KERAS を使った RNN による時系列データ モデリング

山崎和博
NVIDIA Deep Learning Solution Architect

AGENDA

系列データとRNN

Keras/Theano/Pandas/NumPy

ハンズオン: RNNによる時系列データ解析

本ラボの目的

- リカレントニューラルネットワーク(RNN)の基本を学ぶ
- 時系列データ解析へのRNNの適用方法を学ぶ
  - 電子健康記録(EHR)データを例題に
- データ解析の基本を体験する

前提とする知識

- 前提知識
  - ニューラルネットワークの基本

- あると望ましい知識
  - Pythonの文法

注意事項

データ利用条件

データの性質上、以下の条件に同意してラボを起動してください

- データセットのダウンロード、共有、転送、および第三者提供を含むいかなる活動も、このワークショップ/ラボにおいて使用を許可された範囲を超えて行うことはできません。
- すべてのユーザは、このデータの対象者や血縁者、雇用主、家族等を特定したり、接触したりするために、データセットに含まれる情報を使用しないことに同意します。

系列データとRNN

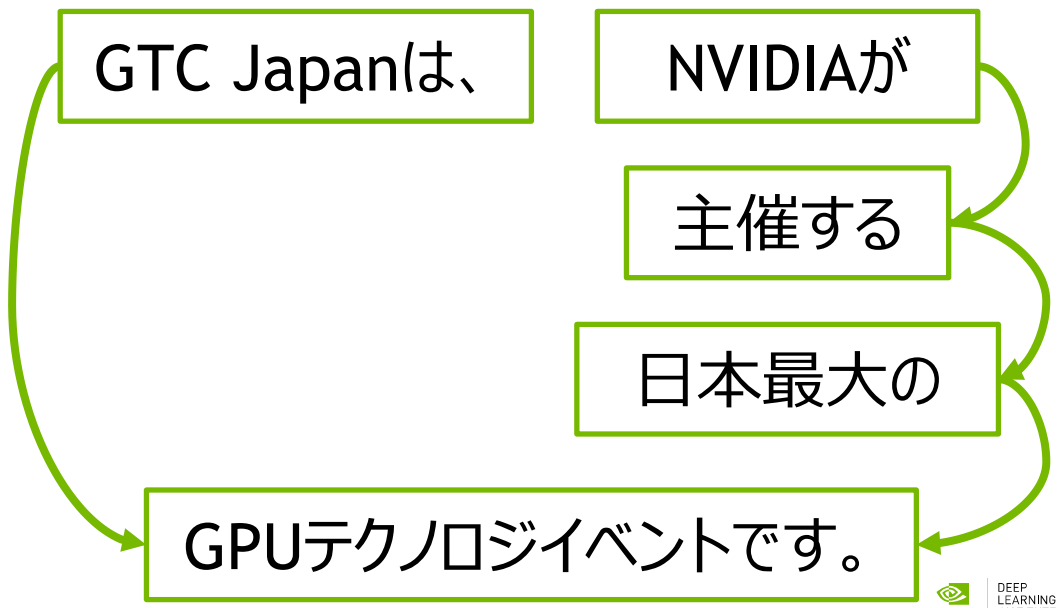
系列データ

要素の間に依存関係を持つようなデータ

時系列データ

言語列データ

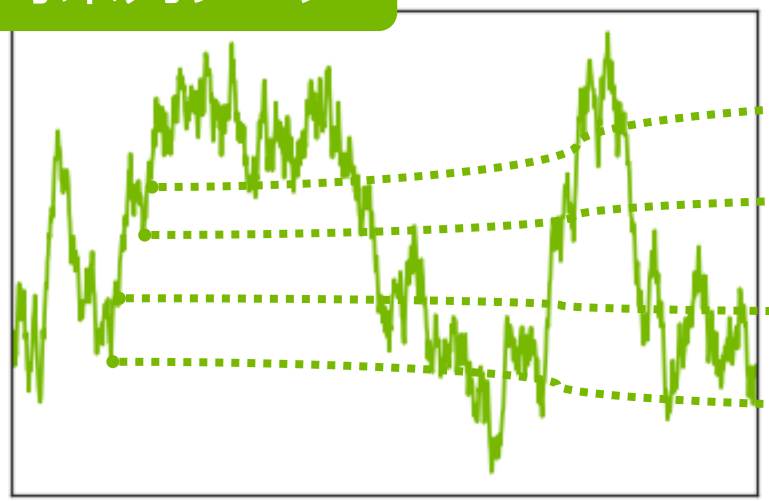
GTC Japanは、NVIDIA が主催する
日本最大の GPU テクノロジ イベントです。



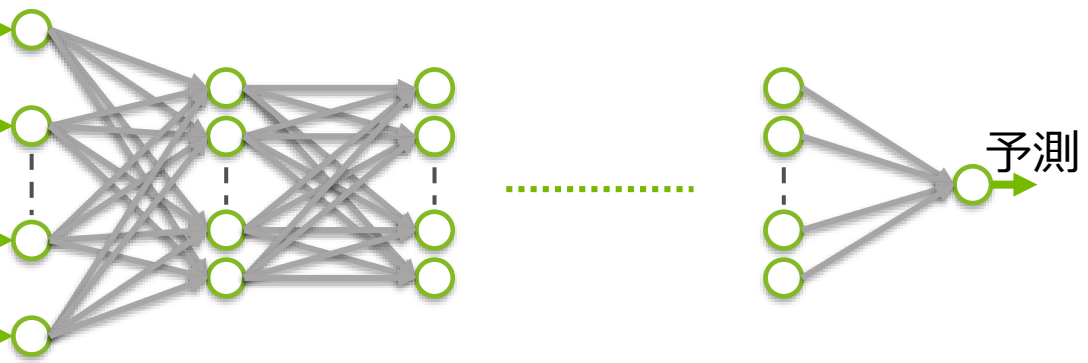
系列データ

要素の間に依存関係を持つようなデータ

時系列データ



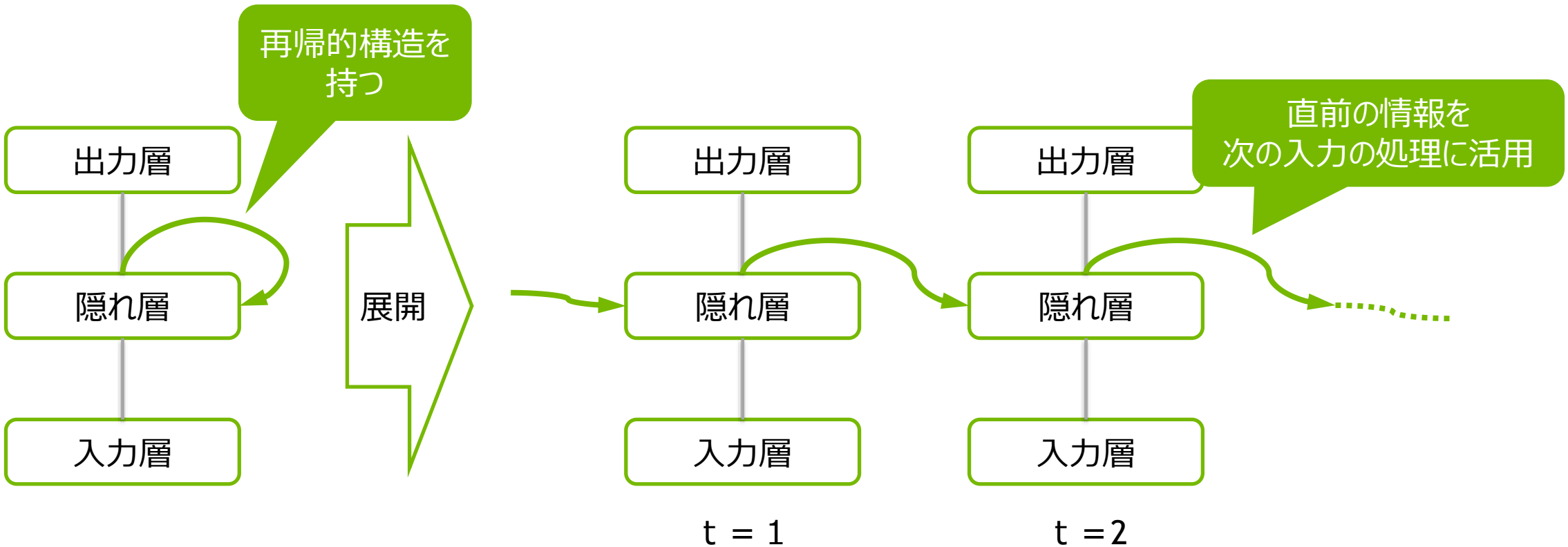
シンプルなNN



時間幅を固定して入力
→ 幅を超える**長期間の関係**を
捉えることが難しい

リカレントニューラルネットワーク: RNN

列として与えられるデータの依存関係を捉える



リカレントニューラルネットワーク: RNN

列として与えられるデータの依存関係を捉える

再帰的構造を持つ

直前の情報を活用

良い点

- 古典的なfeed-forward型ネットワークより自然に系列データを扱える

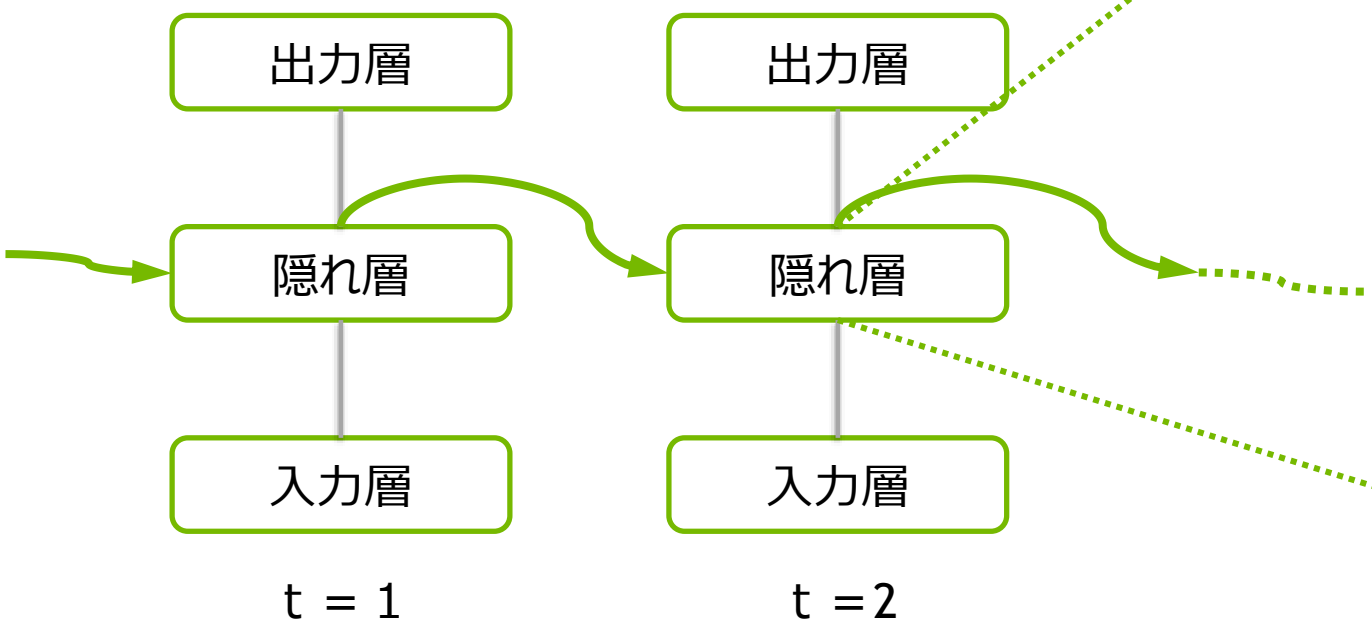
悪い点

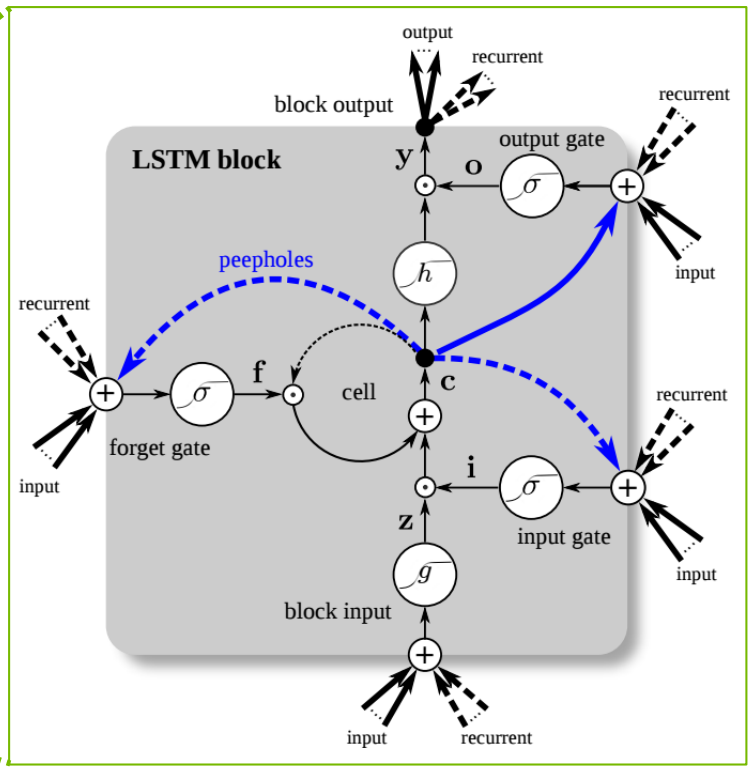
- あまりに長い系列はうまく学習できない(c.f. 勾配消失問題)

LONG SHORT-TERM MEMORY: LSTM

RNNの欠点を克服

RNNの隠れ層を
LSTM blockで置き換える

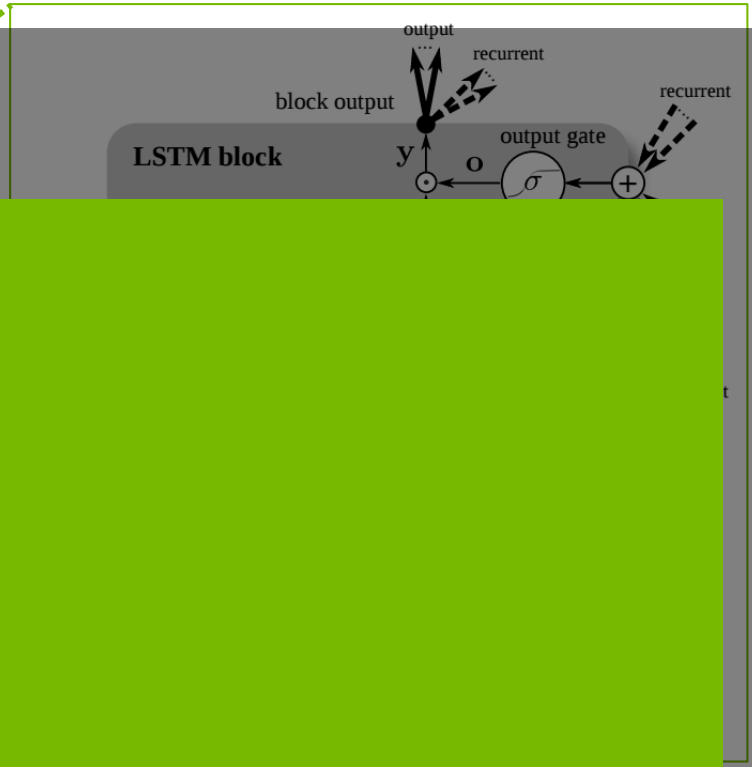




LONG SHORT-TERM MEMORY: LSTM

RNNの欠点を克服

RNNの隠れ層を
LSTM blockで置き換える



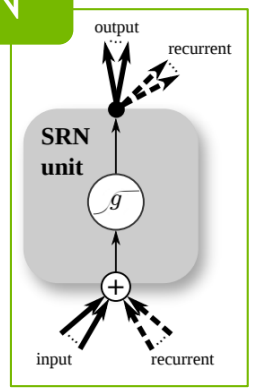
LSTMによって改善されたこと

- より長い系列を処理できるようになる
- 勾配消失問題を回避

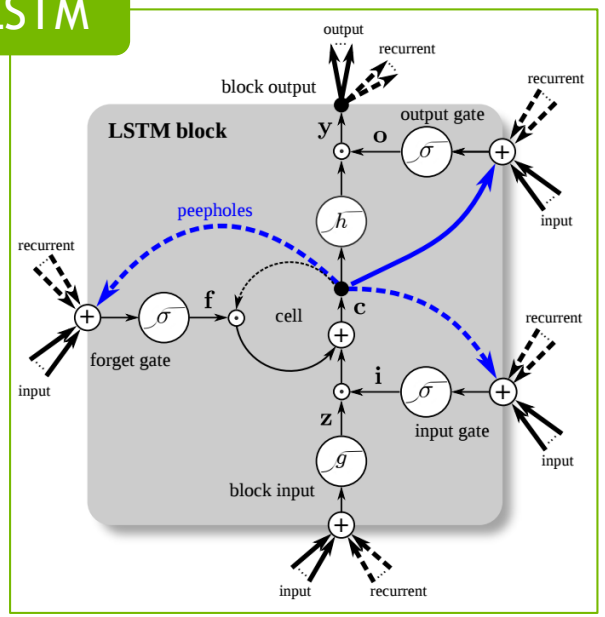
その他のRNN VARIANTS

LSTM以外の派生ネットワーク

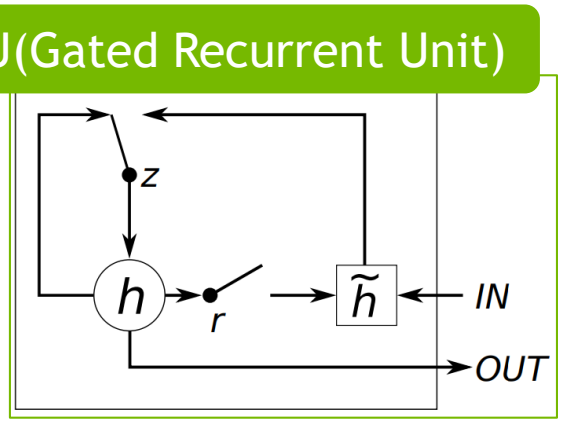
RNN



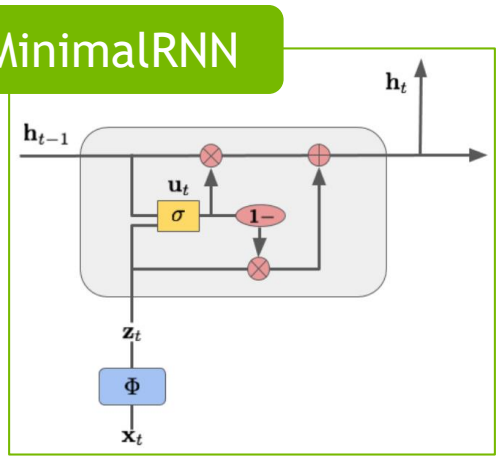
LSTM



GRU(Gated Recurrent Unit)



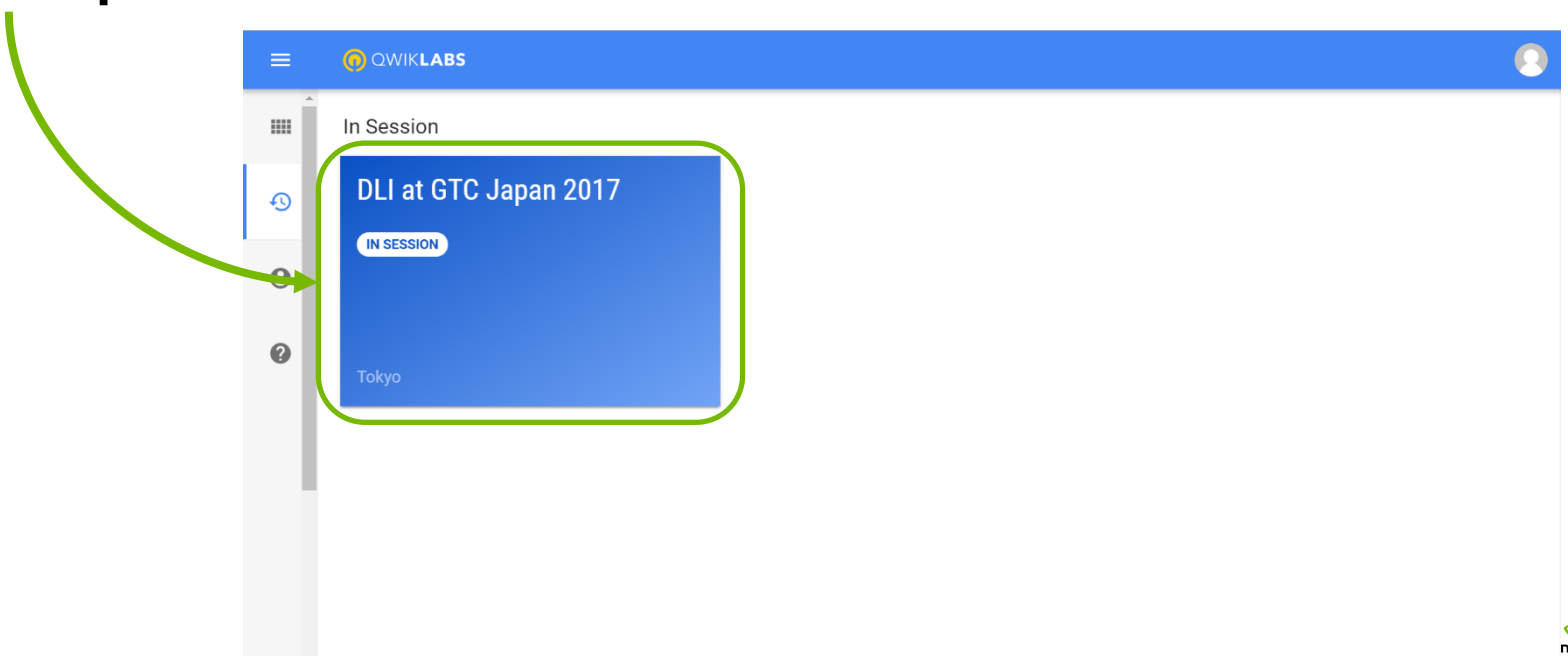
MinimalRNN



ラボのセットアップ

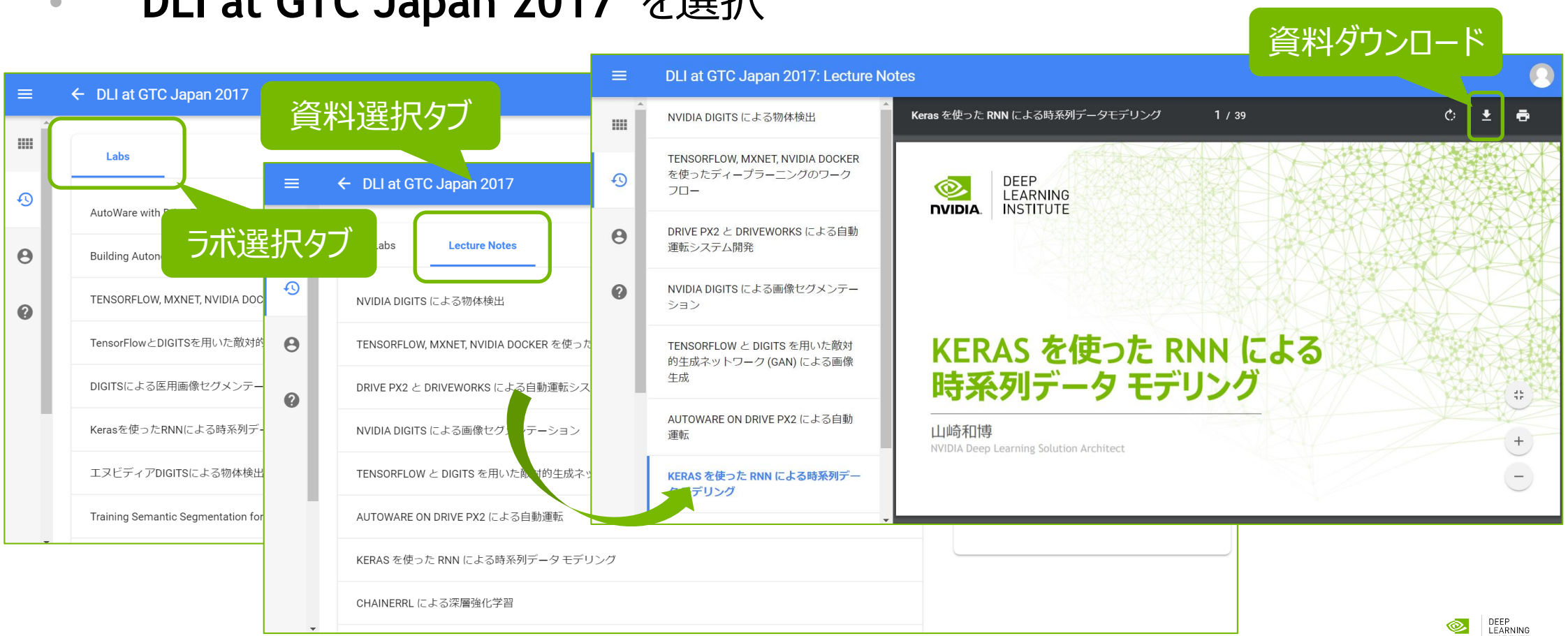
ログインからラボの選択まで

- https://nvlabs.qwiklab.com にアクセスし、ログイン
  - アカウントがなければ新規ユーザ作成
- **“DLI at GTC Japan 2017”を選択**



ログインからラボの選択まで

- “DLI at GTC Japan 2017”を選択



ログインからラボの選択まで

- **“Keras を使った RNN による時系列データ モデリング”**を選択
- ↓（画面遷移）
- “ラボを開始”ボタンを押す

ラボの流れ

EHRデータを対象として重篤度を予測する

1. セットアップ
    1. 各種ツールとデータの読み込み
2. データの準備
    1. データを概観
    2. 前処理(正規化、補間、系列長の調整)
3. KerasによるLSTMネットワークの構築、学習、比較評価

THEANO

オープンソースの老舗ディープラーニングフレームワーク

- モントリオール大のMILA(Montreal Institute for Learning Algorithms)でメンテナンス
  - **v1.0が最後のメジャーリリースとアナウンス (2017/9/28)**
- CUDAでGPUアクセラレートされたバックエンドにより高速
- Python bindingを持ち、NumPyなどとの連携を意識しているため、データ分析などとの結合が容易
- フロントエンドとしてLasagneやKerasなどのライブラリがあり、使いやすい

theano

deeplearning.net/software/theano/index.html
github.com/Theano/Theano

KERAS

各種フレームワークのフロントエンド

- GoogleのFrançois Chollet氏が開発
- 抽象化されたAPIを提供しており、さまざまなフレームワークをバックエンドとして切り替えて使用できる
  - Theano、TensorFlow、CNTK、MxNetなどをサポート
- Pythonで記述されており、実装が容易
  - 互換性も、Python 2.7 – 3.6までをサポート
- CPUだけでなくGPUでも動作

https://keras.io
https://github.com/fchollet/keras

NUMPY

科学技術計算向けライブラリ

- アカデミア、企業を問わず広く用いられている
- 大きなN次元配列や行列を扱うための機能を提供
- 数学演算の関数が充実している
- 関連するライブラリ
  - SciPy: より高度な科学技術計算向け
  - Scikit-learn: 機械学習パッケージ
  - Matplotlib: グラフの描画用ライブラリ

http://www.numpy.org/
https://github.com/numpy/numpy

PANDAS

データ分析のためのツールキット

- アカデミア、企業を問わず広く用いられている
- 高速かつ、効率的にデータを扱うための、DataFrame オブジェクトを有する
- 多様なデータ形式に対応
  - CSV, Microsoft Excel, SQLデータベース, HDF5
- NumPyとの連携がシームレス

本日使用する形式

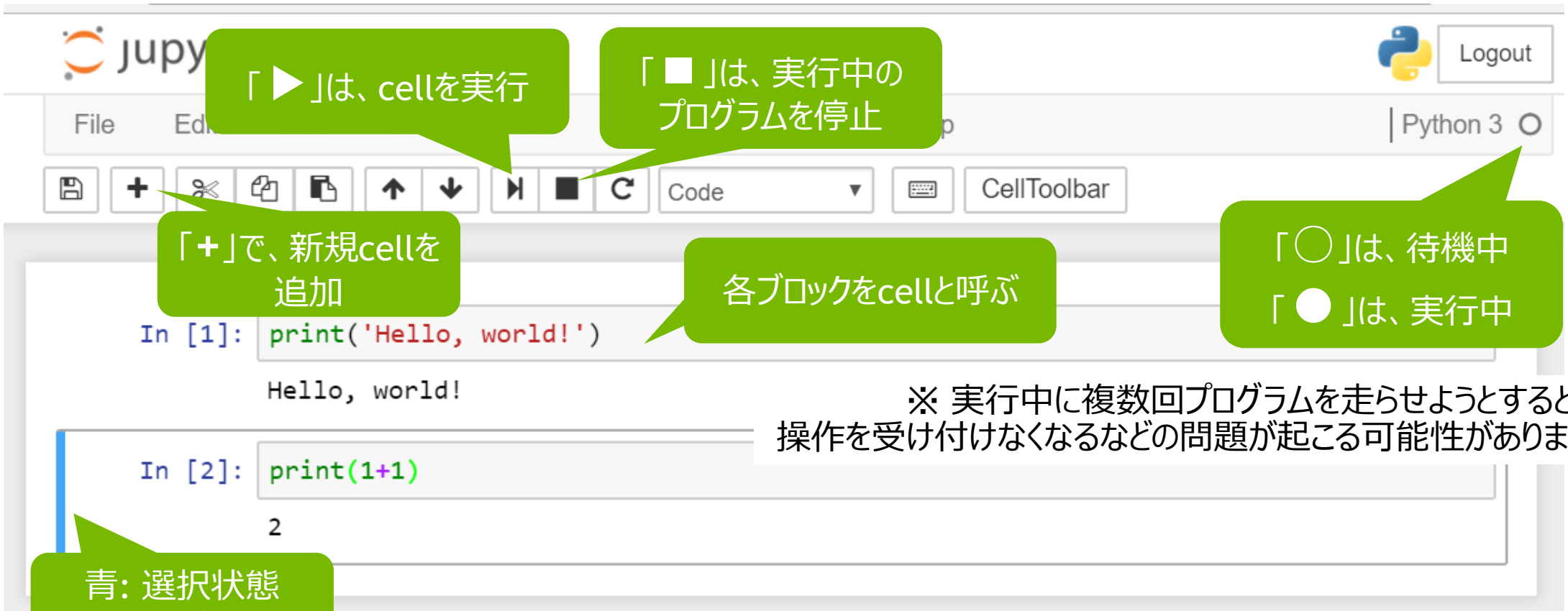
pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$

https://pandas.pydata.org/
https://github.com/pandas-dev/pandas

ラボの流れ

Jupyter notebookの使い方



※ 実行中に複数回プログラムを走らせようとすると、
操作を受け付けなくなるなどの問題が起こる可能性があります

ラボ開始


Kerasを使ったRNNによる時系列データモデリング
© Accelerated Compute Cloud Training の使用について Yamasaki Kazuhiro
5 セットアップ · 115 アクセス時間 · 90 完了
ラボの評価
ラボの詳細
ラボのランニング
ラボを終了
01:54:40
CONNECTION DETAILS
パスワード
V9FLdT8QY
LAUNCH LAB
HostDNS
ec2-35-153-200-159.compu
InstanceId
i-0d45cad96af14c200
Connection
ubuntu@ec2-35-153-200-1
ここをクリック

ハンズオン: EHRデータから重篤度を予測

ここからはJupyter notebookを中心に進めます

#1: セットアップ

Tips: Pandasのデータロード関数

Input/Output API(http://pandas.pydata.org/pandas-docs/stable/api.html#input-output)

- `read_csv()`
- `read_excel()`
- `read_sql()`
- **`read_hdf()`**
- ……

#2: PANDASの使い方

Tips: データの参照方法

DataFrameオブジェクトの中身(例)

d.columns

	Age	Heart rate	Weight	PulseOximetry
1a	11.691097	97.0	37.5	100.0
2b	11.691126	88.0	NaN	98.0
3c	11.691154	85.0	NaN	99.0
4d	11.691183	NaN	NaN	NaN
5e	11.691212	64.0	NaN	99.0
…				

d.index

#2: PANDASの使い方

Tips: データの参照方法

DataFrameオブジェクトの中(例)

d.ix[‘2b’, ‘Heart rate’]

	Age	Heart rate	Weight	PulseOximetry
1a	11.69	97.0	37.5	100.0
2b	11.691126	88.0	NaN	
3c	11.691154	85.0	NaN	99.0
4d	11.691183	NaN	NaN	NaN
5e	11.691212	64.0	NaN	99.0

d[‘Weight’]

d.iloc[3]

#2: PANDASの使い方

Tips: データの参照方法

DataFrameオブジェクトの中(例)

d.index.level[0]

encounterID	absoluteTime	Age	Heart rate	Weight	PulseOximetry
8	1a	11.691097	97.0	37.5	100.0
	2b	11.691126	88.0	NaN	98.0
	3c	11.691154	85.0	NaN	99.0
	4d	11.691183	NaN	NaN	NaN
	5e	11.691212	64.0	NaN	99.0
…	…				

d.index.level[1]

#2: PANDASの使い方

Tips: データの参照方法

DataFrameオブジェクトの中(例)

d.loc[8]

encounterID	absoluteTime	Age	Heart rate	Weight	PulseOximetry
	1a		97.0	37.5	100.0
	2b	11.691126	88.		
8	3c	11.691154	85.0	NaN	99.0
	4d	11.691183	NaN	NaN	NaN
	5e	11.691212	64.0	NaN	99.0
…	…				

d.loc[8].index

d.loc[(8, '4d')]

#3: ROC曲線とAUC

Tips: 結果の評価方法

予測結果の評価例

#	1	2	3	4	5	6	7	8	9	10
正解	1	1	1	1	0	1	0	1	1	1
予測	1	1	1	1	1	1	1	1	1	1

正解率: 80%(8/10)

本当に良い精度なのか？

#3: ROC曲線とAUC

Tips: 結果の評価方法

予測結果の評価例

#	1	2	3	4	5	6	7	8	9	10
正解	1	1	1	1	0	1	0	1	1	1
予測	1	1	1	1	1	1	1	1	1	1

0が一つも
当たっていない

#3: ROC曲線とAUC

Tips: 結果の評価方法

予測結果の評価例

#	1	2	3	4	5	6	7	8	9	10
正解	1	1	1	1	0	1	0	1	1	1
予測	1	1	1	1	1	1	1	1	1	1

	正解が1	正解が0
予測が1	8(8 / 8 = 100%) => true positive	2(2 / 2 = 100%) => false positive
予測が0	0(0 / 8 = 0%) => false negative	0(0 / 2 = 0%) => true negative

#3: ROC曲線とAUC

Tips: 結果の評価方法

予測結果の例

#	1	2	3	4	5	6	7	8	9	10	…
正解	1	1	1	1	0	1	0	1	1	1	…
予測	0.9	0.8	0.88	0.91	0.56	0.51	0.9	0.76	1	0.68	…

予測結果の実体は
確率値

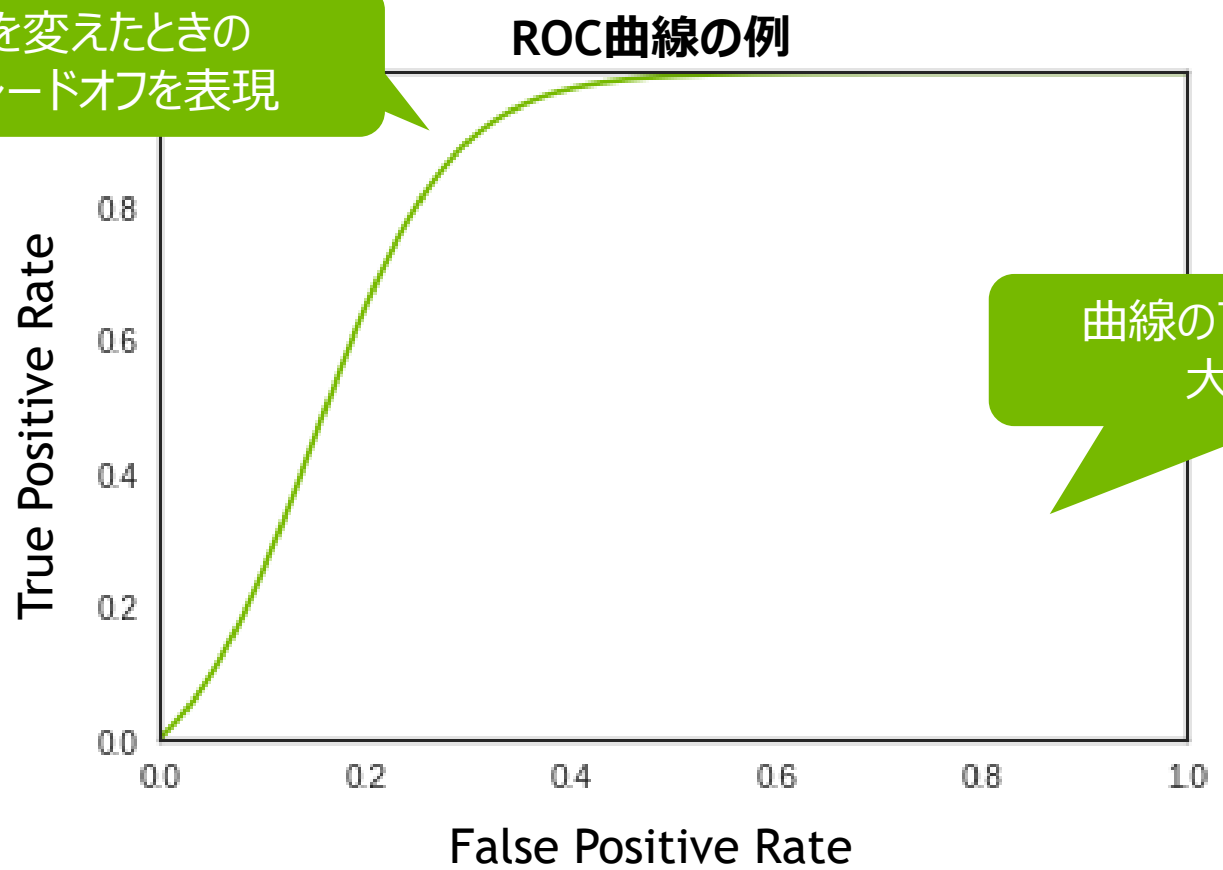
0.56をしきい値として判断すると、
正しく予測していた結果が間違うように

#	1	2	3	4	5						
正解	1	1	1	1	0	1		1	1	1	…
予測	1	1	1	1	0	0	1	1	1	1	…

#3: ROC曲線とAUC

Tips: 結果の評価方法

しきい値を変えたときの
性能のトレードオフを表現



曲線の下の面積(=AUC)が
大きい方が良い

まとめ

まとめ

RNNによる時系列データモデリング

- リカレントニューラルネットワーク(RNN)の基本を解説
  - LSTMなどの派生ネットワークについても紹介
- EHRデータを題材とした時系列データの解析を実施
  - 学習回数が少なくとも、既存の方法と遜色ない精度

www.nvidia.com/dli