코로나 확진자 수 예측

송선영

1. 목적: 코로나 확진자 수 예측
2. 데이터 소개 및 진행 상황 (https://www.kaggle.com/darryldias/corona-virus-update-dd26?select=time_series_covid19_confirmed_global.csv)

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	7/18/21	7/19/21	7/20/21	7/21/21	7/22/21	7/23/2
0	NaN	Afghanistan	33.93911	67.709953	0	0	0	0	0	0	...	137853	141489	142414	142414	143183	14343
1	NaN	Albania	41.15330	20.168300	0	0	0	0	0	0	...	132686	132697	132740	132763	132797	13282
2	NaN	Algeria	28.03390	1.659600	0	0	0	0	0	0	...	153309	154486	155784	157005	158213	15956
3	NaN	Andorra	42.50630	1.521800	0	0	0	0	0	0	...	14273	14359	14379	14379	14464	1449
4	NaN	Angola	-11.20270	17.873900	0	0	0	0	0	0	...	40805	40906	41061	41227	41405	4162

날짜	확진
2020-01-22	1
2020-01-23	1
2020-01-24	2
2020-01-25	2
2020-01-26	3
...	...
2021-07-30	198345
2021-07-31	199787
2021-08-01	201002
2021-08-02	202203
2021-08-03	203926

한국만 뽑은 df (누적)

날짜	확진
2020-01-22	1
2020-01-23	0
2020-01-24	1
2020-01-25	0
2020-01-26	1
...	...
2021-07-30	1539
2021-07-31	1442
2021-08-01	1215
2021-08-02	1201
2021-08-03	1723

한국만 뽑은 df (일일)

날짜	확진
2020-01-22	0.000527
2020-01-23	0.000000
2020-01-24	0.000527
2020-01-25	0.000000
2020-01-26	0.000527
...	...
2021-07-30	0.811709
2021-07-31	0.760549
2021-08-01	0.640823
2021-08-02	0.633439
2021-08-03	0.908755

값을 0~1 사이의 값으로 정규화
MinMaxScaler 사용

데이터 전처리

1. 한국만 따로 뽑은 누적 확진자 수 data

```
corona_df_korea = corona_df[corona_df['Country/Region'] == 'Korea, South'].T[4:]

corona_df_korea = corona_df_korea.reset_index().rename(columns={'index':'날짜', 160:'확진'})
corona_df_korea['날짜'] = pd.to_datetime(corona_df_korea['날짜'])

# 데이터에 오류가 있어서 바꿔줌
corona_df_korea['확진'][551] = 191531

corona_df_korea.index = corona_df_korea['날짜']
corona_df_korea.set_index('날짜',inplace=True)
```

날짜	확진
2020-01-22	1
2020-01-23	1
2020-01-24	2
2020-01-25	2
2020-01-26	3
...	...
2021-07-30	198345
2021-07-31	199787
2021-08-01	201002
2021-08-02	202203
2021-08-03	203926

2. 한국 일일 확진자 수 data

```
# 일일 확진자 수
daily_corona_df = corona_df_korea.diff().fillna(corona_df_korea.iloc[0]).astype('int')
```

날짜	확진
2020-01-22	1
2020-01-23	0
2020-01-24	1
2020-01-25	0
2020-01-26	1
...	...
2021-07-30	1539
2021-07-31	1442
2021-08-01	1215
2021-08-02	1201
2021-08-03	1723

3. MinMaxScaler 사용하여 스케일링

```
from sklearn.preprocessing import MinMaxScaler

sc = MinMaxScaler()

daily_sc = sc.fit_transform(daily_corona_df)
daily_sc_df = pd.DataFrame(daily_sc, columns=['확진'], index=daily_corona_df.index)
```

날짜	확진
2020-01-22	0.000527
2020-01-23	0.000000
2020-01-24	0.000527
2020-01-25	0.000000
2020-01-26	0.000527
...	...
2021-07-30	0.811709
2021-07-31	0.760549
2021-08-01	0.640823
2021-08-02	0.633439
2021-08-03	0.908755

첫번째 시도

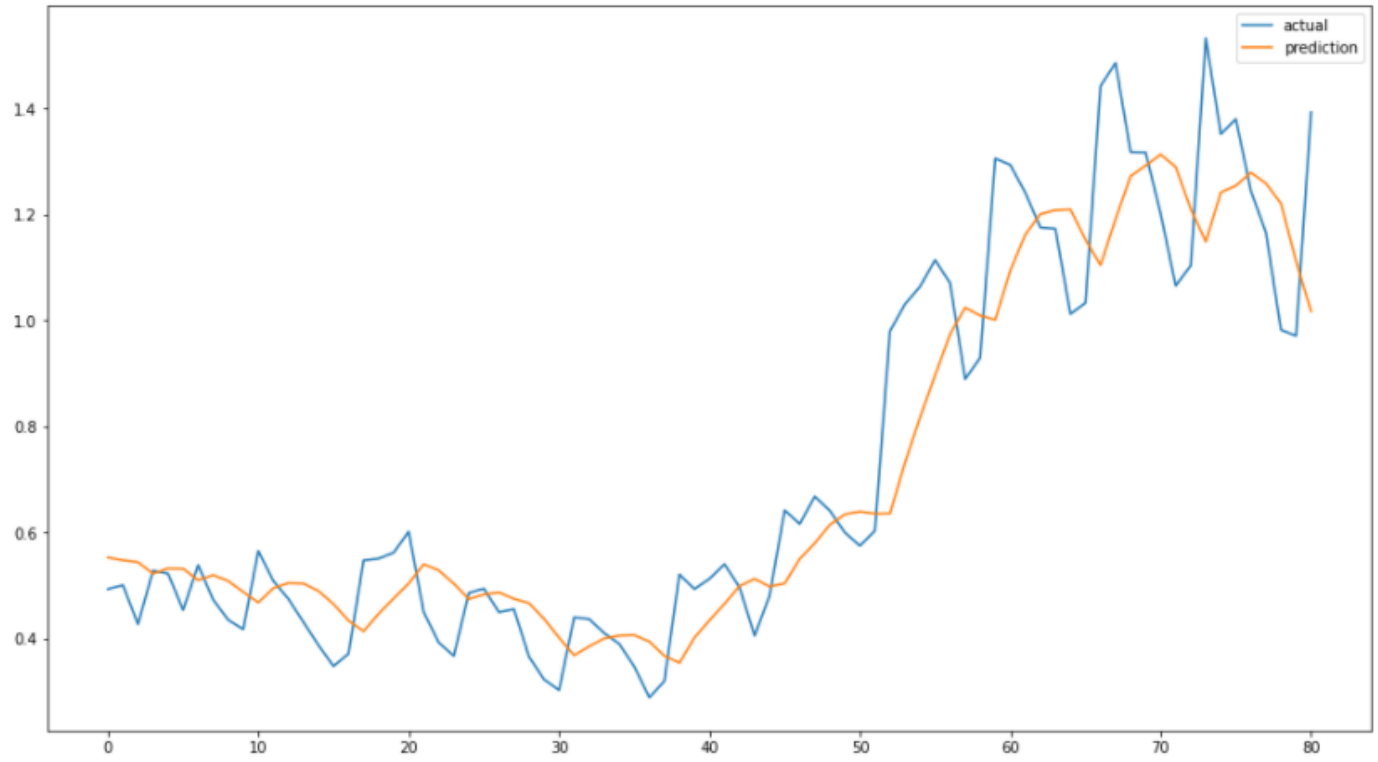
```
def make_dataset(data, seq_length):
    x = []
    y = []
    for i in range(len(data)-seq_length):
        x.append(np.array(data.iloc[i:(i+seq_length)]))
        y.append(np.array(data.iloc[i+seq_length]))
    return np.array(x), np.array(y)
```

```
# 14개씩
seq_length = 14
X_train,y_train = make_dataset(train_sc_df, seq_length)
X_test,y_test = make_dataset(test_sc_df, seq_length)
```

```
X_train.shape
```

```
(421, 14, 1)
```

```
model = Sequential()
model.add(LSTM(32, input_shape=(seq_length,1), activation='relu'))
model.add(Dense(14))
model.add(Dense(1))

model.compile(optimizer='adam',
              loss='mse',
              metrics=['acc'])

early_stop = EarlyStopping(monitor='loss',patience=1)

history = model.fit(X_train, y_train,
                    epochs=100,
                    batch_size=32,
                    callbacks=[early_stop])
```



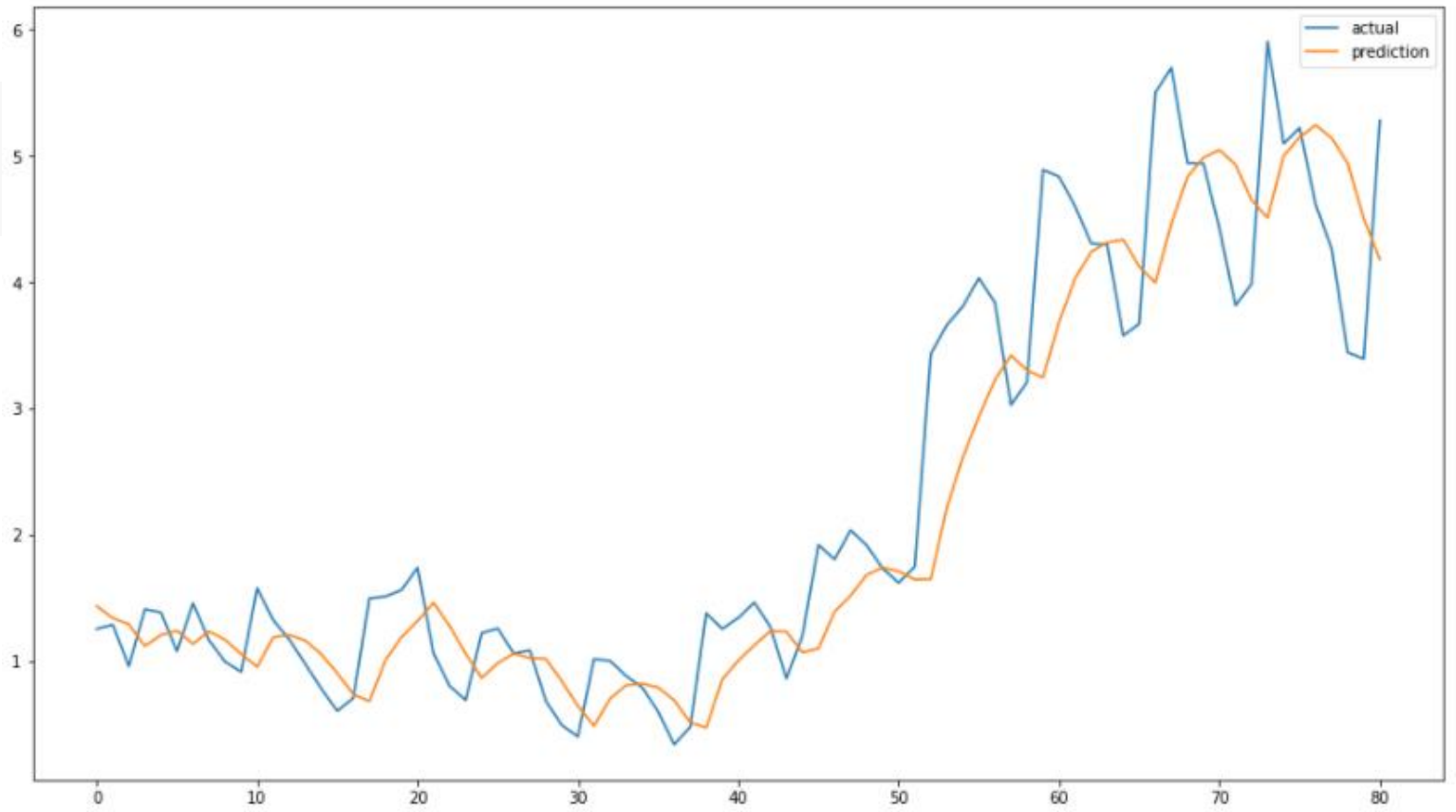
rmse: 0.13863178057138736

- 과대적합을 방지하기 위해 EarlyStopping 콜백 사용 (과대적합이 시작될때 훈련 중지)

두번째 시도

```
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

daily_sc = sc.fit_transform(daily_corona_df)
daily_sc_df = pd.DataFrame(daily_sc, columns=['확진'], index=daily_corona_df.index)
```

```
model = Sequential()
model.add(LSTM(32, input_shape=(seq_length,1), activation='relu'))
model.add(Dense(14))
model.add(Dense(1))

model.compile(optimizer='adam',
              loss='mse',
              metrics=['acc'])

early_stop = EarlyStopping(monitor='loss',patience=1)

history = model.fit(X_train, y_train,
                    epochs=100,
                    batch_size=32,
                    callbacks=[early_stop])
```



rmse: 0.644732030575773

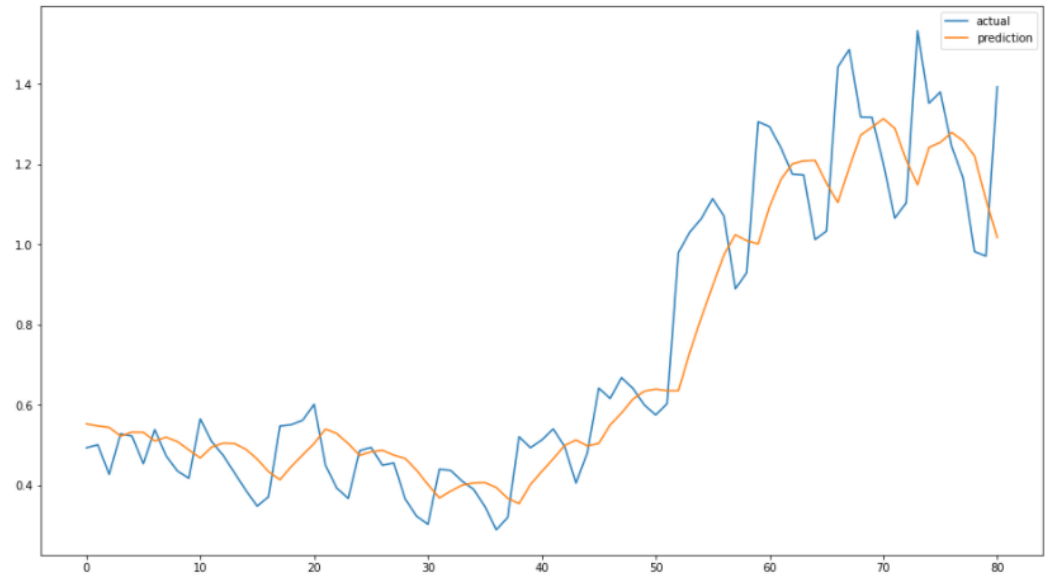
- 데이터 정규화 방법을 MinMaxScaler에서 StandardScaler로 변경

세번째 시도

```
model = Sequential()
model.add(LSTM(32, input_shape=(seq_length,1), activation='relu'))
model.add(Dense(14))
model.add(Dense(1))

model.compile(optimizer='adam',
              loss='mae',
              metrics=['acc'])

early_stop = EarlyStopping(monitor='loss',patience=1)

history = model.fit(X_train, y_train,
                    epochs=100,
                    batch_size=32,
                    callbacks=[early_stop])
```

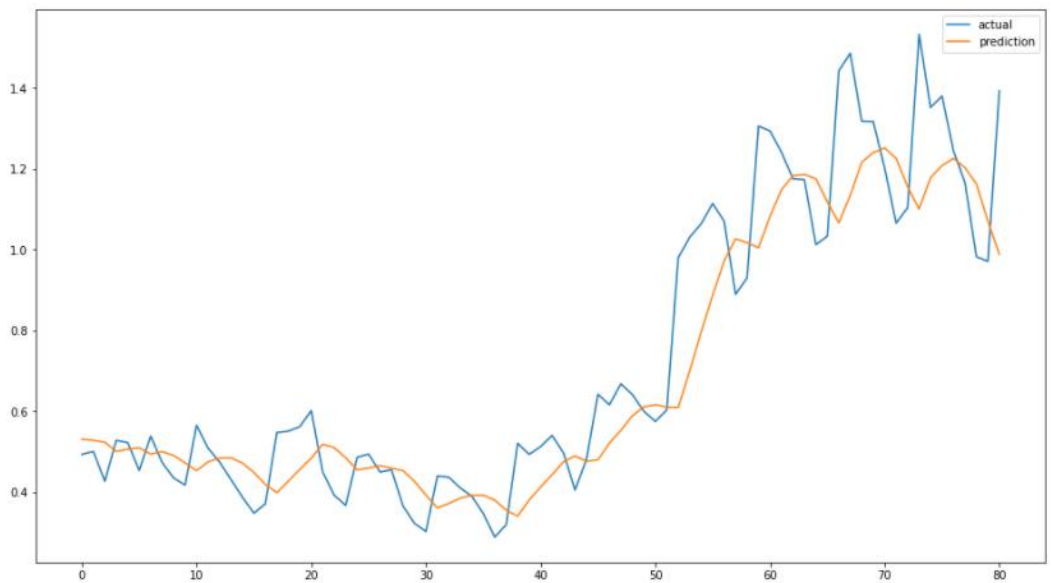
손실함수를
mse (mean_squared_error)
-> mae (mean_absolute_error) 로 변경

첫번째 시도 그래프



→ mse 사용

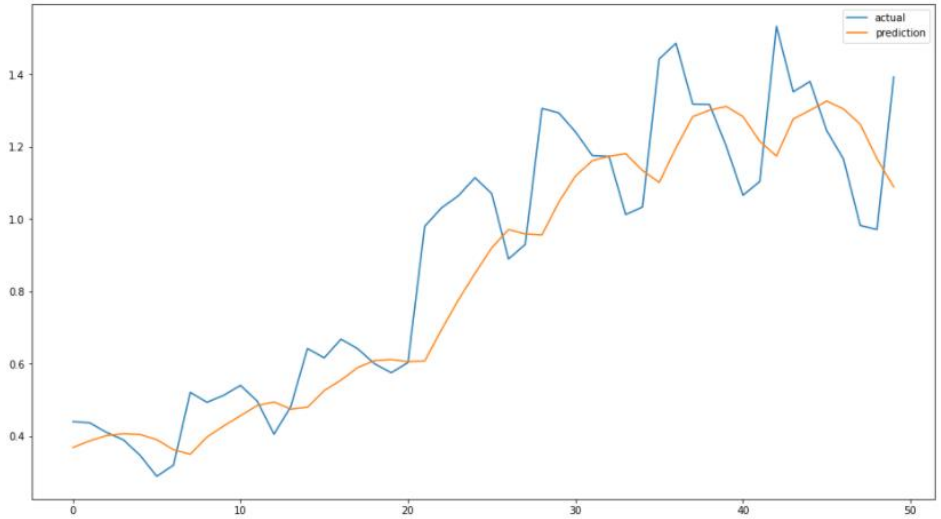
rmse: 0.13863178057138736



→ mae 사용

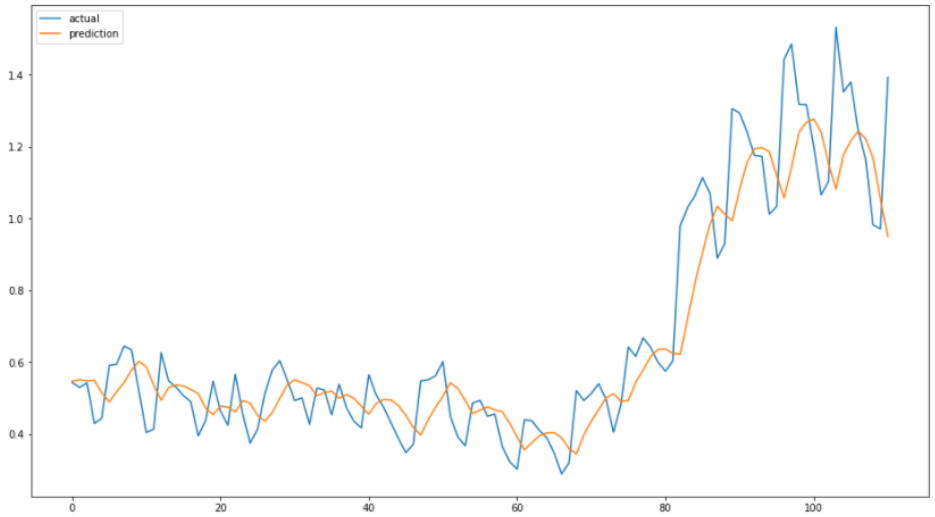
rmse: 0.14418629832144936

네번째 시도



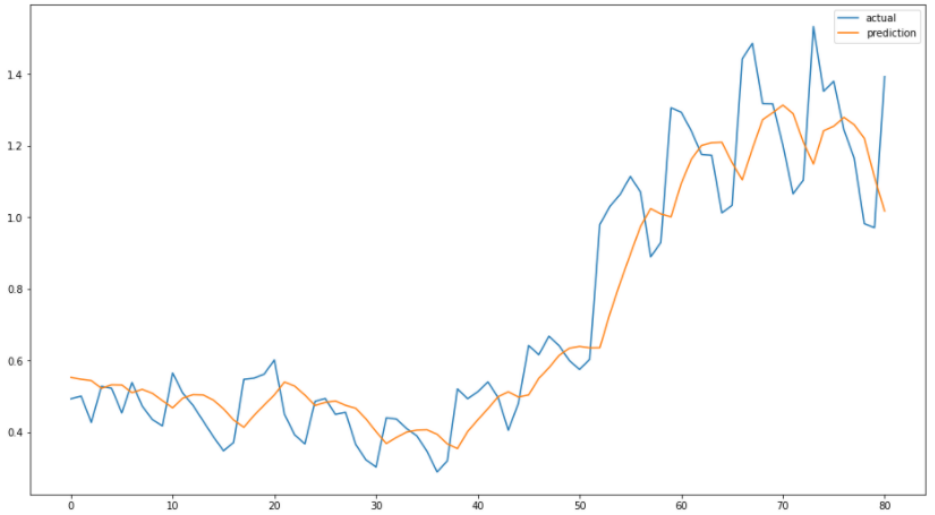
rmse: 0.1706752316205384

Train: 2020/1/22 ~ 2021/5/31
Test: 2021/6/1 ~ 2021/8/3



rmse: 0.13084277528780466

Train: 2020/1/22 ~ 2021/3/31
Test: 2021/4/1 ~ 2021/8/3



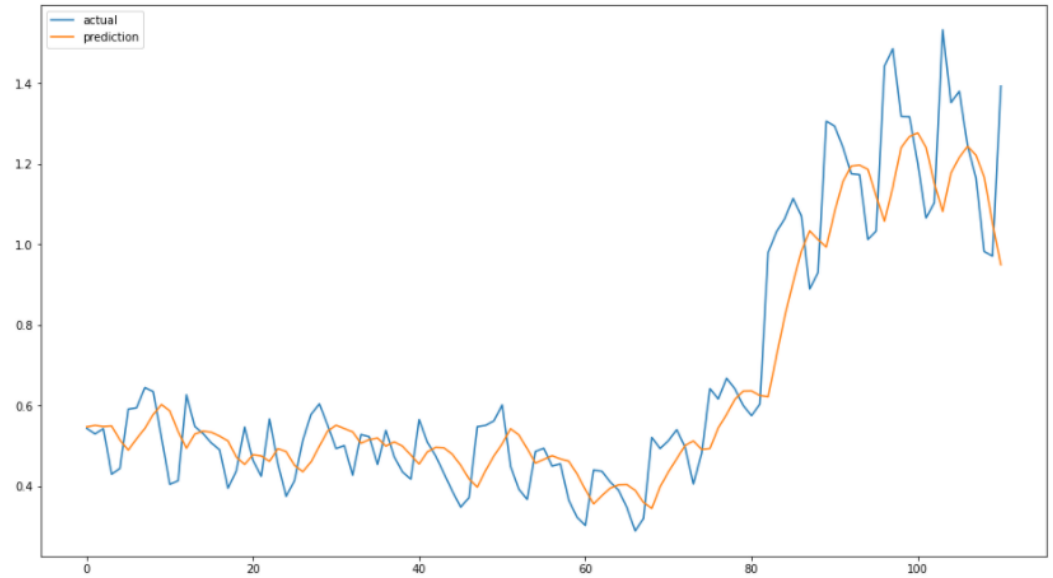
rmse: 0.13863178057138736

Train: 2020/1/22 ~ 2021/4/30
Test: 2021/5/1 ~ 2021/8/3

다섯번째 시도

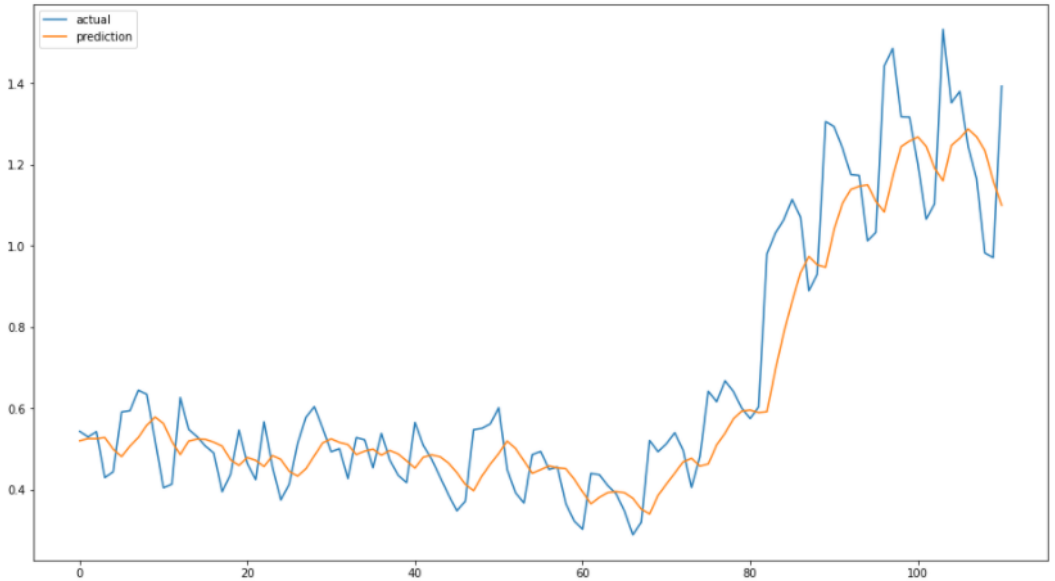
```
model = Sequential()
model.add(LSTM(32, input_shape=(seq_length,1), activation='tanh'))
model.add(Dense(14))
model.add(Dense(1))
model.compile(optimizer='adam',
              loss='mse',
              metrics=['acc'])

early_stop = EarlyStopping(monitor='loss',patience=1)

history = model.fit(X_train, y_train,
                    epochs=100,
                    batch_size=32,
                    callbacks=[early_stop])
```

첫번째 시도 그래프



→ relu 사용

rmse: 0.13084277528780466



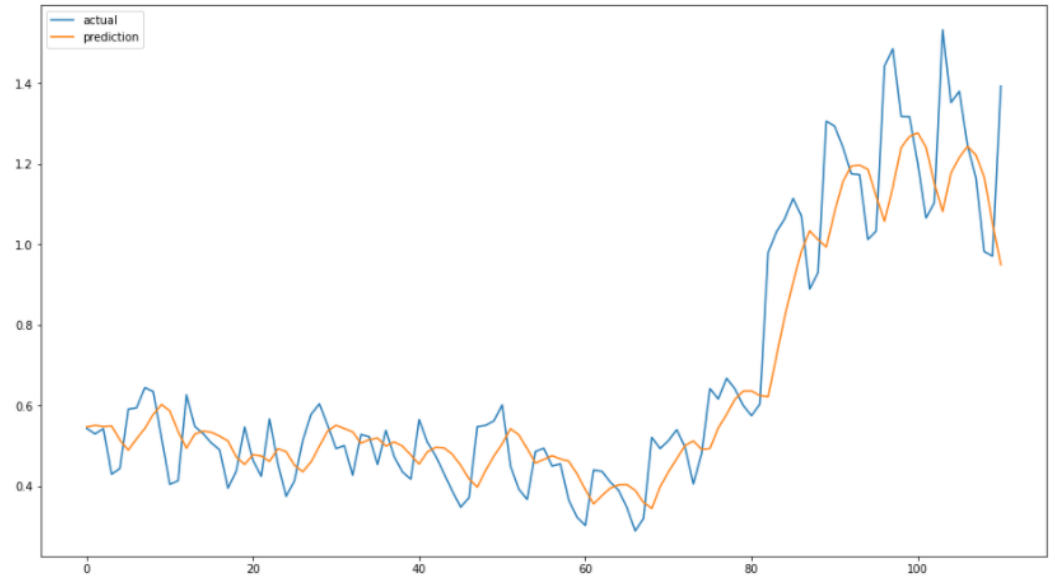
→ tanh 사용

rmse: 0.12877778837978235

여섯번째 시도

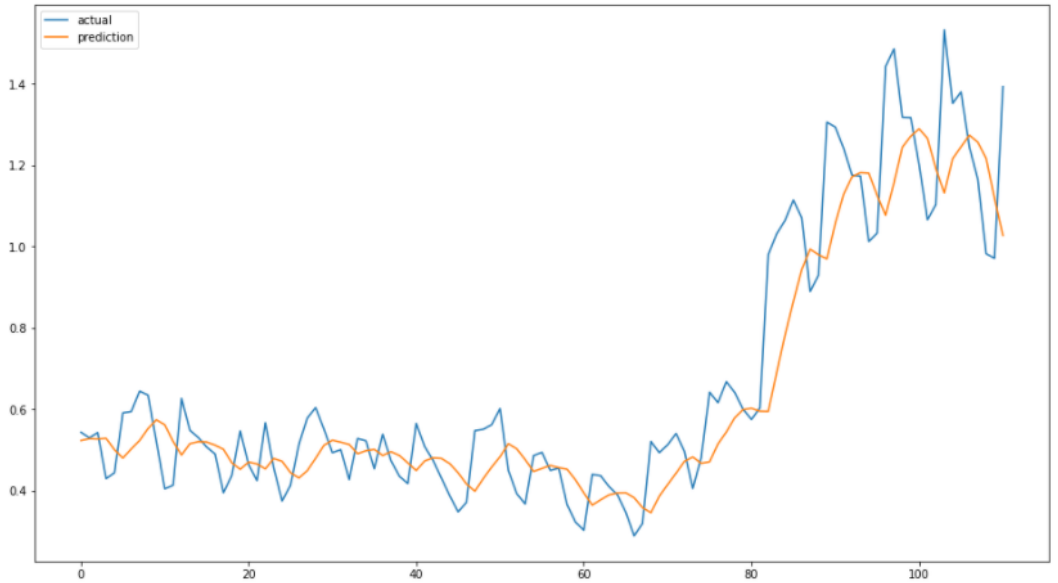
```
model = Sequential()
model.add(LSTM(32, input_shape=(seq_length,1), activation='relu'))
model.add(Dense(14))
model.add(Dense(1))
model.compile(optimizer='rmsprop',
              loss='mse',
              metrics=['acc'])

early_stop = EarlyStopping(monitor='loss',patience=1)

history = model.fit(X_train, y_train,
                    epochs=100,
                    batch_size=32,
                    callbacks=[early_stop])
```

첫번째 시도 그래프



→ adam 사용

rmse: 0.13084277528780466



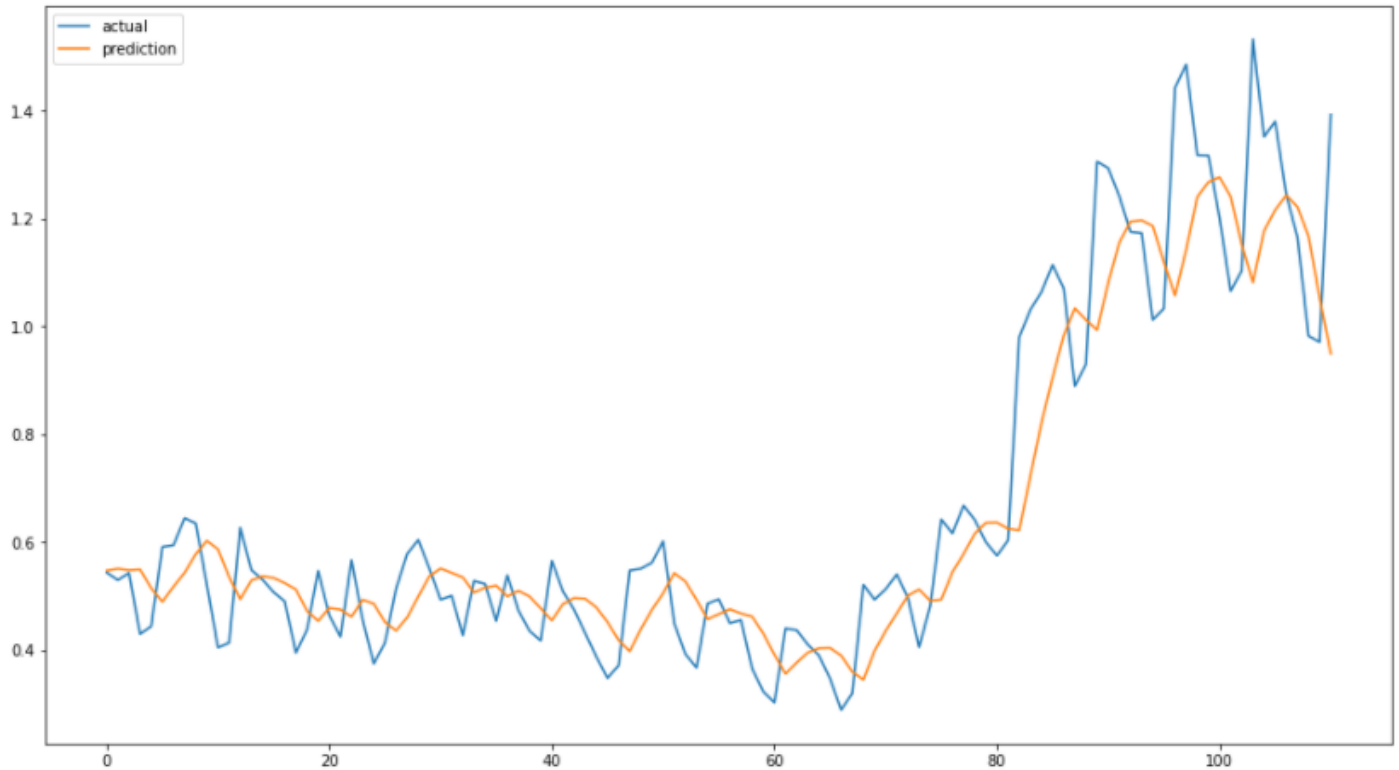
→ rmsprop 사용

rmse: 0.13103930911146722

최종

```
model = Sequential()
model.add(LSTM(32, input_shape=(seq_length,1), activation='relu'))
model.add(Dense(14))
model.add(Dense(1))

model.compile(optimizer='adam',
              loss='mse',
              metrics=['acc'])

early_stop = EarlyStopping(monitor='loss',patience=1)

history = model.fit(X_train, y_train,
                    epochs=100,
                    batch_size=32,
                    callbacks=[early_stop])
```

Train: 2020/1/22 ~ 2021/3/31
Test: 2021/4/1 ~ 2021/8/3



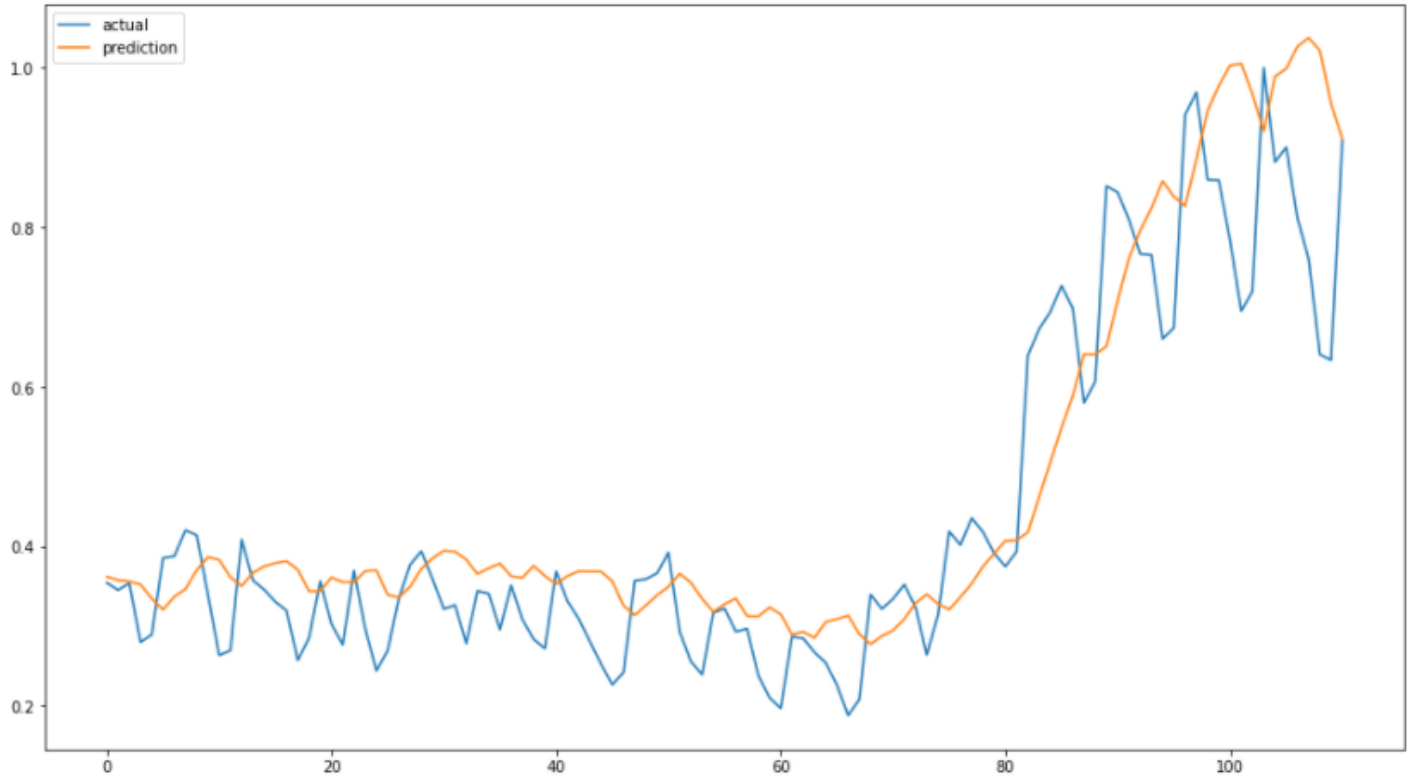
rmse: 0.13084277528780466

추가 시도

- 사망자 수, 회복자 수, 백신 접종자 수 데이터를 추가

```
daily_df = daily_corona_df.join(daily_death_df, how='left').fillna(0)
daily_df = daily_df.join(daily_recover_df, how='left').fillna(0)
daily_df = daily_df.join(daily_vaccine_df, how='left').fillna(0).astype('int')
```

날짜	확진	사망	회복	접종
2020-01-22	1	0	0	0
2020-01-23	0	0	0	0
2020-01-24	1	0	0	0
2020-01-25	0	0	0	0
2020-01-26	1	0	0	0
...	...	...	...	...
2021-07-30	1539	6	1420	59945
2021-07-31	1442	3	1497	13421
2021-08-01	1215	1	931	480
2021-08-02	1201	5	1304	36092
2021-08-03	1723	2	1214	106529



rmse: 0.10815475102688983