# Stock Market Trends with machine learning!

## Utilize the RNN and LSTM

Daeyeop Kim - Date -

# Agenda

## What is our goal?

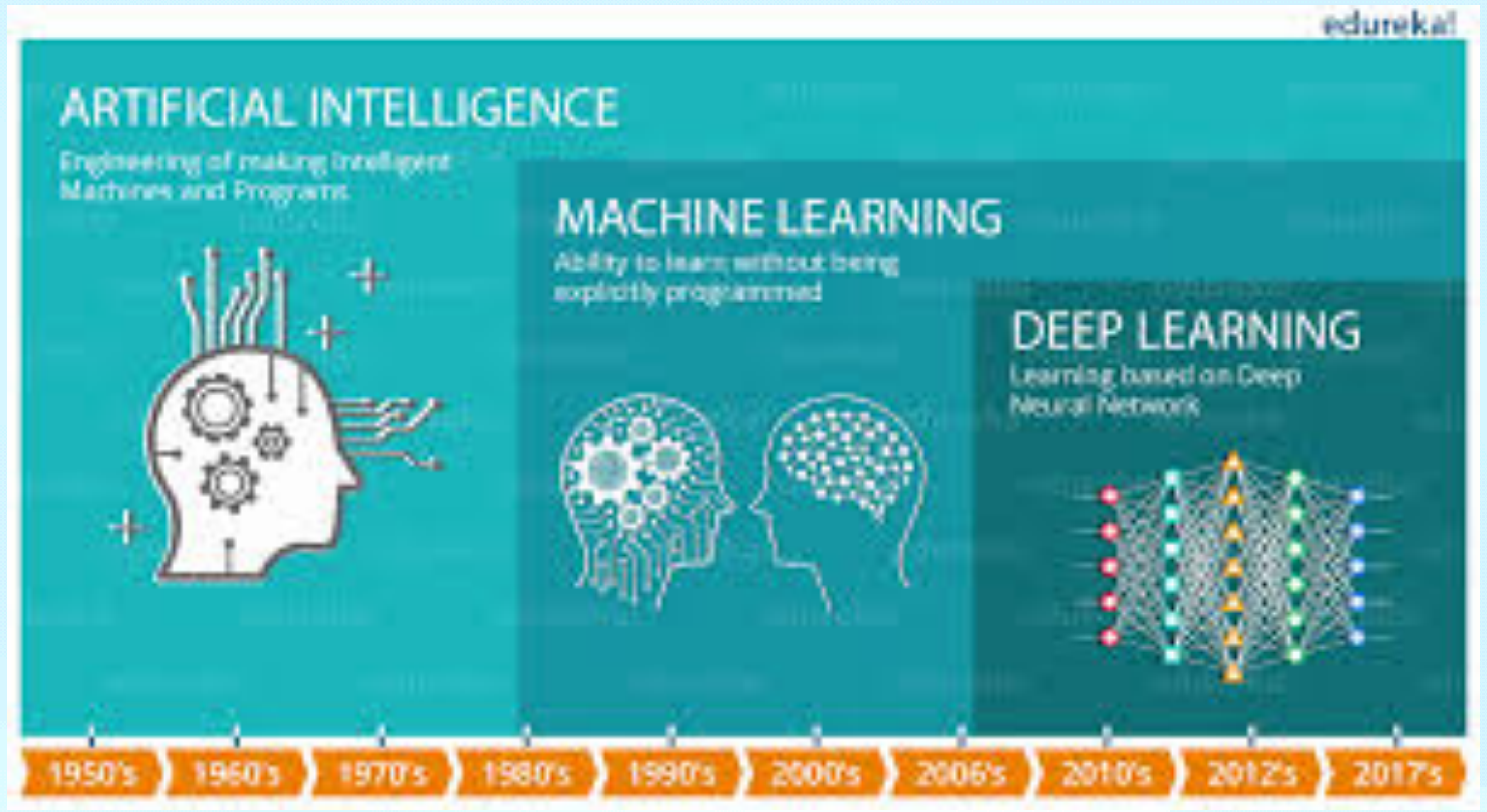
- What is Deep Learning?
- What is the RNN?
- What is LSTM?
- Where We get the Data?
- Data Preprocessing
- Model and scaling
- Model result
- Reference

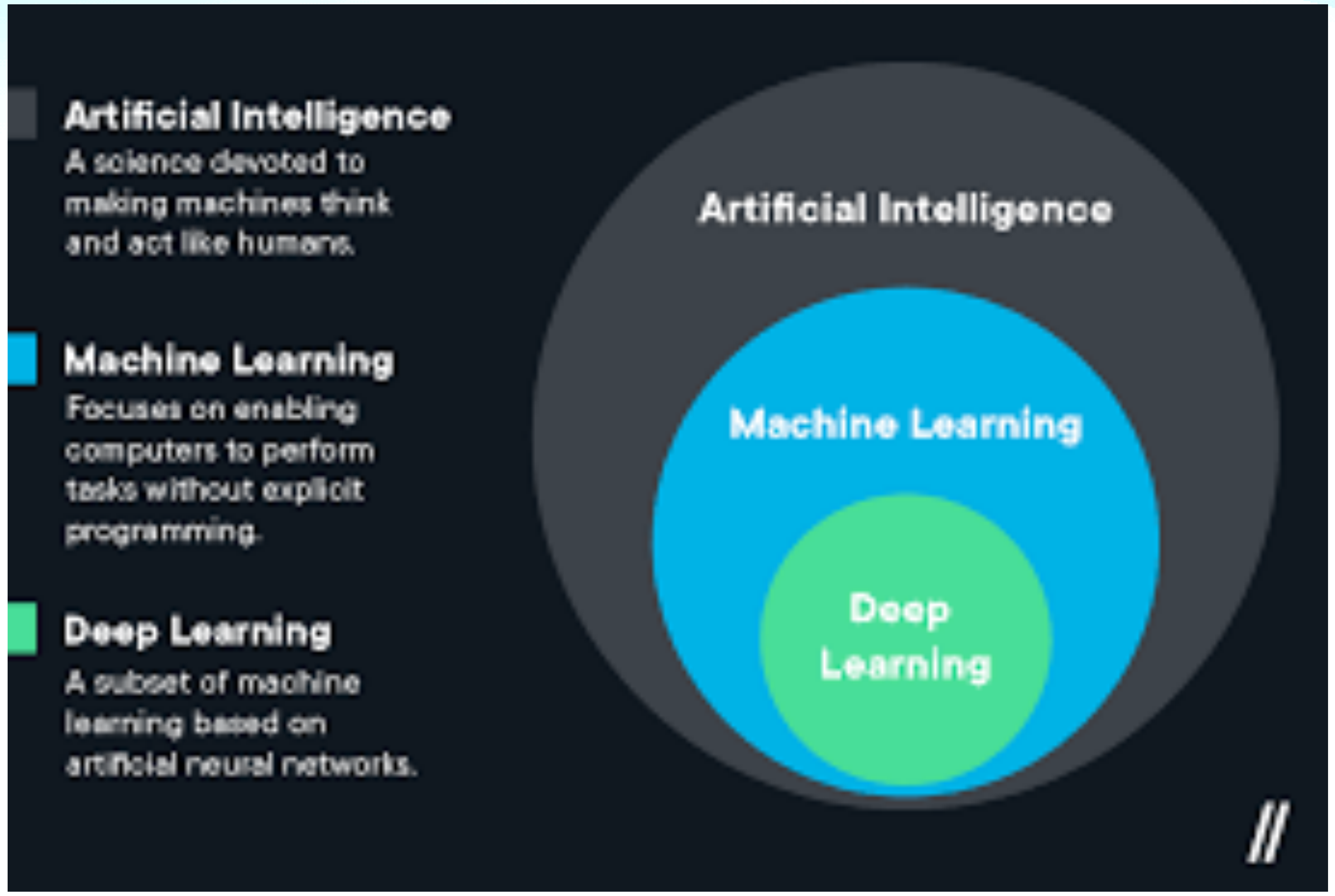
# What is Deep-learning?

## deep - learning

- youtube link for what is the Deep learning
  - https://www.youtube.com/watch?v=WSbgixdC9g8
- Deep learning is a specific subfield of machine learning: a new take on learning representations from data that puts an emphasis on learning successive layers of increasingly meaningful representations.
- In deep learning, these layered representations are learned via models called *neu-ral networks*, structured in literal layers stacked on top of each other.

.


edureka!
ARTIFICIAL INTELLIGENCE
Engineering of making Intelligent Machines and Programs
MACHINE LEARNING
Ability to learn without being explicitly programmed
DEEP LEARNING
Learning based on Deep Neural Network
1950's
1960's
1970's
1980's
1990's
2000's
2006's
2010's
2012's
2017's

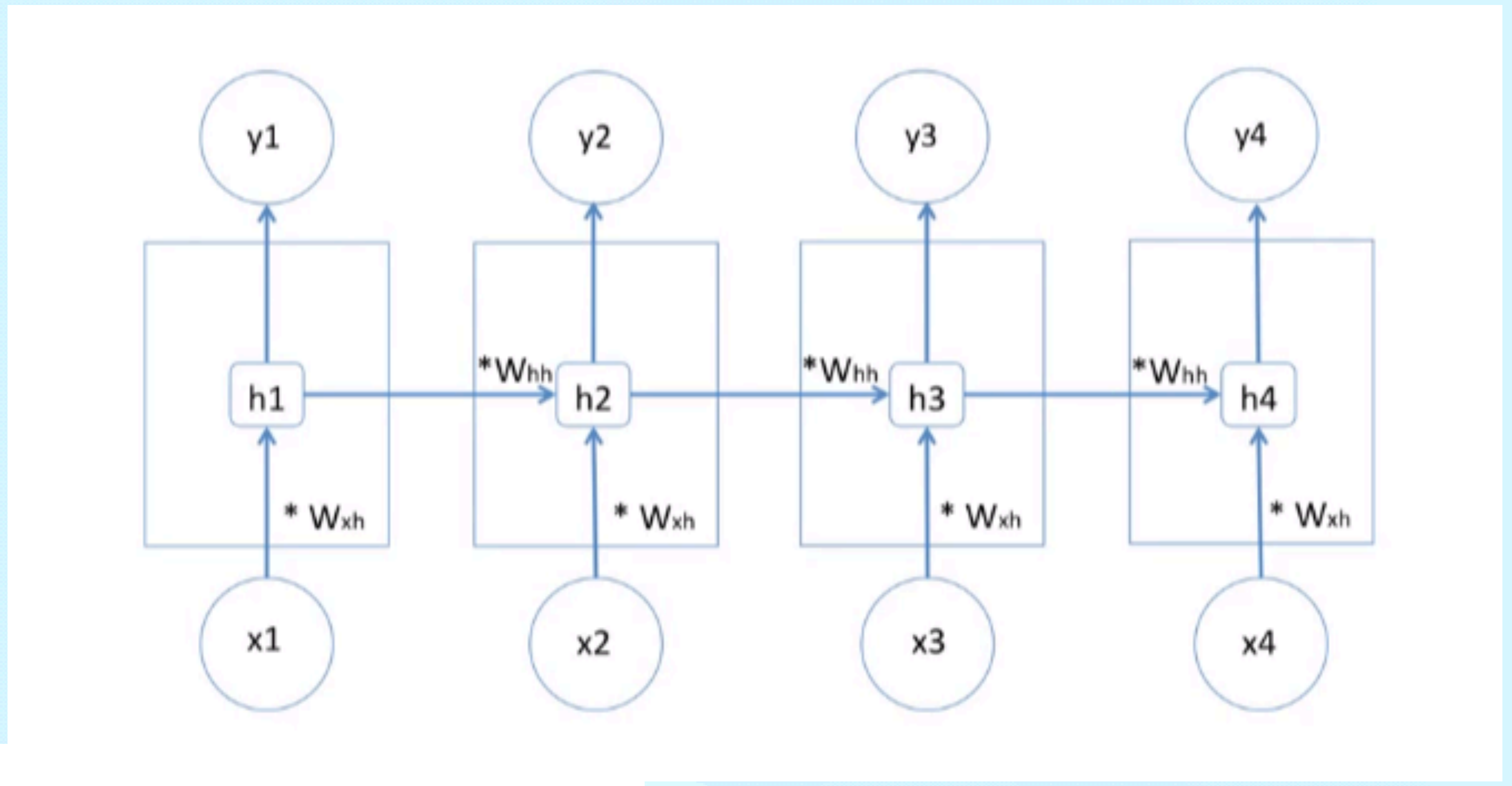

Artificial Intelligence
A science devoted to making machines think and act like humans.
Machine Learning
Focuses on enabling computers to perform tasks without explicit programming.
Deep Learning
A subset of machine learning based on artificial neural networks.
Artificial Intelligence
Machine Learning
Deep Learning

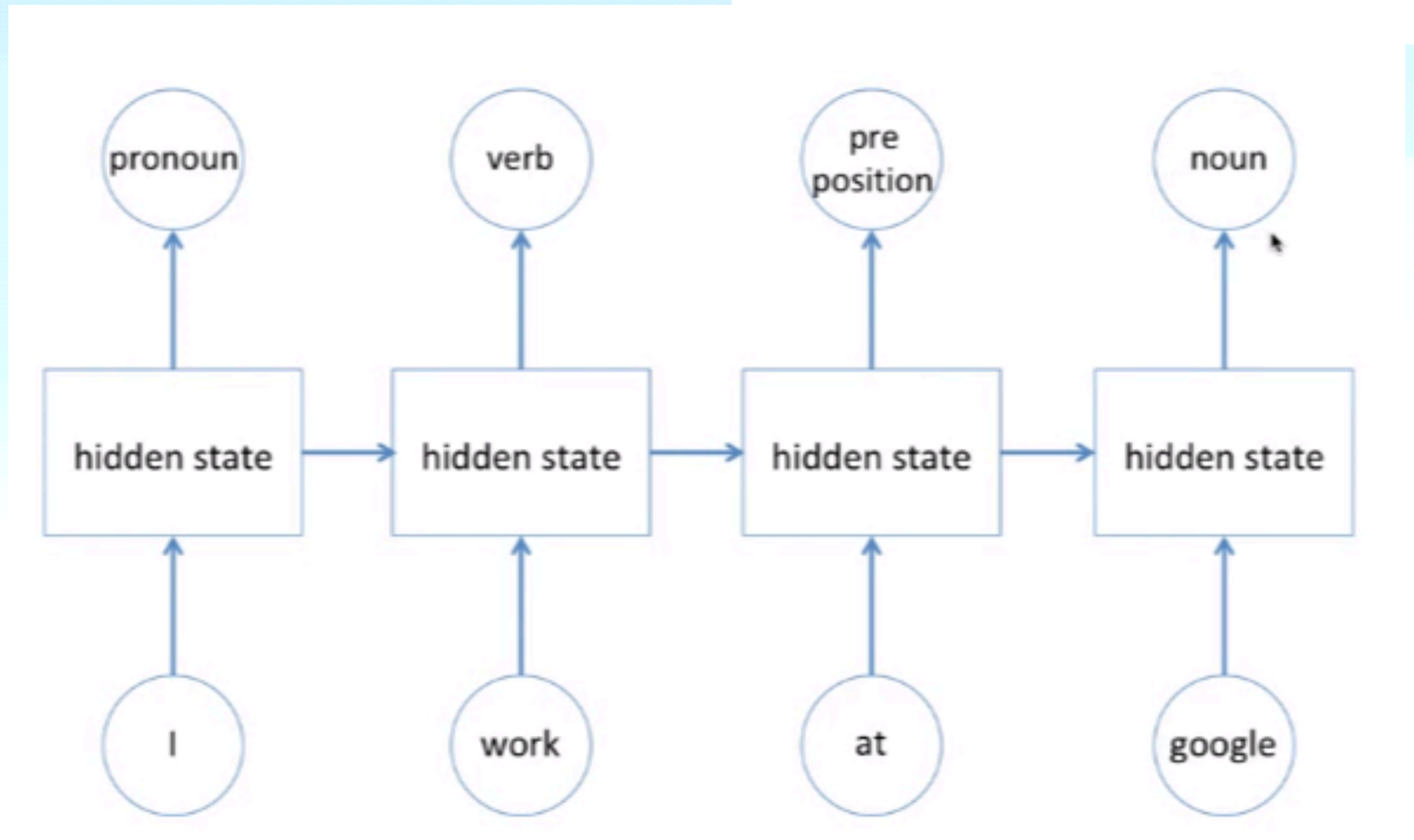
# What is RNN?

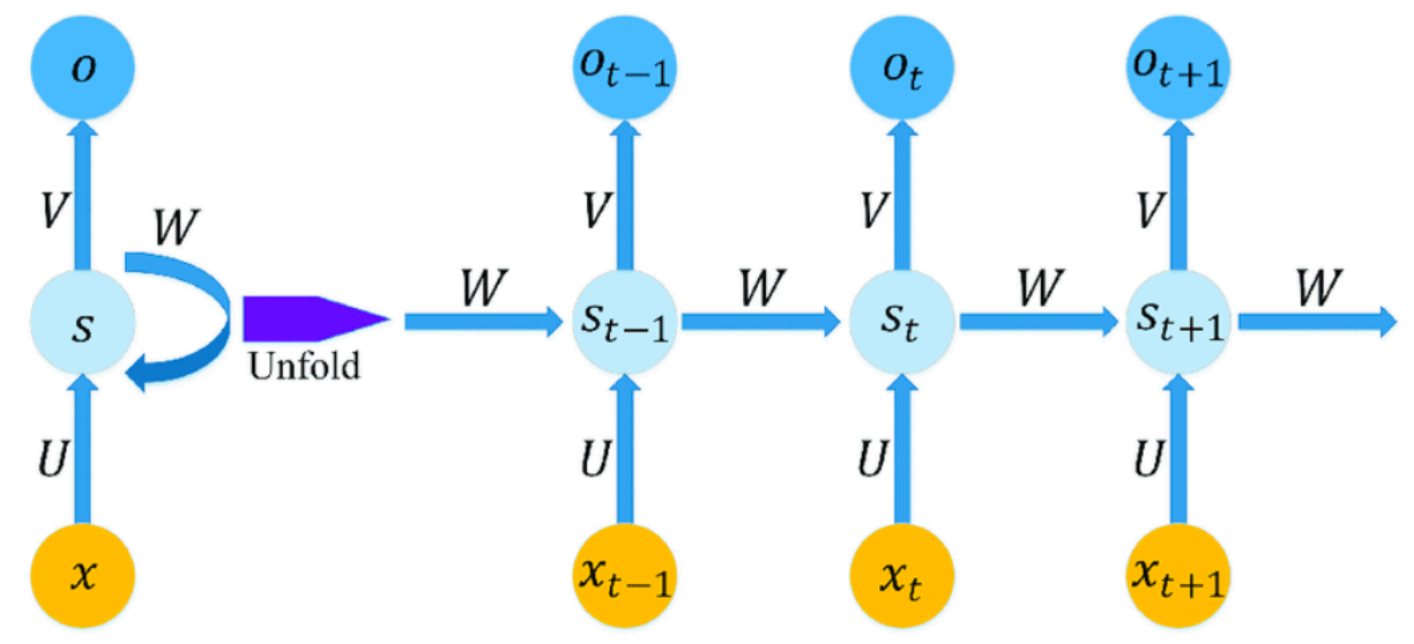
## Recurrent Neural Network

- A recurrent neural network, a type of neural network is used to process data that is sequential in the nature.
- The recurrent neural networks, at each timestep, the previous output of the neuron is also multiplied by the current input via a weight vector.
- Problem with the RNN, it can capture a shorter sequence, and it tends to forget longer sequences.

.



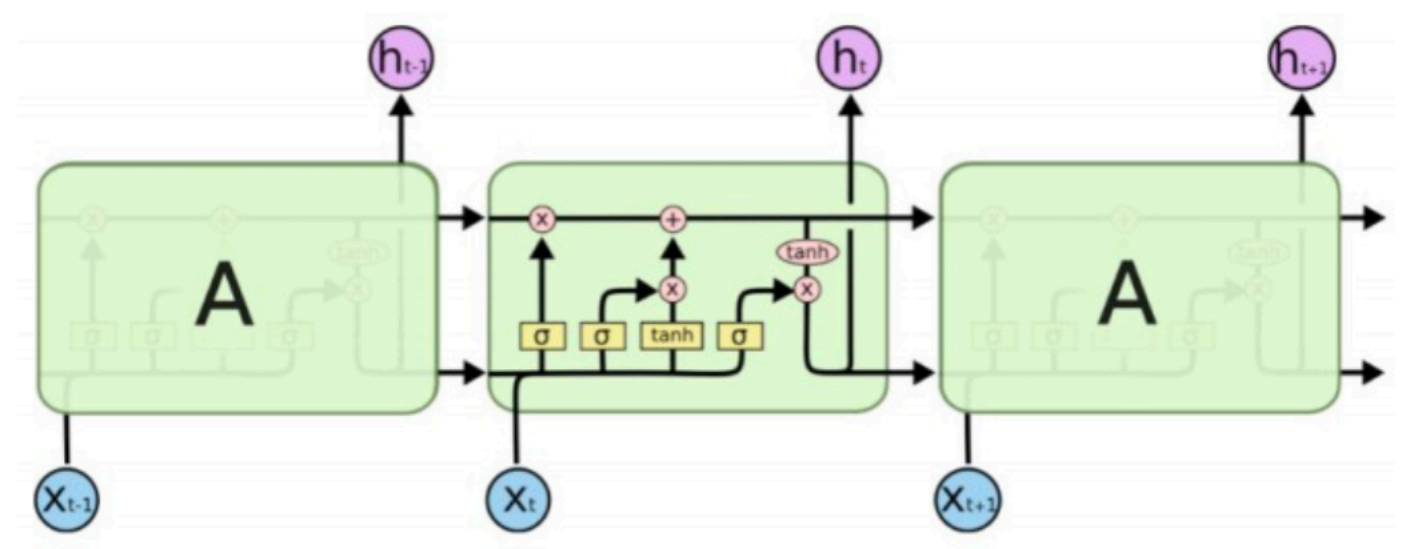




# What is LSTM?

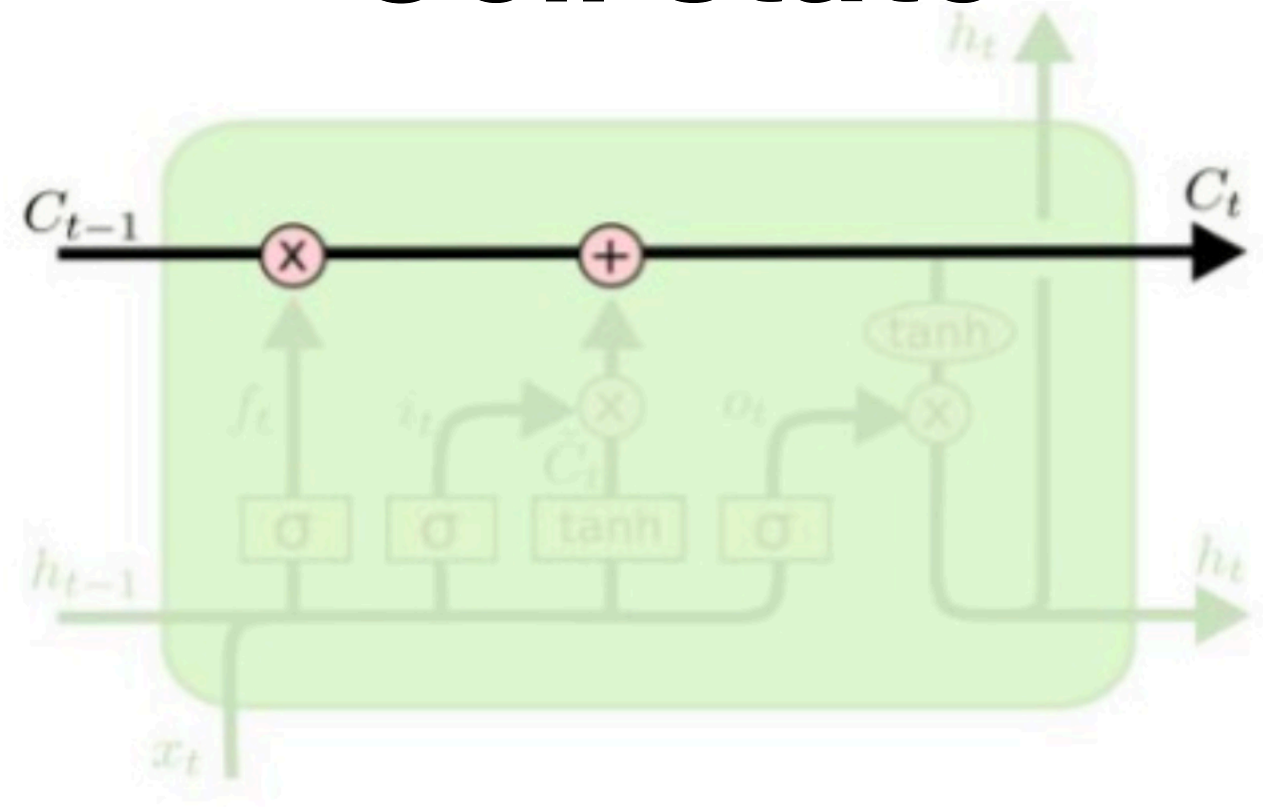
**Long Short-Term Memory models**

- LSTM is the one of RNN model that capable of remembering longer sequences.
- The Cell state in LSTM is responsible for remembering a long sequence.
- LSTM is capable of adding or removing info to a cell state.
- The cell state is a part of the previous info to remember and which info to forget.



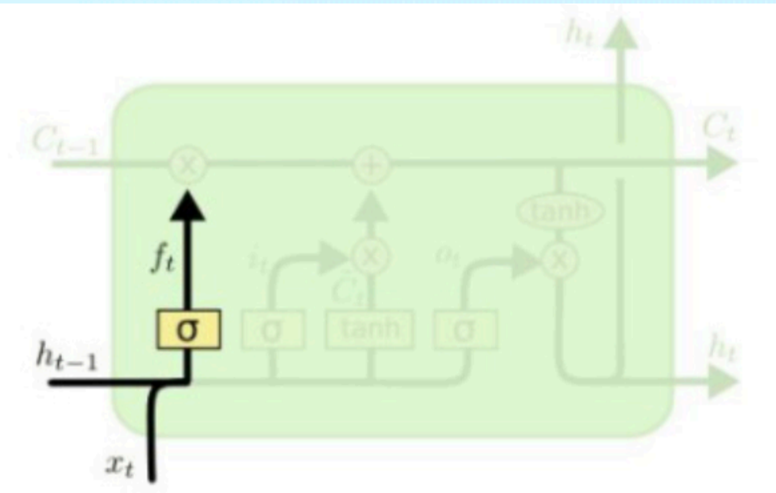
LSTM: Single Input - Single Output

**Cell state**



# Process of LSTM

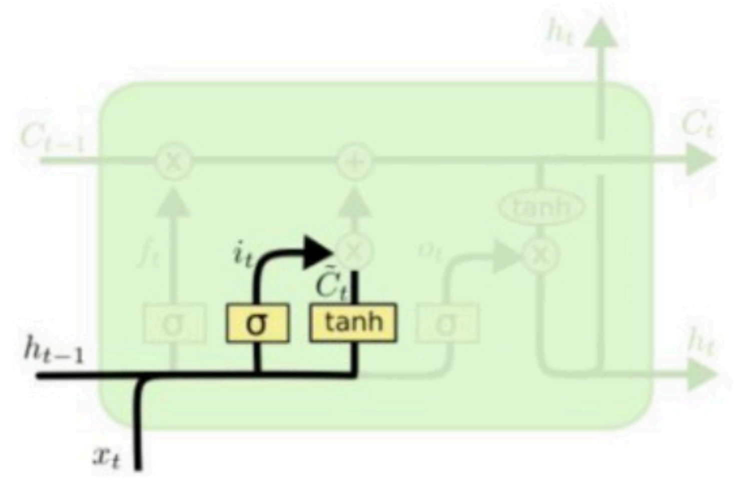
**1. Forget Gate**



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

- The forget gate is used to decide which information to remember or forget.

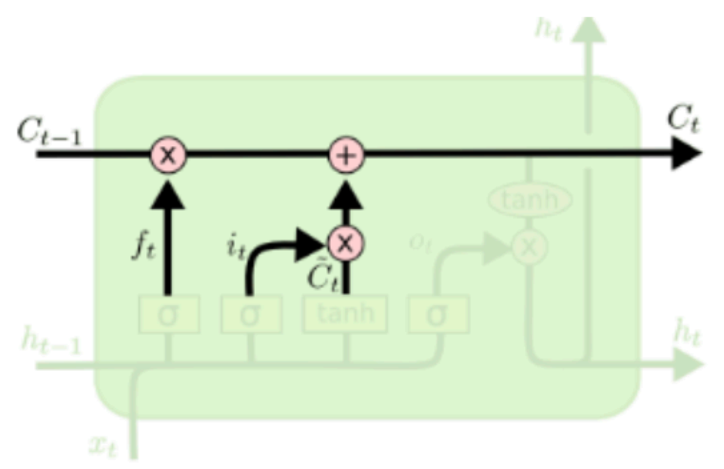
**2. Input Gate**



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

- The input gate is responsible for updating or adding any new information.

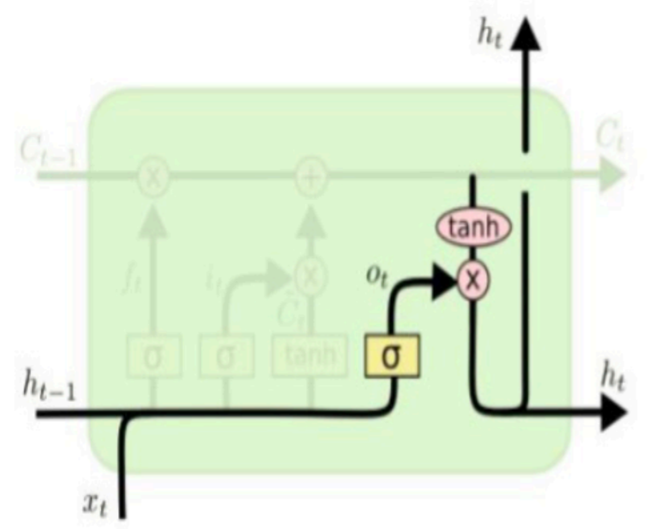
**3. Update Gate**



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

- The update gate is used to preform forget and input operations.

**4. Output Gate**



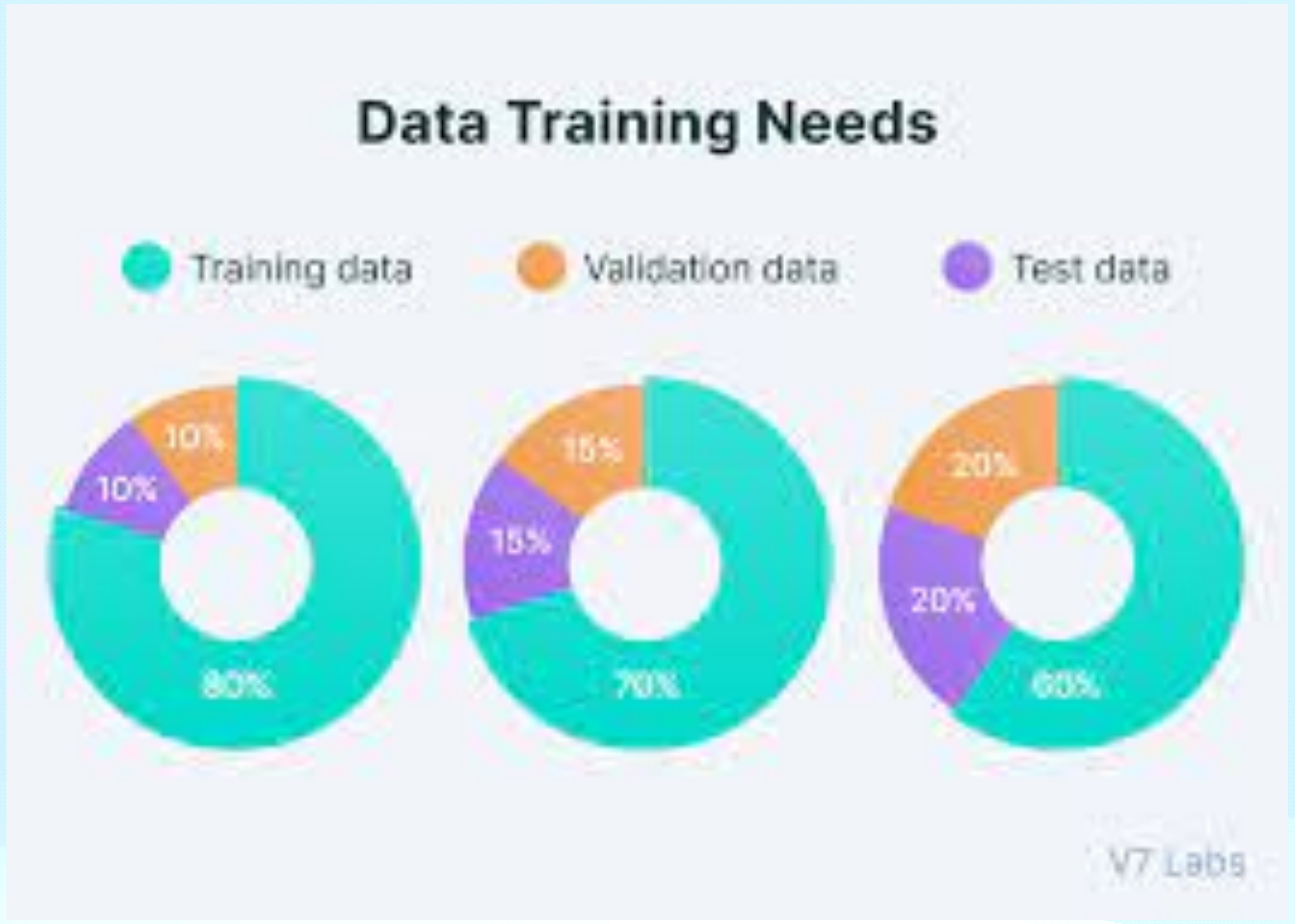
$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

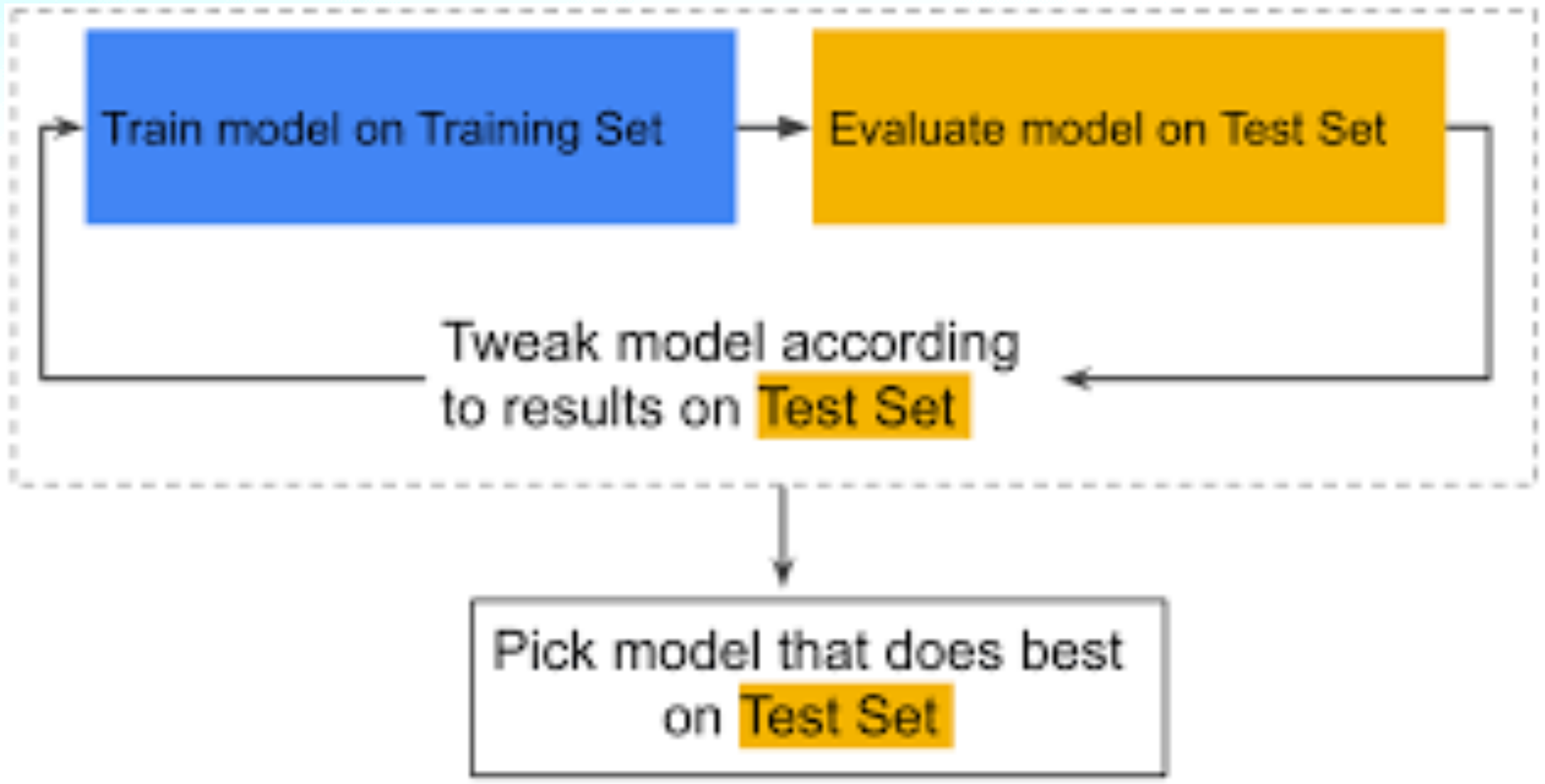
- The output gate outputs the hidden state and the output just like a common recurrenet neural network.

# What data we need ?

## Where we can get the data

- For this project, we need the two different dataset for training and test.
- For the training dataset, we need the long term of the historical dataset.
- For the testing dataset, we need the short term of the historical dataset.

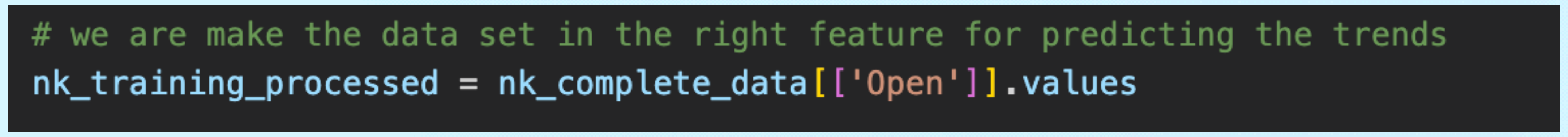



Train model on Training Set
Evaluate model on Test Set
Tweak model according to results on Test Set
Pick model that does best on Test Set

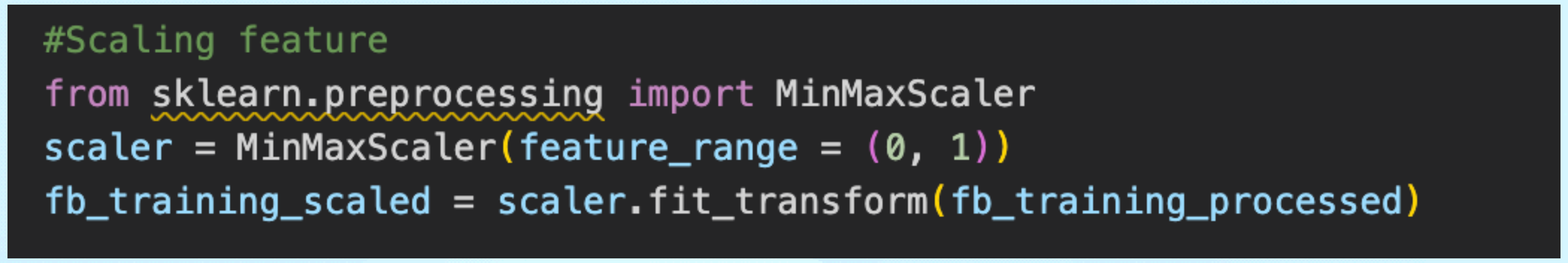
# Data processing

## Featuring the data

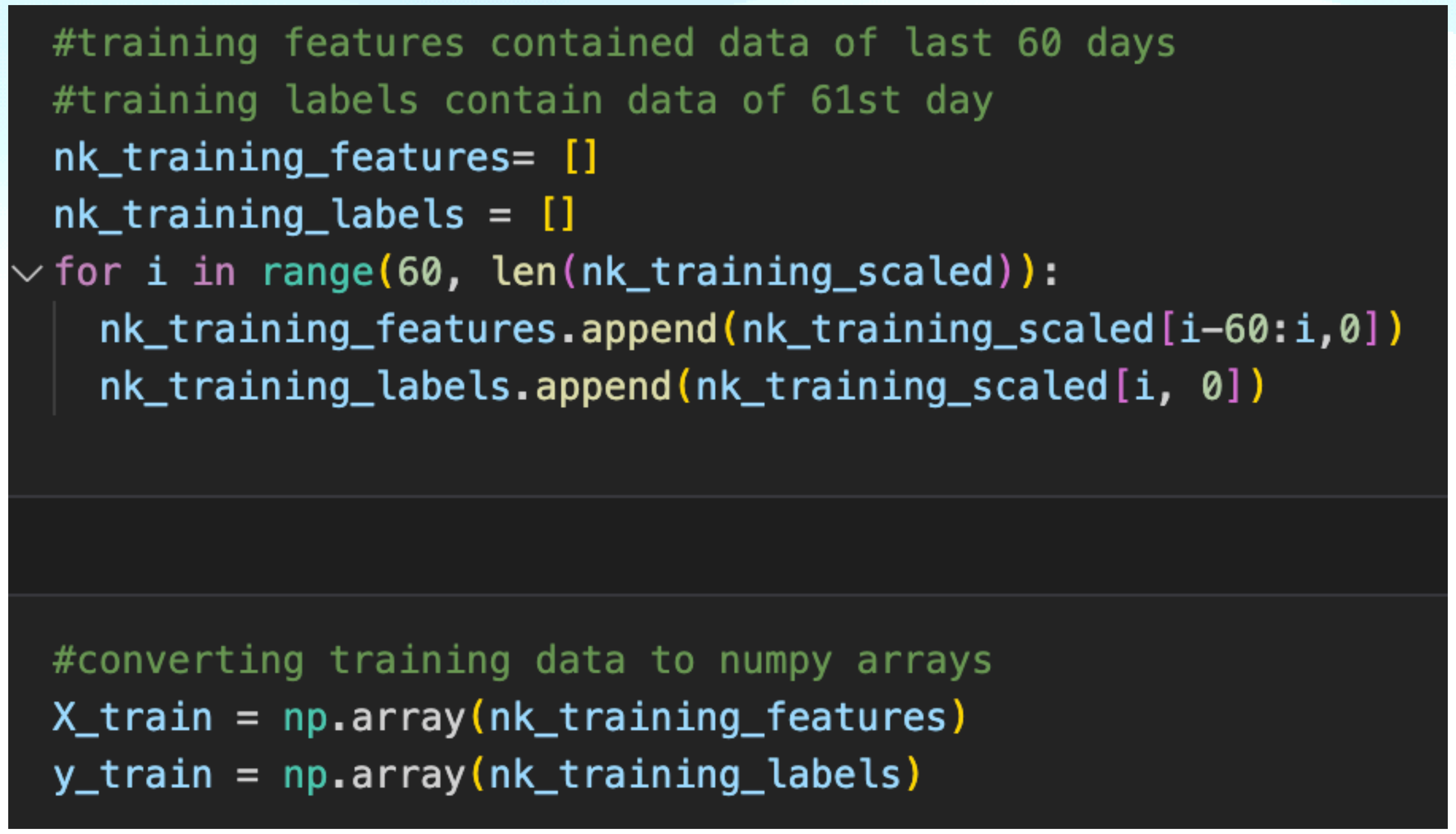
- Let's first import the data and then remove all the columns from the test data except the Open column.
- Let's concatenate the training and test sets. We do this because to predict the first value in the test set, the input will be the data from the past 60 days, which is basically the data from the last 60 days in the training set.
- Since our feature set are given in 2-dimensional, need to reshape in 3 dimensional because the LSTM algorithm in Keras accepts data in 3-dimensional.



```
# we are make the data set in the right feature for predicting the trends
nk_training_processed = nk_complete_data[['Open']].values
```

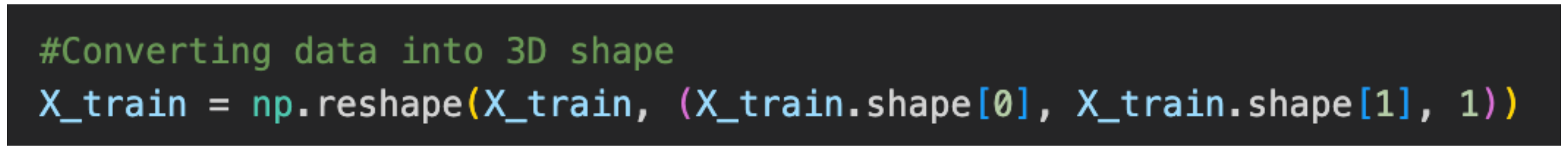


```
#Scaling feature
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (0, 1))
fb_training_scaled = scaler.fit_transform(fb_training_processed)
```



```
#training features contained data of last 60 days
#training labels contain data of 61st day
nk_training_features= []
nk_training_labels = []
for i in range(60, len(nk_training_scaled)):
  nk_training_features.append(nk_training_scaled[i-60:i,0])
  nk_training_labels.append(nk_training_scaled[i, 0])
```

```
#converting training data to numpy arrays
X_train = np.array(nk_training_features)
y_train = np.array(nk_training_labels)
```

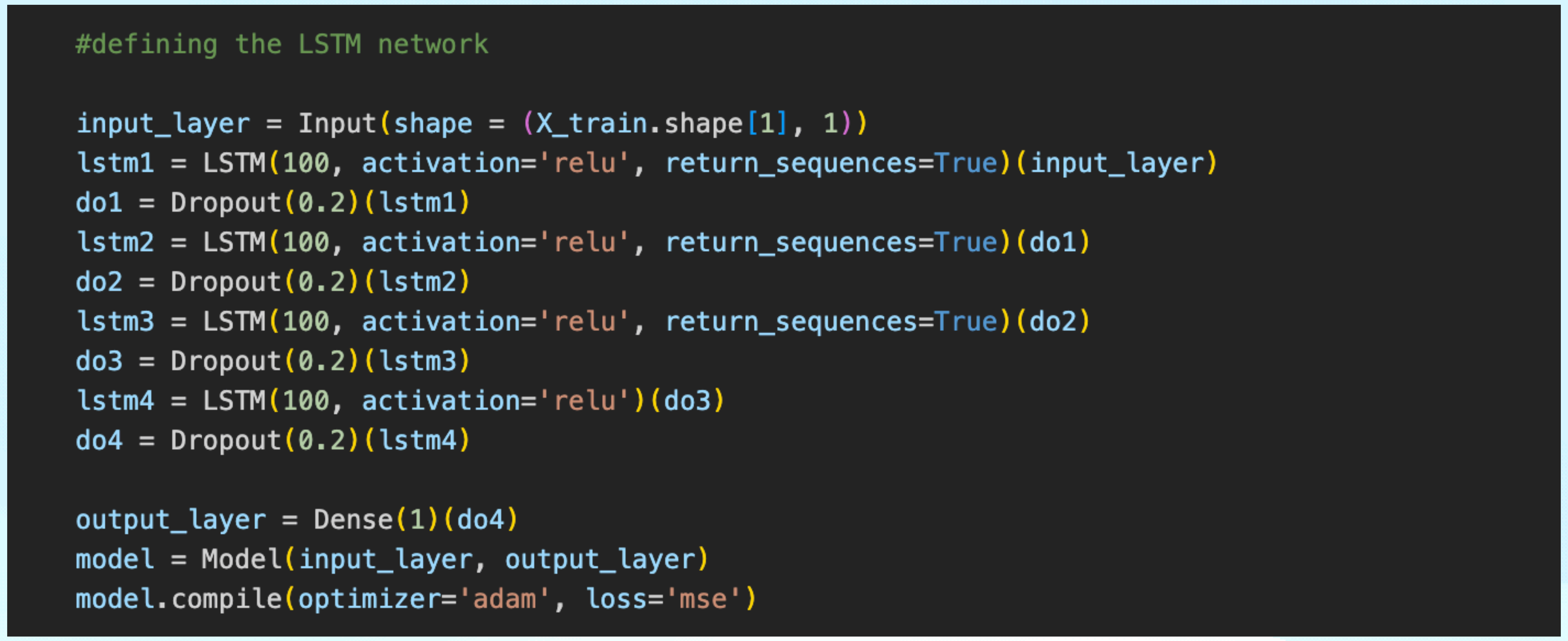


```
#Converting data into 3D shape
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))
```

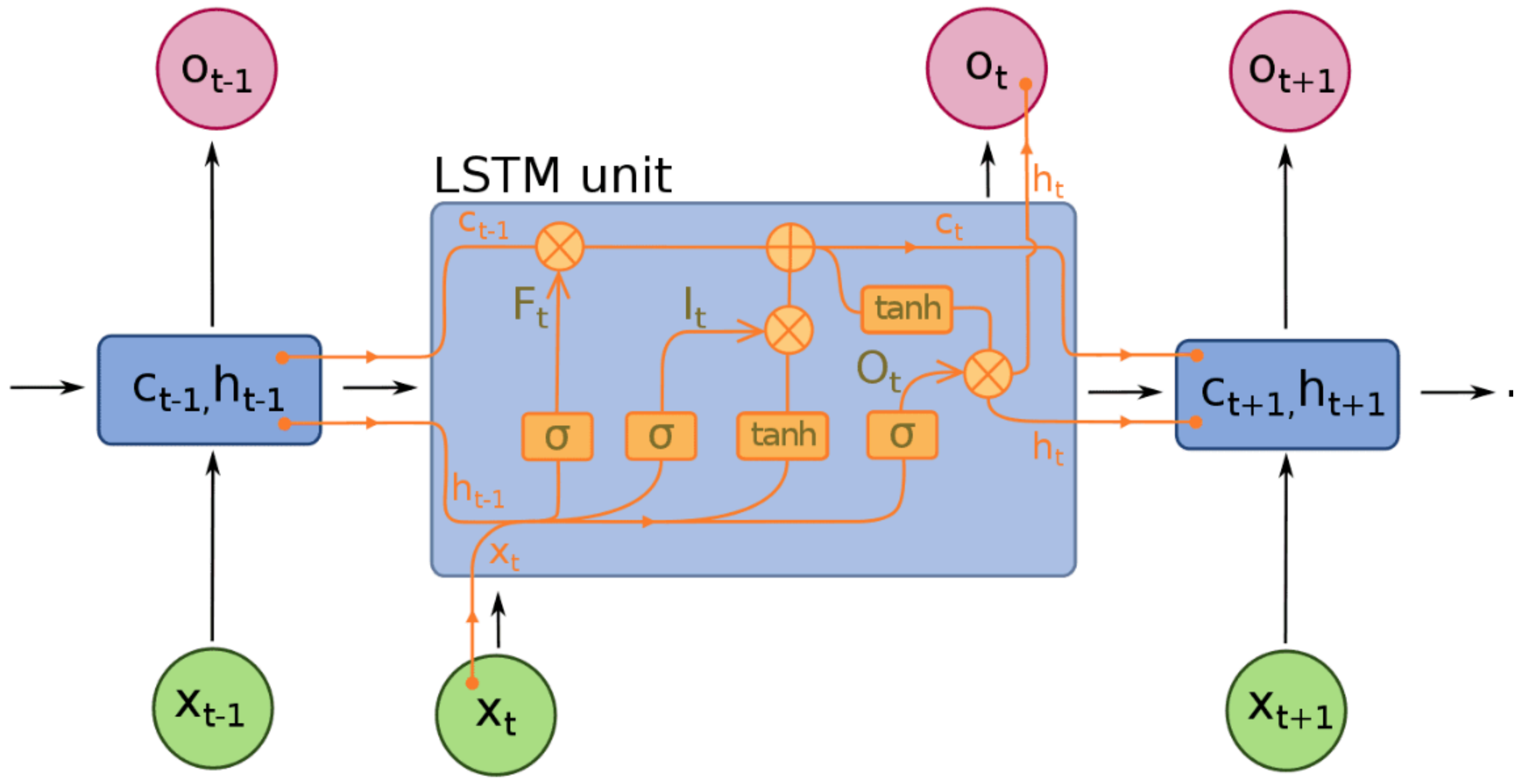
# Model and Scaling

## Use the LSTM model

- The following script creates our LSTM model. We have 4 LSTM layers with 100 nodes each.
- Each LSTM layer is followed by a dropout layer to avoid overfitting. The final dense layer has one node since the output is a single value.



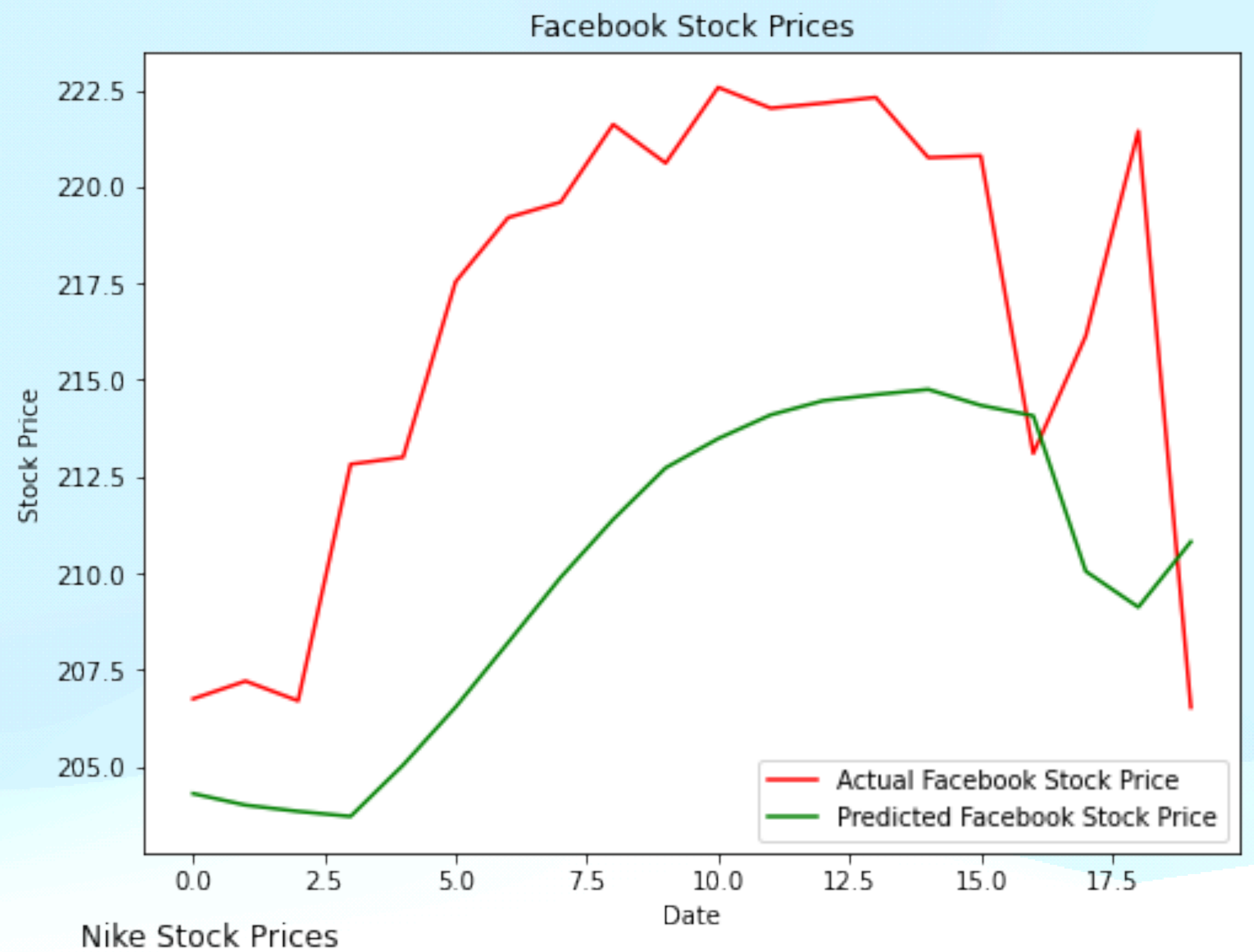
```
#defining the LSTM network

input_layer = Input(shape = (X_train.shape[1], 1))
lstm1 = LSTM(100, activation='relu', return_sequences=True)(input_layer)
do1 = Dropout(0.2)(lstm1)
lstm2 = LSTM(100, activation='relu', return_sequences=True)(do1)
do2 = Dropout(0.2)(lstm2)
lstm3 = LSTM(100, activation='relu', return_sequences=True)(do2)
do3 = Dropout(0.2)(lstm3)
lstm4 = LSTM(100, activation='relu')(do3)
do4 = Dropout(0.2)(lstm4)

output_layer = Dense(1)(do4)
model = Model(input_layer, output_layer)
model.compile(optimizer='adam', loss='mse')
```

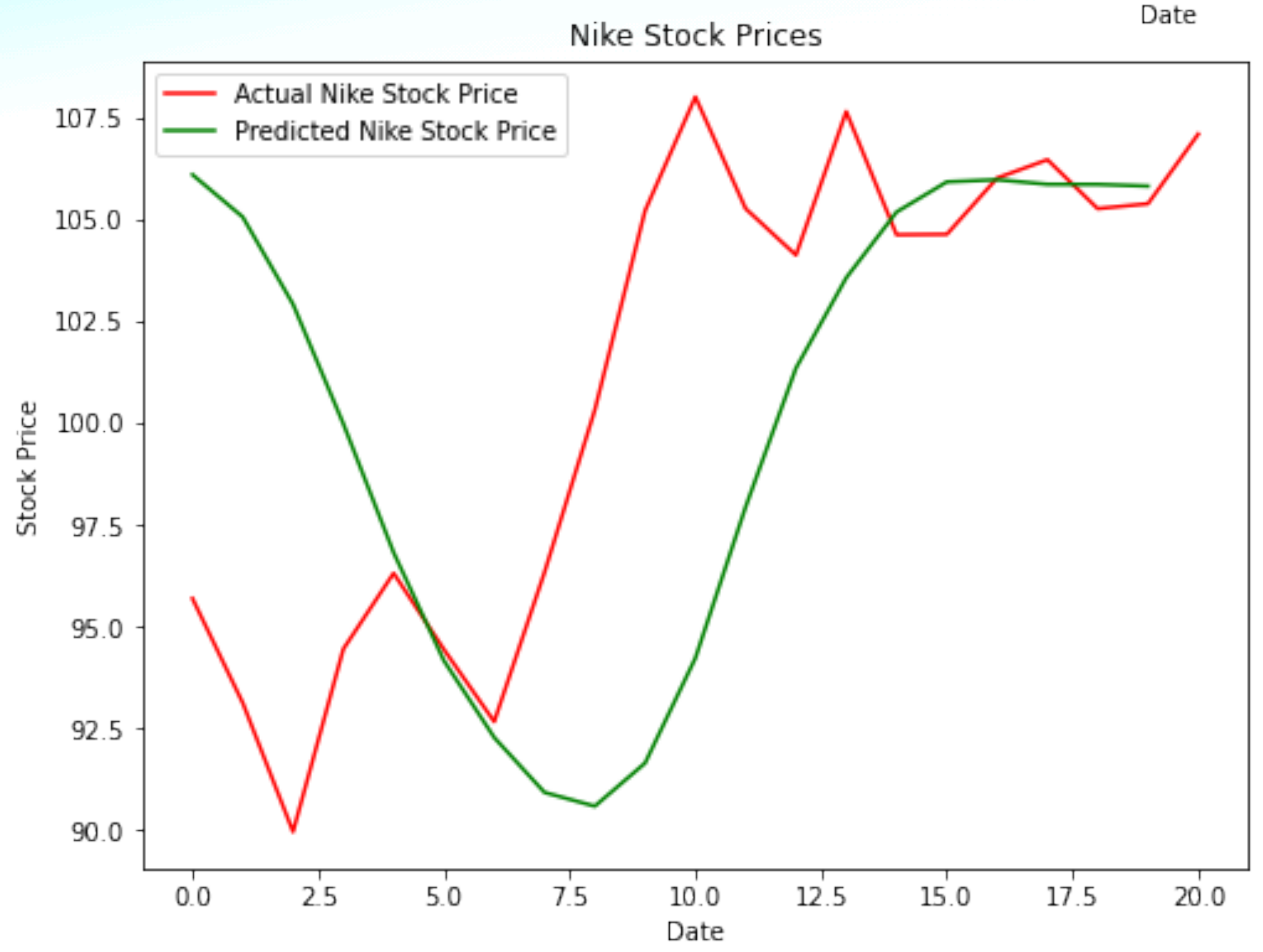


# Result

**What can we inference from this graph**

- Since we scaled our input feature, we used the inverse_transform() method of the scaler object on the predicted output to get the original output values.
- Finally, to compare the predicted output with the actual stock price values, you can plot the two values





## Reference

### links ...

Machine Learning book
https://legacy.cs.indiana.edu/classes/a310-dgerman/fall2022/resources/bicos/deep-learning.pdf
Project book
https://legacy.cs.indiana.edu/classes/a310-dgerman/fall2022/resources/sem01.pdf

Dataset

**https://finance.yahoo.com/quote/NKE?p=NKE&.tsrc=fin-srch**

**https://finance.yahoo.com/quote/FB/history?p=FB**