

【分析の概要】 東京電力が公開する電気消費実績のデータを用いて、  
RNN/GRU/LSTMの3つのモデルで将来の電気消費量を予測する

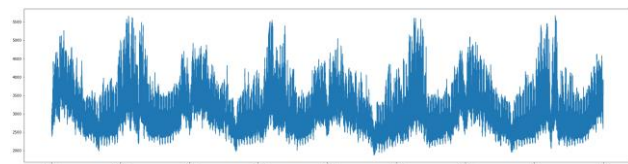
【分析の目的】 上記分析において**各モデルの精度変化の様子を比較**する

## 【データの取得】

1. 東京電力の公開する電気消費量の実績データを取得
2. 気象庁の公開する過去の気象データを取得  
(2018/01/01 01:00～ 2021/12/31 23:00)
3. “気温”、“降水量”、“天気”、“湿度”の列を追加
4. “月”、“日”、“時”、“曜日”の列を追加
5. 上記2つのデータを横に結合

※天気情報は3時間おきにしか観測がなかったため、  
直前直後の天気情報を複製して欠損値を補完

※「晴れや快晴→0」、「雨や雷→1」のように  
外出の妨げになる天気は1になるよう数値を変換

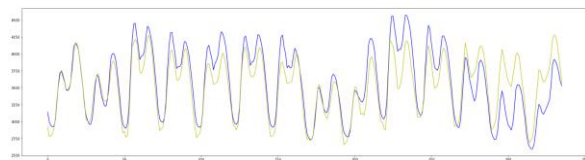
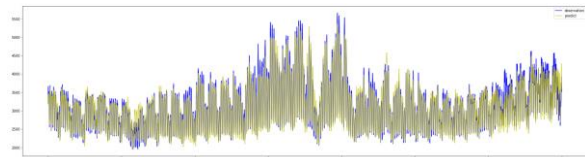
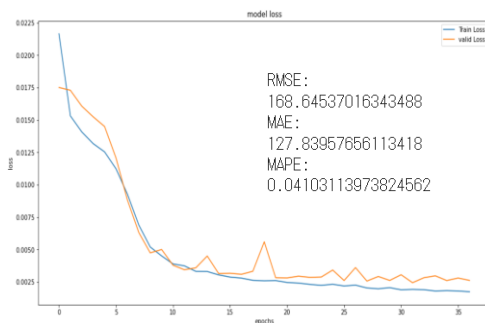


	date	time	kwh	datetime	month	day	hour	weekday	temperature	precipitation	weather	humidity
0	2018/1/1	1:00	2797	2018-01-01 01:00:00	1	1	1	0	1.5	0.0	1.0	83
1	2018/1/1	2:00	2669	2018-01-01 02:00:00	1	1	2	0	1.0	0.0	1.0	80
2	2018/1/1	3:00	2586	2018-01-01 03:00:00	1	1	3	0	1.2	0.0	1.0	85
3	2018/1/1	4:00	2543	2018-01-01 04:00:00	1	1	4	0	0.6	0.0	1.0	80
4	2018/1/1	5:00	2570	2018-01-01 05:00:00	1	1	5	0	1.9	0.0	1.0	78
...	...	...	...	...	...	...	...	...	...	...	...	...
35058	2021/12/31	19:00	3888	2021-12-31 19:00:00	12	31	19	4	1.5	0.0	2.0	50
35059	2021/12/31	20:00	3823	2021-12-31 20:00:00	12	31	20	4	1.0	0.0	1.0	44
35060	2021/12/31	21:00	3717	2021-12-31 21:00:00	12	31	21	4	1.4	0.0	1.0	47
35061	2021/12/31	22:00	3589	2021-12-31 22:00:00	12	31	22	4	0.7	0.0	1.0	43
35062	2021/12/31	23:00	3528	2021-12-31 23:00:00	12	31	23	4	0.6	0.0	1.0	44

35063 rows \* 12 columns

## 【結果の出力と評価】

1. 全てのモデルにおいて右図の4つを出力
2. 精度評価の指標は“RMSE”、“MAE”、“MAPE”の3つ
3. 予測の可視化としてテストデータ7,000件(約10カ月)の  
観測値、予測値をグラフにプロット
4. 同様に**末尾2週間**のグラフをプロット(黄色が予測値)



## 【結果】

作成したモデルにおいて最も精度が高かったのは  
**timesteps=12, unit数256のGRU**であった。  
(そのモデルの出力グラフの右下の2つ)

全学習を通して次のような傾向にあった。

**RNN**：計算は短いが精度に難あり

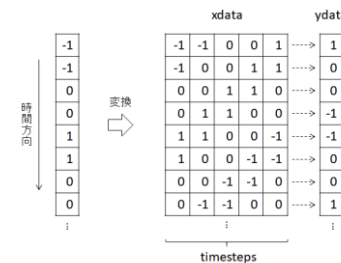
**GRU**：計算時間は中間で、精度が安定している

**LSTM**：計算が長く、複雑なモデルで精度が出る

## 【データ変形とモデル構築】

1. “timesteps”(学習の際に参照する直前の時間数)を6としてデータの形を3次元に変換

```
data_x = []
xarr = np.array
for i in range(timesteps, train_x.shape[0]):
    xset = []
    for j in range(train_x.shape[1]):
        d = train_x.iloc[i-timesteps:i, j]
        xset.append(d)
    xarr = np.array(xset).reshape(timesteps, train_x.shape[1])
    data_x.append(xarr)
x_train = np.array(data_x)
```



2. 上記のデータセットを学習するモデルを構築  
作成したモデルは以下のパターンの組み合わせ

• **RNN / GRU / LSTM**

• unit数が **64 / 128 / 128\*2(スタック) / 256**

上記12パターンのモデルをtimesteps=6, 12, 24の  
データセット全てに対して学習、予測を行う。  
(計36パターン)

model_LSTM_64.summary()		
Model: "sequential_1"		
Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 64)	18688
dropout (Dropout)	(None, 64)	0
dense (Dense)	(None, 1)	65
Total params: 18,753		
Trainable params: 18,753		
Non-trainable params: 0		

## 【モデルの比較と考察】

**全体の結論：今回使用したデータセットに対してはGRUの精度が最も高いという結果になった。**  
**しかしさらに長期データを用いる場合には層を増やしたLSTMが勝ると予想できる。**

### 【考察】

timestepsを増やした場合、**LSTMはよりモデルの複雑さを増すほど精度が高まる傾向**にあった。

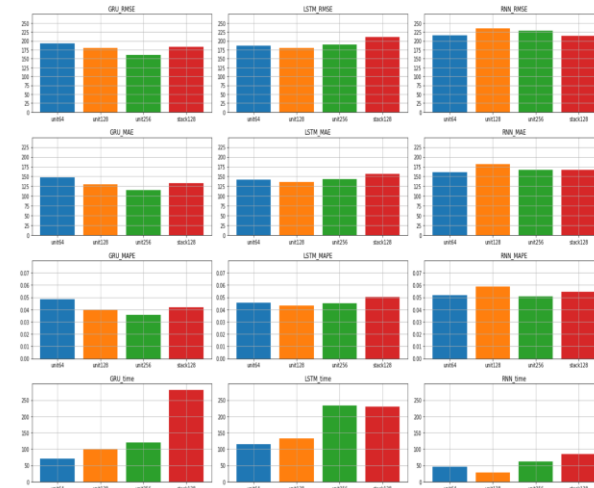
さらに長期のデータで学習を行う場合には、LSTMによって学習を行う方が精度が上がる可能性がある。

GRUはtimestepsが6と12で、unit256とスタックモデルの精度が逆転していたため、**timestepsによる調整によりさらに精度が上がる可能性**がある。

### 【今後の課題】

- 特徴量のバリエーションによる精度が未検証
- **現状では予測期間の天気情報がないと予測が不可能**
- 計算コストの高いモデルの検証が個人のPCだと限界あり

今後は予測を入力として受け取ることを繰り返して、  
未知の期間を予測できるモデルの作成を学習する予定です。



【timesteps=12の結果一覧】