

## 테슬라 주가 데이터



### Tesla, Inc. (TSLA) Historic Data and Prices | Nasdaq

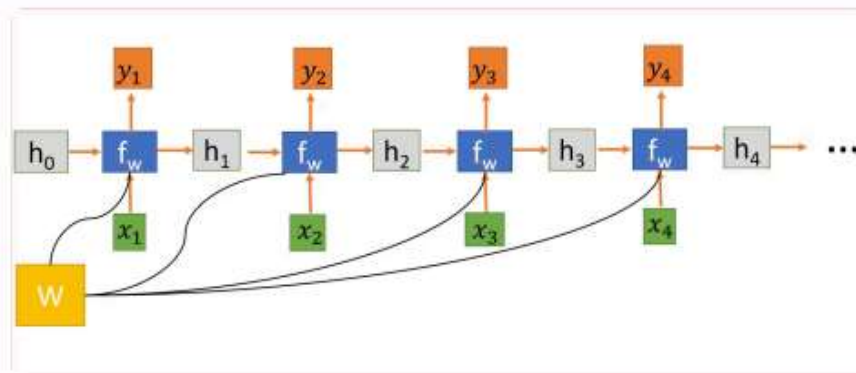
Find the latest historical data for Tesla, Inc. Common Stock (TSLA) at Nasdaq.com. View historical data in a monthly, bi-annual, or yearly format.

	A	B	C	D	E	F
1	Date	Close/Last	Volume	Open	High	Low
2	11/01/2024	\$248.98	57544760	\$252.04	\$254.00	\$246.63
3	10/31/2024	\$249.85	66575290	\$257.99	\$259.75	\$249.25
4	10/30/2024	\$257.55	53993580	\$258.04	\$263.35	\$255.82
5	10/29/2024	\$259.52	80521750	\$264.51	\$264.98	\$255.51
6	10/28/2024	\$262.51	107653600	\$270.00	\$273.54	\$262.24
7	10/25/2024	\$269.19	161611900	\$256.01	\$269.49	\$255.32
8	10/24/2024	\$260.48	204491900	\$244.68	\$262.12	\$242.65
9	10/23/2024	\$213.65	80938890	\$217.13	\$218.72	\$212.11
10	10/22/2024	\$217.97	43268740	\$217.31	\$218.22	\$215.26
11	10/21/2024	\$218.85	47328990	\$218.90	\$220.48	\$215.73

2014/11/03 ~ 2024/11/01 데이터

## 순환 신경망 (RNN, LSTM)

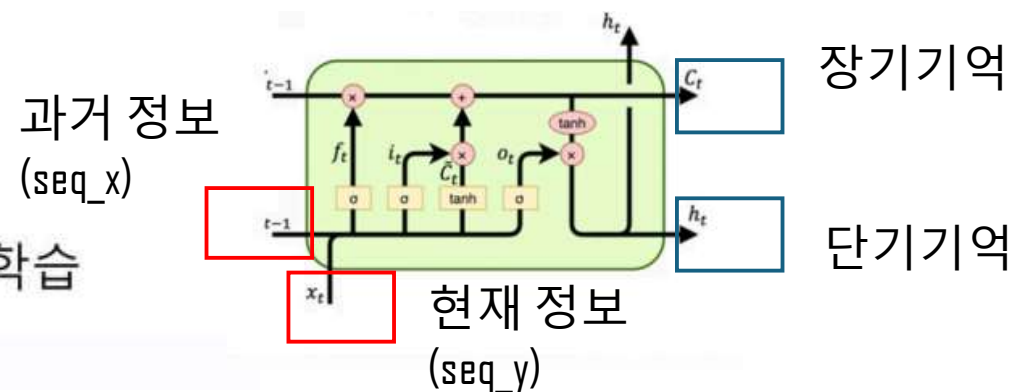
- (1) LSTM(Long Short-Term Memory)와 RNN(Recurrent Neural Network)은 순차적인 데이터(시퀀스)를 처리하는 데 특화된 신경망.
- (2) 일반 신경망과 달리 RNN/LSTM은 이전 입력을 기억하고 현재 입력과 결합하여 새로운 출력을 생성하는 메커니즘.
- (3) 자연어 처리, 음성 인식, **주가 예측**, IoT센서 데이터 분석 등 순차적인 데이터에 응용.



## 1. 데이터 불러오기

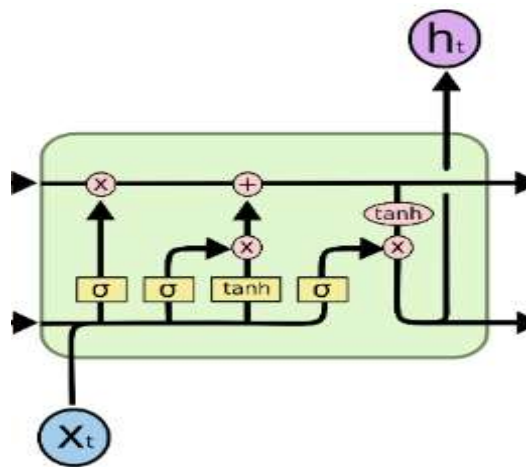
```
18 directory = r'C:\Users\dahae\machine learning\ch8_code'
19 dataset = pd.read_csv(directory + "\\\" + "HistoricalData_tesla.csv", index_col="Date", parse_dates=["Date"])
20
21
22 dataset = dataset.sort_index(ascending=True)#인덱스를 오름차순으로 정렬하여 시간 순으로 데이터가 배열
23 dataset['Close/Last'] = dataset['Close/Last'].replace(['\$',], '', regex=True).astype(float)#$와 , 제거
24 dataset['Open'] = dataset['Open'].replace(['\$',], '', regex=True).astype(float)
25 dataset['High'] = dataset['High'].replace(['\$',], '', regex=True).astype(float)
26 dataset['Low'] = dataset['Low'].replace(['\$',], '', regex=True).astype(float)
27 print(dataset)
28
29 tstart = 2017
30 tend = 2023
31 dataset.loc[f"{tstart}":f"{tend}", "High"].plot(figsize=(16, 4),c='blue', legend=False)#2017-2023 High(주가 고점)데이터를 가지고 와서 파랑색 표시
32 dataset.loc[f"{tend+1}":, "High"].plot(figsize=(16, 4), c='black',legend=False)#2024 High(주가 고점)데이터를 가지고 와서 검정색 표시
33 plt.title("Tesla stock price from 2017 to 2024")
34 plt.show()
35
36 x_train = dataset.loc[f"{tstart}":f"{tend}", "High"].values #Train data로 2023 - 2024 data
37 original_test = dataset.loc[f"{tend+1}":, "High"].values # Test data로 2024년 이후 data
38 scaler = MinMaxScaler(feature_range=(0, 1))#MinMaxScaler 정규화 설정
39 x_train_norm = x_train.reshape(-1, 1)#2차원 배열로 변환하여 스케일러에 적합한 형식으로 만들
40 x_train_scaled = scaler.fit_transform(x_train_norm)#MinMaxScaler 정규화 적용(0~1로 데이터 변환)
```

## 2. 학습 데이터 추출 코드



```
42 def split_sequence(sequence, n_steps):
43     X, y = list(), list()
44     for i in range(len(sequence)):
45         end_ix = i + n_steps # 61
46         if end_ix > len(sequence) - 1:
47             break
48         seq_x, seq_y = sequence[i:end_ix], sequence[end_ix]#seq_x 과거의 data(1~60) , seq_y(61): 예측 할 값
49         X.append(seq_x)
50         y.append(seq_y)
51     return np.array(X), np.array(y)
52
53 timesteps = 60 # 과거 60일간 주가
54 features = 1 # 주가변동 데이터-특징벡터 1개
55 X_train, y_train = split_sequence(x_train_scaled, timesteps)
56
```

### 3. LSTM 모델 구축



```
57 model_lstm = Sequential()#순차적으로 레이어를 쌓아 올리는 모델 구축 방법
58 model_lstm.add(InputLayer(input_shape=(timesteps,features)))#input_shape(timesteps, features):
59 model_lstm.add(LSTM(units=125,activation="tanh",return_sequences=True))#return_sequences=True를 통해 모든 타임 스텝에서 출력값이 다음 LSTM 모델로 넘어감
60 model_lstm.add(LSTM(units=64,activation="tanh"))#return_sequences 옵션이 없으므로 마지막 스텝에서 출력값만 반환
61 model_lstm.add(Dense(units=1))#Dense 는 linear
62
63 model_lstm.compile(optimizer="RMSprop", loss="mse")
64 model_lstm.summary()
65 model_lstm.fit(X_train, y_train, epochs=50, batch_size=32)
66
```

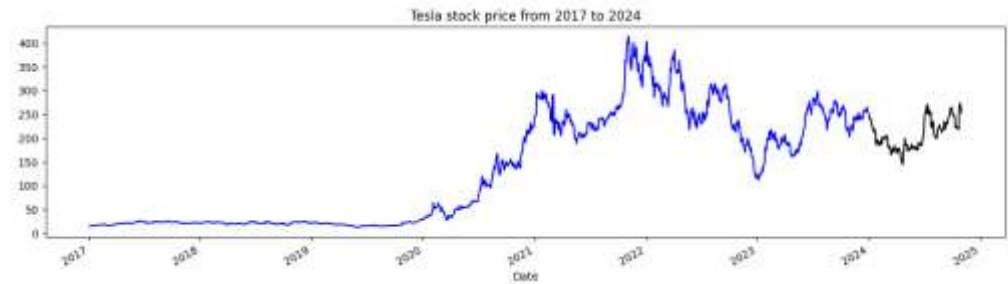
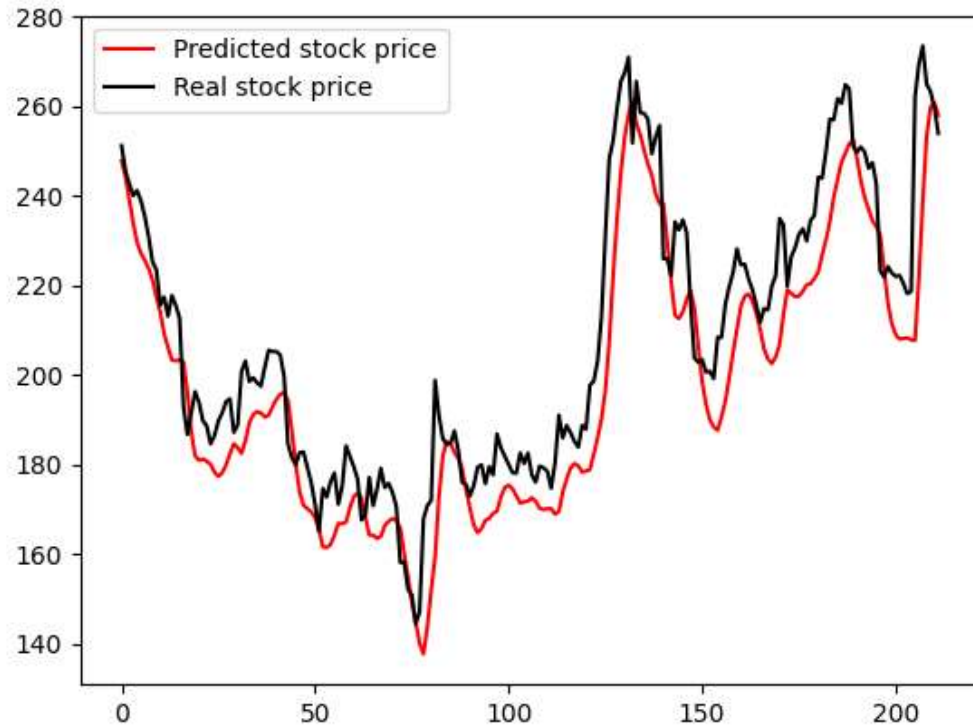
## 4. LSTM 모델 예측 코드

```
# # 예측
dataset_total = dataset.loc[:, "High"]
x_test = dataset_total[len(dataset_total) - len(original_test) - timesteps:].values() #테스트 데이터 예측을 위해 필요한 과거 60일치 데이터와 테스트 데이터를 함께 가져오는 코드
x_test_norm = x_test.reshape(-1, 1)
x_test_scaled = scaler.transform(x_test_norm)

X_test, y_test = split_sequence(x_test_scaled, timesteps)
predicted_stock_price = model_lstm.predict(X_test) #학습된 lstm 모델을 사용하여 주가 예측
predicted_stock_price = scaler.inverse_transform(predicted_stock_price) #예측된 주가 데이터를 원래 스케일로 복원

plt.figure()
plt.plot(predicted_stock_price, c='red', label='Predicted stock price')
plt.plot(original_test, c='black', label='Real stock price')
plt.legend()
plt.show()
```

## 5. LSTM 모델 결과



## 6. 다른 데이터에 적용해보기

META / Historical Quotes

Meta Platforms, Inc. Class A Common Stock (META)

Historical Quotes

1M 6M YTD 1Y 5Y MAX Download historical data

Date	Close/Last	Volume	Open	High	Low
11/05/2024	\$572.43	9,775,379	\$567.305	\$573.48	\$566.67
11/04/2024	\$560.68	12,064,600	\$564.10	\$568.69	\$557.89
11/01/2024	\$567.16	15,303,240	\$567.605	\$573.67	\$562.56
10/31/2024	\$567.58	26,838,400	\$585.00	\$589.90	\$563.01
10/30/2024	\$591.80	26,864,930	\$600.98	\$601.20	\$589.38
10/29/2024	\$593.28	13,019,090	\$580.145	\$593.67	\$575.398
10/28/2024	\$578.16	10,925,130	\$582.00	\$583.7499	\$574.12
10/25/2024	\$573.25	11,337,870	\$573.935	\$581.29	\$571.72
10/24/2024	\$567.78	7,184,651	\$567.125	\$568.8799	\$561.52
10/23/2024	\$563.69	14,248,420	\$579.97	\$585.00	\$562.50

Show: 10 < 1 2 3 ... 252 > Showing 1 - 10 of 2515

[https://www.nasdaq.com/market-activity/stocks/meta/historical?page=1&rows\\_per\\_](https://www.nasdaq.com/market-activity/stocks/meta/historical?page=1&rows_per_)

<https://www.nasdaq.com/market-activity/stocks/nvda/historical>