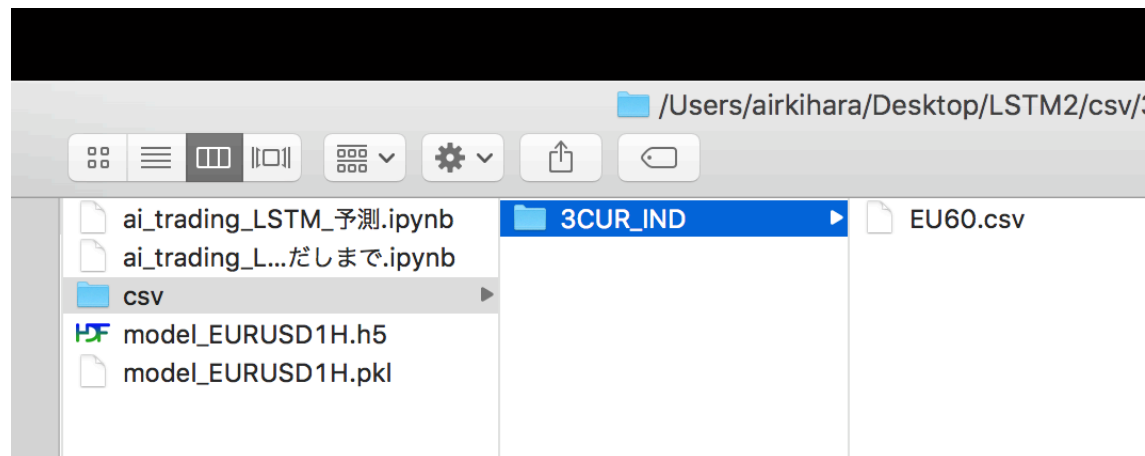


赤字のアンダーバー部分確認・作っていただきたい部分になります。

## ファイル説明



/csv/3CUR_IND/EU60.csv	データ
ai_trading_LSTM2.ipynb	モデルを作るソース
ai_trading_LSTM_予測.ipynb	前回お手伝いいただいたファイルの 予測版（ちょっと手が入っていますが基本的には同じモノです）このファイルを確認・修正していただきます。
model_EURUSD1H.h5	上記モデルを作った後に出来たモデル（=model.h5）をリネームしたもの
model_EURUSD1H.pkl	上記モデルを作った後に出来たモデル（=model.pkl）をリネームしたもの

## 今回の作業依頼詳細

前回依頼した内容は 0~12~0 の架空のデータをいれて試したモノでしたが、今回ものは EURUSD(ユーロ/ドル)の為替データを使うモノです。ai\_trading\_LSTM2.ipynb がモデルを生成します。

ai\_trading\_LSTM\_予測.ipynb

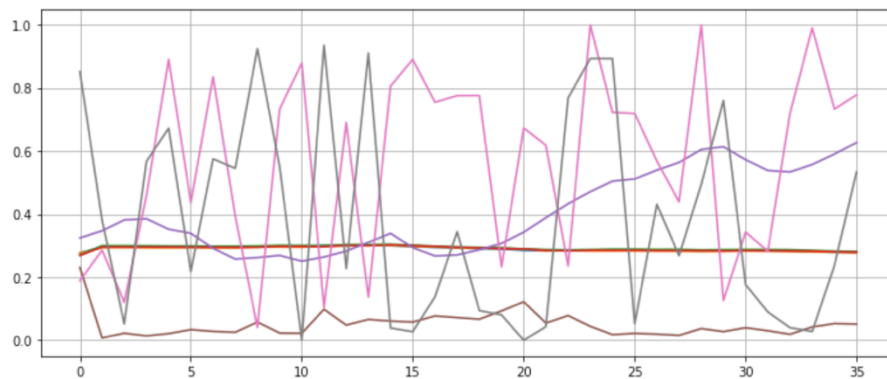
In[10]あたりでは、試しに csv データの最後の 36 件を表示します。36 件の csv データのプロットと、そのうちの末尾の 12 件分を予測値で予測したいと思っています。もともとモデルの精度は低いし、タイミングがわるく値動きが大きく動くタイミングですので、ますます予測が追従できないのはわかりますが、今回このソースでは、それを検証する目的で作りました。

以下のキャプチャは、末尾の 36 件と、前半の 24 件をならべてみたところです。このあと（右に出力されるであろう）12 件を推測します。

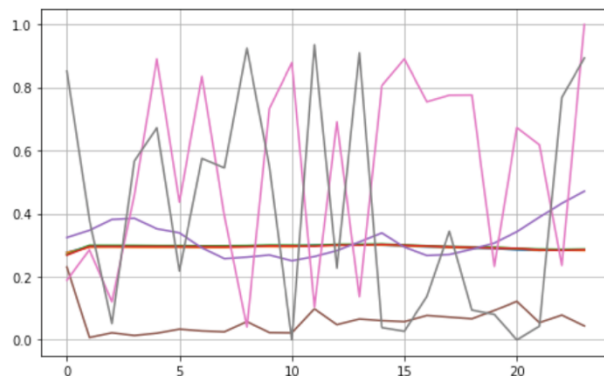


先日依頼した create\_dataset は修正して試していたのですが、下図の一番したの「予測値」というものが、この依頼したモノを当てはめてプロットしてみたのですが、大幅におかしな値が出てきています。モデルの精度の問題で、この値で正しいのであれば、これはこれで OK と教えていただければと思います。

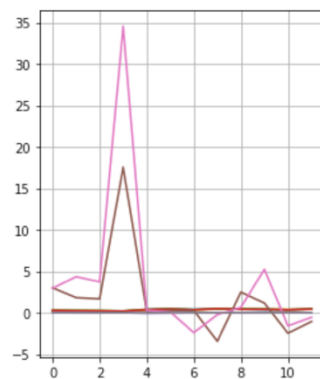
dataset 過去のNbar=36



予測値直前 = dataset 過去のNbar=-36~-12



予測値 = last\_result\_scale



上記画像の一番したの画像が予測した値（標準化）をプロットしたモノですが、2本先には大幅におかしくなっていますね。

これは仕方ないのか、プログラムのバグなのか？をはっきりさせたいです。

## 本当の理想型（できればここまでやっていただきたい）

CSV の N 本目から M 本先までを予測。M が 1 2 だとすると、

予測した値をチャートに出力するが、その際、

N 本目（N-1 を predict へ渡す）を予測し、合計 M 本を予測 →プロット

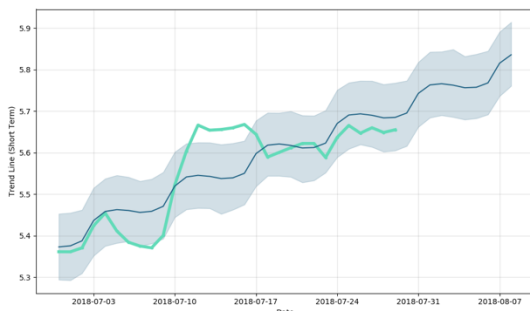
N+1 本目（N を predict へ渡す）を予測し、合計 M-1 本を予測 →プロット

N+2 本目（N+1 を predict へ渡す）を予測し、合計 M-2 本を予測 →プロット

:

と予測したときの N 迄のデータ + （連結） M 予測出力バーチャートをそれぞれ別画像（別プロット）で出力したいです。（1 枚のプロットで良い方法があればよいのですが）後半になればなるほど、予測本数は少なくなり、このチャートを見比べることで、M が何本くらいまでなら予測値として使えそうか？を調べるプロットとなります。実際は M=1～2 くらいしか使えないのではないかとと思っています。

N は任意で変更できるといいです。どの値を表示するかは、0～3 列（Close/High/Low）の中から一つまたはすべてを選びたいですね（Open はあまり気にしていません。からむ設定はハードコーディングで OK です）



この薄い網掛け（背景グレー）部分が 予測した High/Low であったり・・・どんな方法がいいのでしょうか。

見れば普通のラインのプロットで構いません。

ai\_trading\_LSTM2.ipynb の一番下に表示される pyplot ローソクバーに予測値を線で追加していますが、この方法でもかまいません。（理想はこちらかもしれませんが）、M 件を同表記するか・・・チャートを分けるしかなさそうですね。台風の進路情報のように濃淡で予測範囲がみえるなんてのもりそうです。

## データについての補足

Csv ファイルは以下の項目で出来ています。ユーロドルの 1 時間データを使っています。

'Close','High','Low','Open','ADXadx','HighLowDiff','OpenRelativePosition','CloseRelativePosition'

Close/High/Low/Open はそれぞれ 終値・高値・安値・初値となっております。その後の ADX は統計値（インジケータの値）。

ここまでの値を Dense(5)として出力値として model を作ってあります。

'HighLowDiff','OpenRelativePosition','CloseRelativePosition' は私オリジナルの値でして、ローソク足の形を示しています。

'HighLowDiff' : 高値－安値 を 0 ～ 1. 0 で表現

'OpenRelativePosition' : 高値－安値を 1. 0 として、初値がどのあたりだったか

'CloseRelativePosition' : 高値－安値を 1. 0 として、終値がどのあたりだったか

を示します。予測では、'Close','High','Low','Open','ADXadx' ここまでしか予測していませんので、

'HighLowDiff','OpenRelativePosition','CloseRelativePosition'これらの値は、予測前に「予測値から計算」する必要があります。

```
In [11]: 1 for i in range(0,input_bars_for_predicate):
2
3     dataset_tmp,dataset_nouse = create_dataset(dataset_input,look_back)
4     print(dataset_tmp.shape)
5     dataset_tmp = dataset_tmp[-1]
6     # dataset_tmp = dataset_tmp.reshape(1,look_back,len(dataset_tmp))
7     dataset_tmp = dataset_tmp.reshape(1,len(dataset_tmp),look_back)
8
9
10
11
12     predict_tmp = model.predict(dataset_tmp) #,batch_size=1,steps=None)
13
14     print("")
15     print('----- 予測のために与える値(標準値化後) -----')
16     print(dataset_tmp.shape)
17     print(dataset_tmp[-5,0])
18
19     print('----- 予測値(標準値) -----')
20     print(predict_tmp.shape)
21     print(predict_tmp[-5,0])
22     result_for_scale = np.hstack([predict_tmp, np.zeros([predict_tmp.shape[0],len(dataset_input[0])-input_output_columns ])] ) #
23     print(result_for_scale.shape)
24
25     # 予測値をデータとしていれられるよう、足らない右N (3列) を作る。 0列目と同じ値 (行は同じ数字になるはず)
26     #result_for_scale[-1,4] = result_for_scale[-1,0]
27     #result_for_scale[-1,5] = result_for_scale[-1,0]
28     #result_for_scale[-1,6] = result_for_scale[-1,0]
29
30     # 'Close','High','Low','Open',
31     result_for_scale[-1,4]=result_for_scale[-1,1]-result_for_scale[-1,2]
32     result_for_scale[-1,5]=((2*(result_for_scale[-1,3]))-result_for_scale[-1,1]-result_for_scale[-1,2])/(result_for_scale[-1,1]-(result
33     result_for_scale[-1,6]=((2*(result_for_scale[-1,0]))-result_for_scale[-1,1]-result_for_scale[-1,2])/(result_for_scale[-1,1]-(result
34
35
36     # 次の予測のための準備：先頭を消し
37     # dataset_input = np.delete(dataset_input,0,axis=0)
38     # 次の予測のための準備：末尾に予測値をセット
39
40     print(future_result.shape)
41     print(predict_tmp.shape)
42
43     dataset_input = np.vstack((dataset_input,result_for_scale[-1,]))
44
45     future_result= np.vstack((future_result,predict_tmp))
46
47
```

In[11]の 30～32 行がその計算です。

HighLowDiff = High-Low

OpenRelativePosition=((2\*(Open))-High-Low)/(High-(Low))

CloseRelativePosition = ((2\*(Close))-High-Low)/(High-Low))

## 蛇足

予測値を使って予測する場合は、すべてのデータを列を予測出力にするのが普通なのでしょうか？