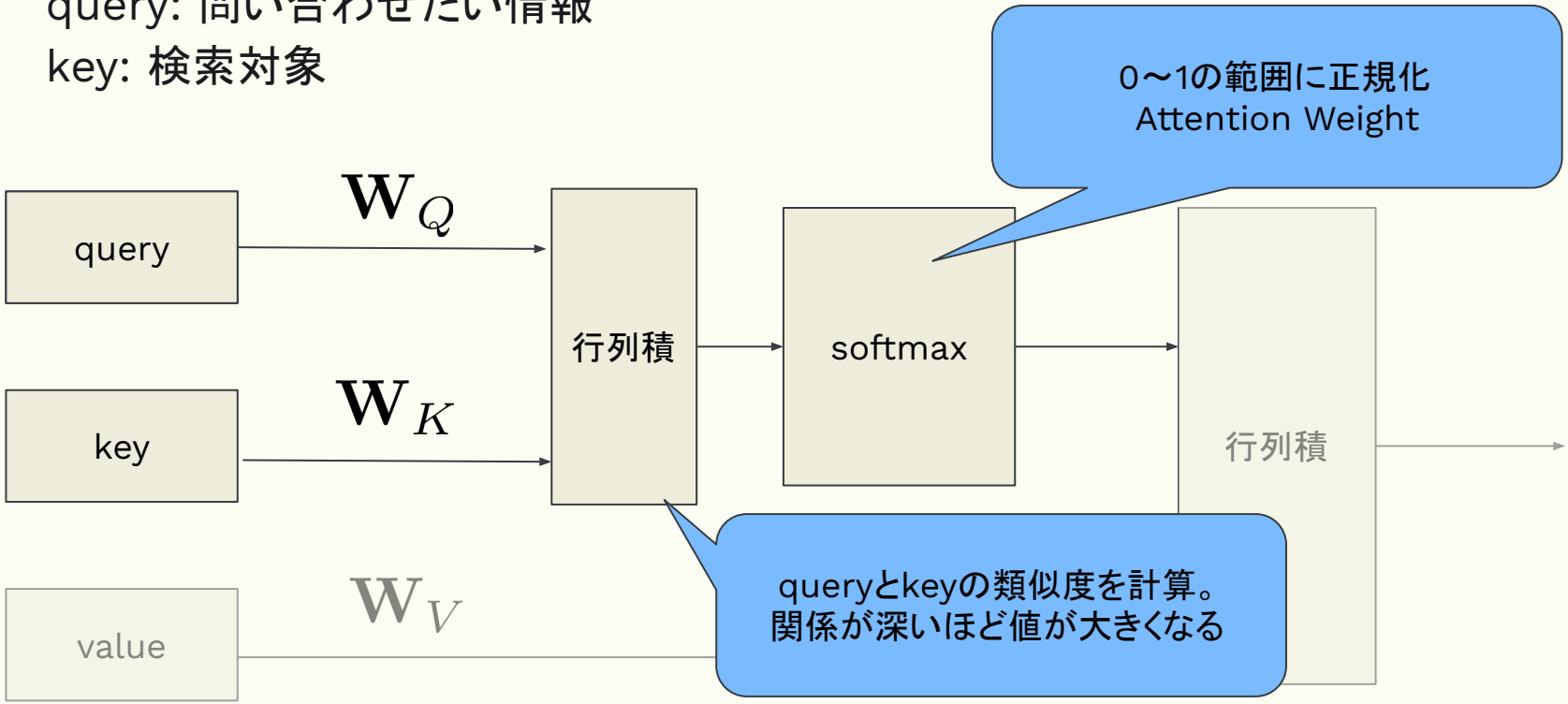
入力系列のうち重要な要素に注目（=Attention）する構造



© Copyright NTT Communications Corporation All Rights Reserved.

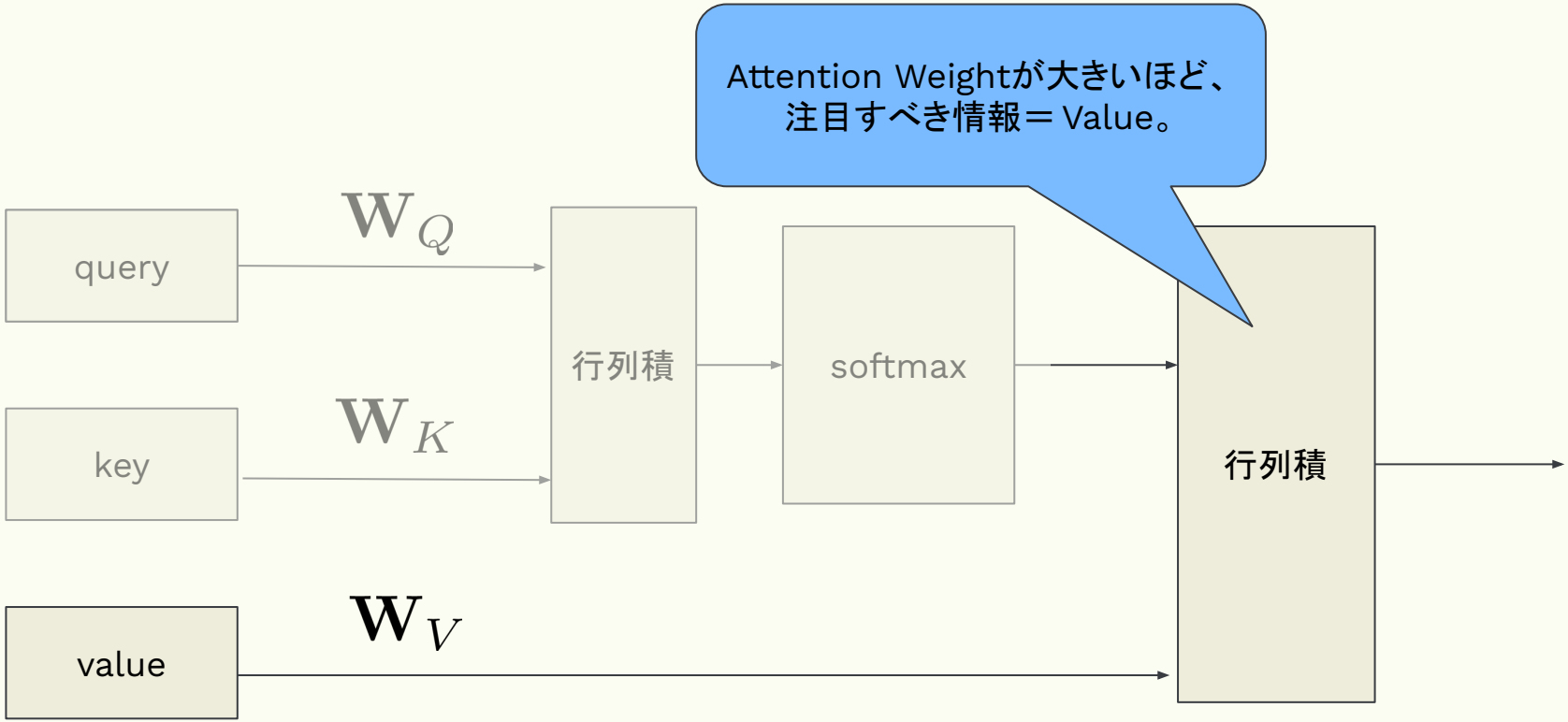
Attention

query: 問い合わせたい情報
key: 検索対象



© Copyright NTT Communications Corporation All Rights Reserved.

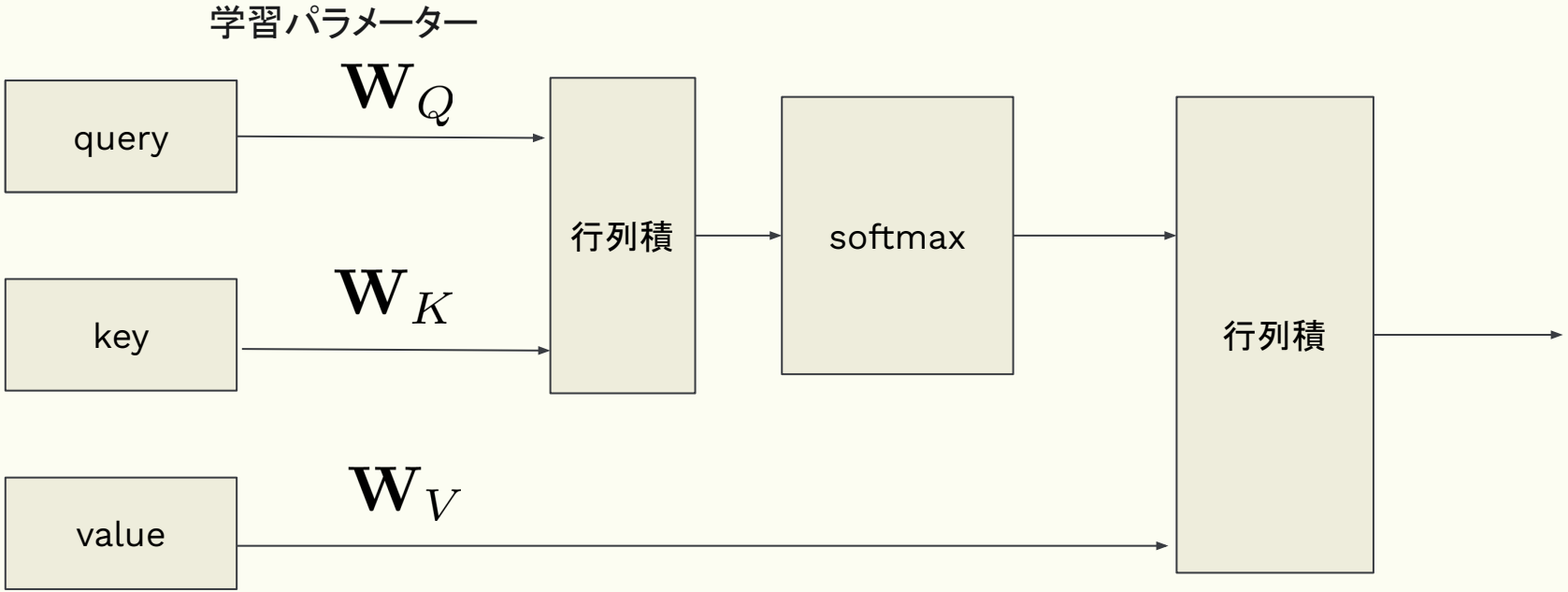
Attention



© Copyright NTT Communications Corporation All Rights Reserved.

Self-Attention

query = key = value
すなわち、自分自身(=self)の要素間のうち注目する部分を計算するイメージ



© Copyright NTT Communications Corporation All Rights Reserved.

Transformerベースのモデル

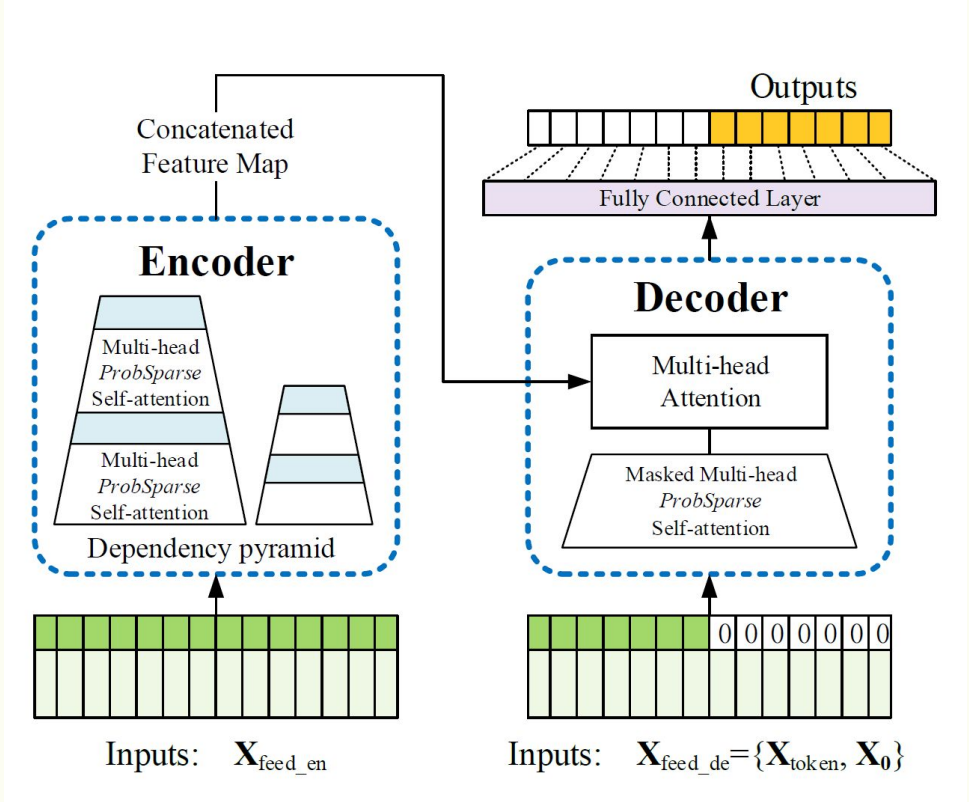
多数提案されている

- 計算量削減
  - LogTrans [Li+ 19]
  - **Informer [Zhou+ 21]**
  - Pyraformer [Liu+ 22]
- 時系列的な特徴（周波数領域，自己相関，季節トレンド分解など）
  - Autoformer [Wu+ 21]
  - FEDformer [Zhou+ 22]
- 非定常系列
  - Non-stationary Transformers

© Copyright NTT Communications Corporation All Rights Reserved.

Informer

- トップ国際学会 AAAI2021
  ベストペーパー
- 特に計算コストの削減に着目
  - Attentionの計算は重たい
  - 系列長 L に対して、
    - Transformer: $O(L^2)$
    - Informer: $O(L \log L)$



Zhou, Haoyi, et al. "Informer: Beyond efficient transformer for long sequence time-series forecasting." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 35. No. 12. 2021.

© Copyright NTT Communications Corporation All Rights Reserved.

ハンズオン

© Copyright NTT Communications Corporation All Rights Reserved.

Are Transformers Effective for Time Series Forecasting?

- AAAI2023で発表
- Attentionの性質により時間情報が失われてしまうのでは？という仮説
- 恥ずかしいぐらいシンプルな1層線形モデルとTransformerを比較
- 9種類のデータセットで実験し、Transformerの性能を上回ることを確認

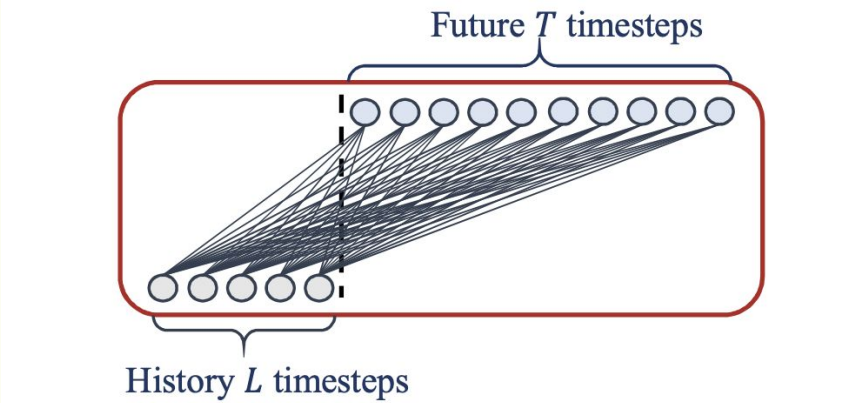


Figure 2: Illustration of the basic linear model.

Zeng, Ailing, et al. "Are transformers effective for time series forecasting?." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 37. No. 9. 2023.

© Copyright NTT Communications Corporation All Rights Reserved.

Are Transformers Effective for Time Series Forecasting?

Linear: 1層の線形モデル

NLinear: 正規化処理

- データに分布シフトが含まれている場合に有利

DLinear: 季節-トレンド分解

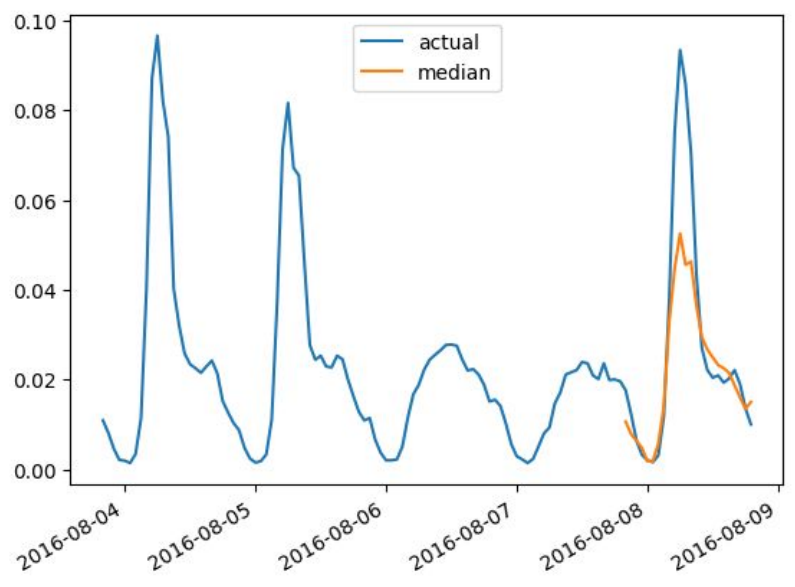
- データにトレンド成分があるときに有利

Methods		IMP.	Linear*		NLinear*		DLinear*		FEDformer		Autoformer		Informer		Pyraformer*		LogTrans	
Metric		MSE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	27.40%	**0.140**	**0.237**	0.141	**0.237**	**0.140**	**0.237**	0.193	0.308	0.201	0.317	0.274	0.368	0.386	0.449	0.258	0.357
	192	23.88%	**0.153**	0.250	0.154	**0.248**	**0.153**	0.249	0.201	0.315	0.222	0.334	0.296	0.386	0.386	0.443	0.266	0.368
	336	21.02%	**0.169**	0.268	0.171	**0.265**	**0.169**	0.267	0.214	0.329	0.231	0.338	0.300	0.394	0.378	0.443	0.280	0.380
	720	17.47%	**0.203**	0.301	0.210	**0.297**	**0.203**	0.301	0.246	0.355	0.254	0.361	0.373	0.439	0.376	0.445	0.283	0.376
Exchange	96	45.27%	0.082	0.207	0.089	0.208	**0.081**	0.203	0.148	0.278	0.197	0.323	0.847	0.752	0.376	1.105	0.968	0.812
	192	42.06%	0.167	0.304	0.180	0.300	**0.157**	0.293	0.271	0.380	0.300	0.369	1.204	0.895	1.748	1.151	1.040	0.851
	336	33.69%	0.328	0.432	0.331	0.415	**0.305**	0.414	0.460	0.500	0.509	0.524	1.672	1.036	1.874	1.172	1.659	1.081
	720	46.19%	0.964	0.750	1.033	0.780	**0.643**	**0.601**	1.195	0.841	1.447	0.941	2.478	1.310	1.943	1.206	1.941	1.127

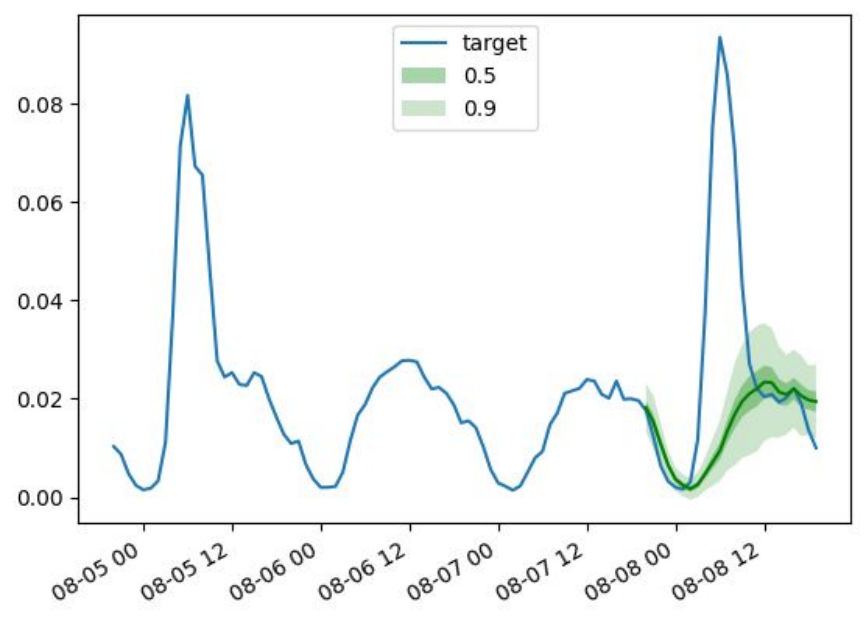
Yes, Transformers are Effective for Time Series Forecasting 🤗

サンフランシスコの交通量予測[Lai+ 17] データセット

Autoformer



DLinear



Yes, Transformers are Effective for Time Series Forecasting (+ Autoformer) (huggingface.co)

© Copyright NTT Communications Corporation All Rights Reserved.

Yes, Transformers are Effective for Time Series Forecasting 🤗

Dataset	Transformer (uni.)	Transformer (mv.)	Informer (uni.)	Informer (mv.)	Autoformer (uni.)	DLinear
Traffic	**0.876**	1.046	0.924	1.131	0.910	0.965

- 意外なことに素のTransformerが一番いい性能
  - そもそも単純なLinearモデルは多変量に対応できない
- 単変量(uni.)に比べて、多変量(mv.)は性能が悪い
  - 系列間の相関や因果関係の推定が難しい
    - → CrossFormer (2023), CARD (2023)
  - 多変量モデルは大規模データセットで有利
    - Transformerの可能性を最大限引き出すには、**大規模時系列データが不足**
    - 時系列: ～10GB, 画像, 自然言語: 数百GB～TB

© Copyright NTT Communications Corporation All Rights Reserved.

結論

- Deep Learningベースの時系列予測モデルは
  - MLP, RNN, CNN, Transformer が代表的
- その多くは、他分野からの応用
  - 時系列専用モデルはまだまだ発展
- 近年はTransformerベースの手法が盛んに研究されている。
  - そのうち、Informerについてハンズオン解説を行った。
- Transformerのような大規模モデルを活用するには、
  大規模時系列データセットが必要
  - よいモデルを作るためには、データ収集やドメイン知識が大切

© Copyright NTT Communications Corporation All Rights Reserved.

まとめ

© Copyright NTT Communications Corporation All Rights Reserved.

まとめ

今回は時系列データ分析の概略を紹介しました。

- データ収集、課題探索・設定
  - → 今回はオープンデータで予測
- データの可視化
- 前処理
- モデリング・評価
  - 線形モデル
  - Deep Learning モデル

© Copyright NTT Communications Corporation All Rights Reserved.

今日話せなかったこと

- データ収集、課題探索・設定
  - 分類、異常検知 etc...
- データの可視化
  - 周波数領域での解析, etc...
- 前処理
  - 次元削減 etc...
- モデリング・評価
  - スパースモデリング、因果探索
  - パラメーター探索

つづきは **ごちきか**: https://gochikika.ntt.com で！

© Copyright NTT Communications Corporation All Rights Reserved.

時系列の課題・トピック

予測モデル

MLP, CNN, RNN, Transformer

- 主にネットワーク構造を工夫
- 定常/非定常性の考慮
  - 例: 季節性/トレンド成分分解
- 長期記憶の改善
  - 例: 繰り返しパターンの学習

Uncertainty

- Conformal Prediction
- Probabilistic Forecasting

Online学習

表現学習

- 下流タスクに有利な特徴量(表現)を学習
- 得られた表現を使って簡単な線形モデルなどで十分な精度を出す

Random Conv. Kernel (ROCKET)

Contrastive Learning

ドメイン適応

- 特に時系列の文脈だと分布シフトに対処
- 元の学習データをソースドメイン, 学習済みモデルの適用先をターゲットドメインと呼ぶ

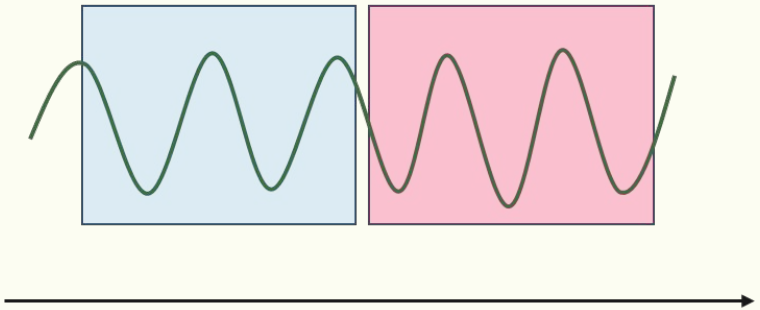
メタ学習

- 他のデータから学習の仕方を学習する手法。目的タスクにおいて、少数データしかなくても性能を高める。
- 特にFew-Shot Learningの文脈で語られることが多い
- 例: 工場A, B, Cのタスクから、新工場Dの少量データで学習したい

定常/非定常

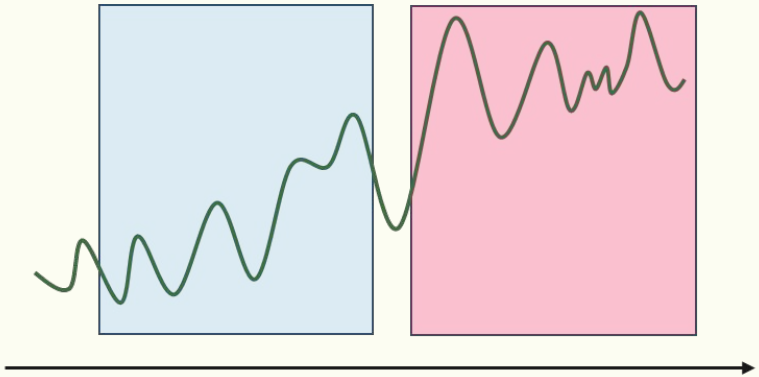
定常

統計量（平均分散自己相関）が
どこでも同じ



非定常

例えばトレンド性



- 多くの時系列は非定常
- 非定常性は分布シフトを引き起こす

© Copyright NTT Communications Corporation All Rights Reserved.

Node AI

IoTセンサーや売上などの時系列データの分析に特化

予測や異常検知への適用や因果分析、要因分析などの課題に対応

プログラミング・AIの習得が「難しい」

ブラウザ上で**ドラッグ&ドロップ**で分析フローを構築
無料で気軽に利用開始

AIプロジェクトの進め方が「難しい」

Node-AI上でデータを見ながら
コメントで議論、報告書も作成

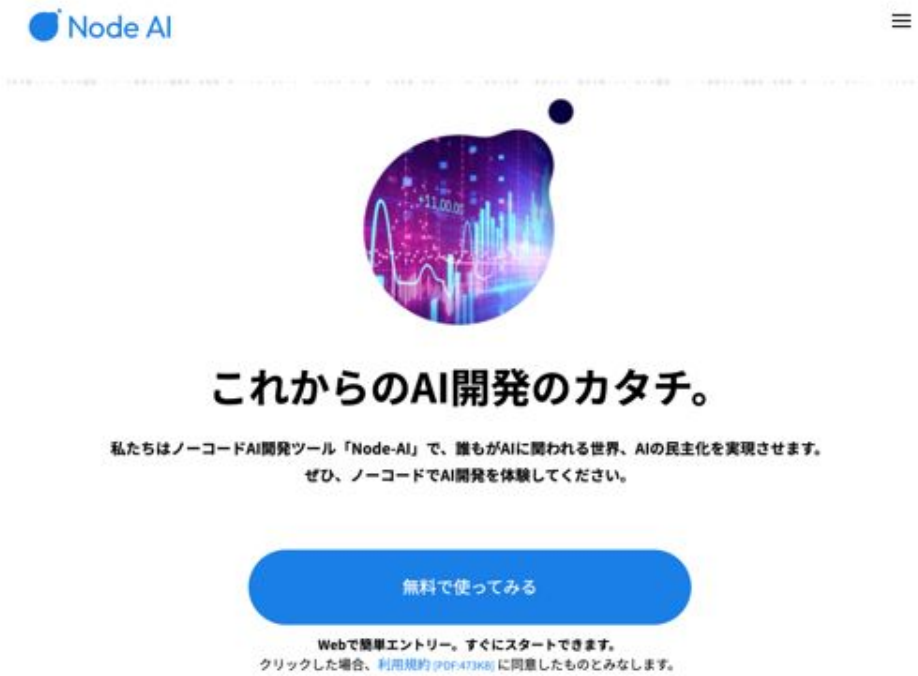
AIとシステム連携が「難しい」

開発したAIモデルを
簡単に**システム連携可能**

© NTT Communications Corporation All Rights Reserved.

Work all illustrations by Storyset

（new）現在無料ベータ版公開中



- ✓ 無料、期限なしで利用可能
- ✓ メールアドレスとパスワードを登録するだけ
- ✓ 「https://nodeai.io」にアクセス！

© NTT Communications Corporation All Rights Reserved.