

原因と対策の説明

1. 原因箇所

以下の関数の赤字(ループ回数を設定する所です)

```
def create_dataset(dataset, look_back=1):
    dataX, dataY = [], []
    for i in range(len(dataset)-look_back-1):
        xset = []
        for j in range(dataset.shape[1]):
            a = dataset[i:(i+look_back), j]
            xset.append(a)
        dataY.append(dataset[i + look_back, ])
        dataX.append(xset)
    return np.array(dataX), np.array(dataY)
```

2. 原因説明

今回の設定では、 $\text{len}(\text{dataset})=20$, $\text{look_back} = 8$ になっていますので、 $\text{len}(\text{dataset}) - \text{look_back} - 1 = 11$ になるので、`for i in range(len(dataset)-look_back-1)` のループでは 11 回のループになります。

Dataset の長さは 20 なので、dataset の値は t_0, t_1, \dots, t_{19} とします。そうすると、`create_dataset` を実行する結果、dataX の結果は以下のようになります

```
[[t0-t7],
 [t1-t8],
 [t10-t17]
]
```

この dataX を使って、予測すると、次のタイムスタンプを予測するので、 t_{18} に近い値を予測します。

この後、dataX の先端の値を削除して、予測データを dataX の最後に入れると、2 回目の予測で使う Dataset は以下のようなイメージになります。

$t_1, t_2, \dots, t_{19}, t_{18}$ (予測結果は t_{18} に近い値になるので、 t_{18} の記号を使います)。そうすると、dataset の規則が崩れ、予測値の結果は期待したものにはなっていないと考えられます。

3. 対策案

今回の要望”現時点での最新データを使い予測値を取得し、その値を使い”というものであれば、dataset の値は t_0, t_1, \dots, t_{19} の場合、create_dataset を実行する時の戻り値 dataX は以下のような必要があります。

```
[[t0-t7],  
 [t1-t8],  
 [t10-t17],  
 [t11-t18],  
 [t12-t19]  
]
```

そうすると、予測値は次のタイムスタンプ t_{20} に近い値を予測してくれますので、要望に対応できる形になります。

4. 対応ソース (赤字は対応する所です)

```
def create_dataset(dataset, look_back=1):  
    dataX, dataY = [], []  
    for i in range(len(dataset)-look_back+1):  
        xset = []  
        for j in range(dataset.shape[1]):  
            a = dataset[i:(i+look_back), j]  
            xset.append(a)  
            //dataY.append(dataset[i + look_back,])    #配列の index 外にアクセス  
            するので、エラーになるため、ここはコメントアウト。予測では、dataY を使って  
            いないため、影響がない  
        dataX.append(xset)  
    return np.array(dataX), np.array(dataY)
```