Predicting a binary valued parameter depending on OHLC Time series data

We have used the **Long-Term-Short-Memory (LSTM)** Framework to train our model on the given **OHLC - time series** data to predict the parameter **y**. LSTM is a **Recurrent Neural Network (RNN)** based architecture that is widely used in natural language processing and time series forecasting.The LSTM rectifies a huge issue that recurrent neural networks suffer from: short-memory. Using a series of 'gates,' each with its own RNN, the LSTM manages to **keep, forget or ignore** data points based on a probabilistic model.

LSTMs also help solve exploding and vanishing gradient problems. In simple terms, these problems are a result of **repeated weight adjustments** as a neural network trains. With repeated epochs, gradients become larger or smaller, and with each adjustment, it becomes easier for the network's gradients to **compound** in either direction. This compounding either makes the gradients way too large or way too small. While exploding and vanishing gradients are huge downsides of using traditional RNN's, LSTM architecture severely mitigates these issues.

After a prediction is made, it is **fed back** into the model to predict the next value in the sequence. With each prediction, some error is introduced into the model. To avoid exploding gradients, values are '**squashed**' via **sigmoid & tanh** activation functions prior to gate entrance & output.

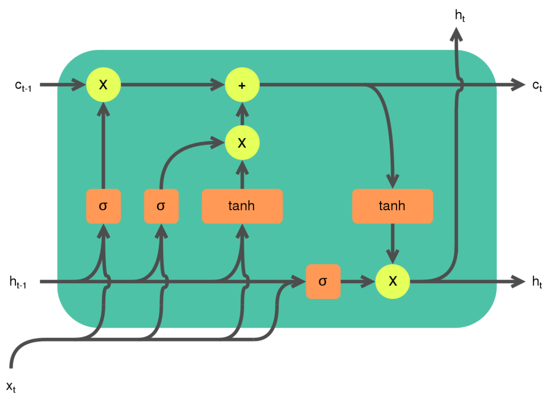
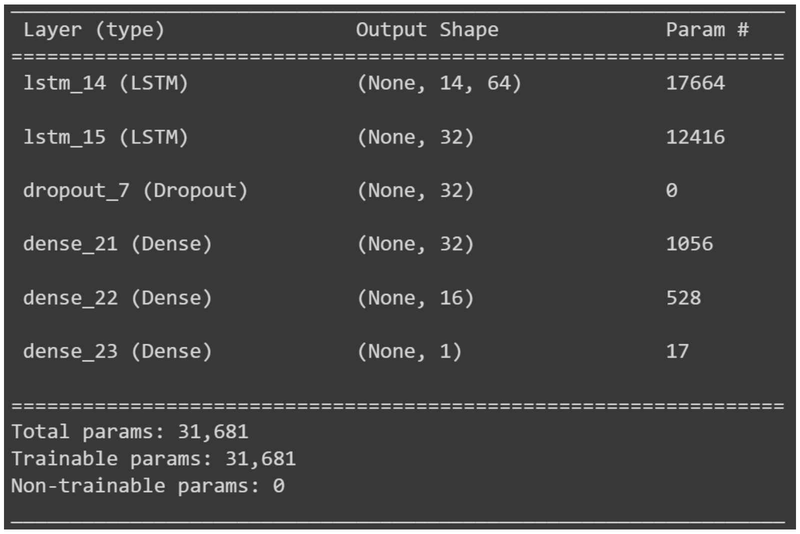


diagram of LSTM architecture

Initial Parameters

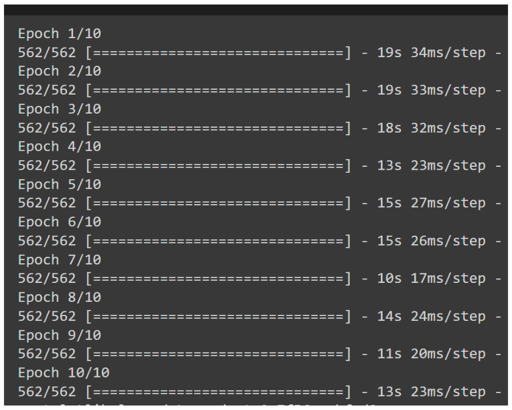
- `trainX = [], trainY = []` are the empty lists to be populated using formatted training data.
- `n_future = 1` is the number of days we want to look into the future based on the past days. `n_past = 14` number of past days we want to use to predict the future.

- We have incorporated **6 Layers** in our architecture, including the **two LSTM** Layers, **one dropout** layer, and **three dense** layers with a total of **31,681** trainable parameters in total. The details of the architecture is shown in the figure below.

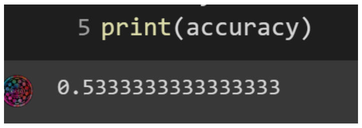


```
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_14 (LSTM)              (None, 14, 64)            17664

 lstm_15 (LSTM)              (None, 32)                12416

 dropout_7 (Dropout)         (None, 32)                0

 dense_21 (Dense)            (None, 32)                1056

 dense_22 (Dense)            (None, 16)                528

 dense_23 (Dense)            (None, 1)                 17

=================================================================
Total params: 31,681
Trainable params: 31,681
Non-trainable params: 0
```

- We have used "**Rectified Linear Unit(ReLU)**" Activation and the "**Adam**" optimiser to train our Model.The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero, and **Adam** is a replacement optimization algorithm for stochastic gradient descent for training deep learning models. Adam combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems.
- In our model, `batch_size=16, validation_split=0.1` are the preset parameters of the Training and Evaluation sets.
- We trained our LSTM Model for **10 - Epochs** and attained an accuracy of **53.3%** on the testset of the given data.



```
Epoch 1/10
562/562 [==============================] - 19s 34ms/step -
Epoch 2/10
562/562 [==============================] - 19s 33ms/step -
Epoch 3/10
562/562 [==============================] - 18s 32ms/step -
Epoch 4/10
562/562 [==============================] - 13s 23ms/step -
Epoch 5/10
562/562 [==============================] - 15s 27ms/step -
Epoch 6/10
562/562 [==============================] - 15s 26ms/step -
Epoch 7/10
562/562 [==============================] - 10s 17ms/step -
Epoch 8/10
562/562 [==============================] - 14s 24ms/step -
Epoch 9/10
562/562 [==============================] - 11s 20ms/step -
Epoch 10/10
562/562 [==============================] - 13s 23ms/step -
```



```
print(accuracy)

0.5333333333333333
```

- Our predictions for the given data are submitted in the filename "**predictions.csv**"