

## Deep Learning – Case Study

### Stock Price Prediction Using LSTM.

**Name:** Pooja Sinojiya

**Enrollment Number:** 18012021078

**Batch:** DL2

### 1. Introduction

Predicting stock prices is an uncertain task which is modelled using machine learning to predict the return on stocks. There are a lot of methods and tools used for the purpose of stock market prediction. The stock market is considered to be very dynamic and complex in nature. An accurate prediction of future prices may lead to a higher yield of profit for investors through stock investments. As per the predictions, investors will be able to pick the stocks that may give a higher return.

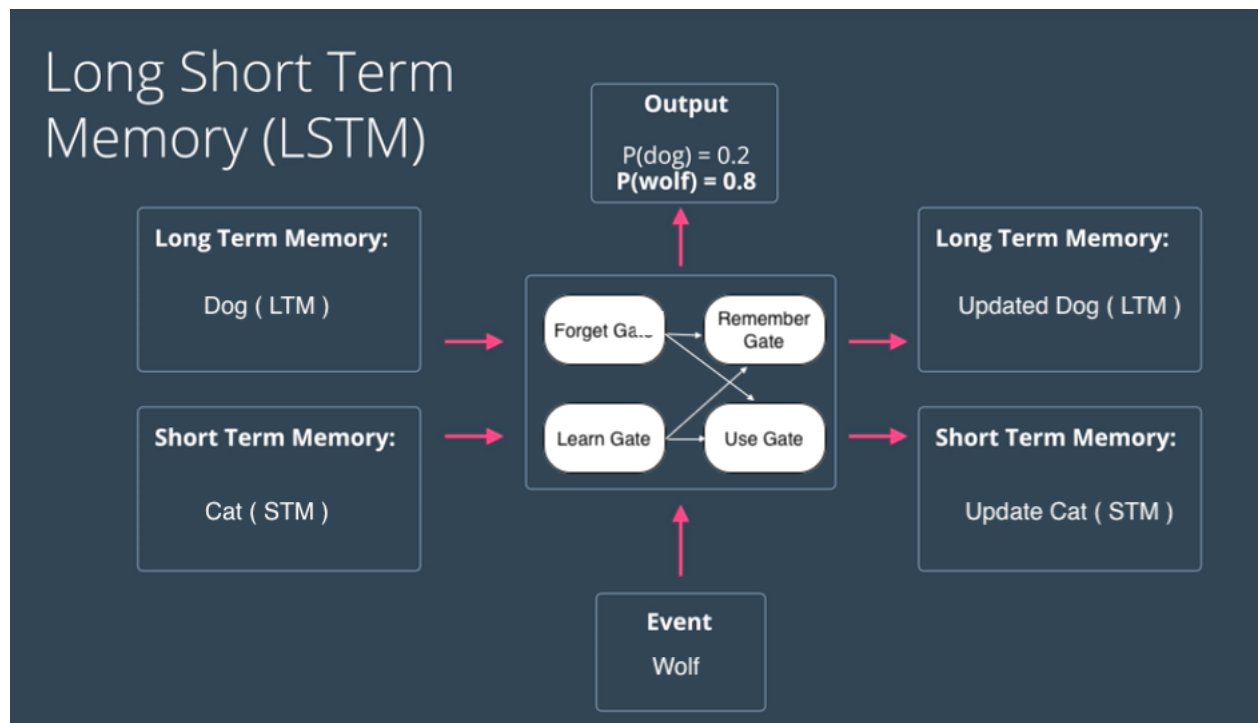
### 2. Tools and Technology.

Tools and Libraries	Usage
Keras	
Matplotlib	Matplotlib is a <b>cross-platform, data visualization and graphical plotting library for Python</b> and its numerical extension NumPy. As such, it offers a viable open source alternative to MATLAB.
Sklearn	Scikit-learn is probably the most useful library for <b>machine learning</b> in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression,

	clustering and dimensionality reduction.
--	--

### 3. Model Architecture

The basic difference between the architectures of RNNs and LSTMs is that the hidden layer of LSTM is a gated unit or gated cell. It consists of four layers that interact with one another in a way to produce the output of that cell along with the cell state. These two things are then passed onto the next hidden layer. Unlike RNNs which have got the only single neural net layer of tanh, LSTMs comprises of three logistic sigmoid gates and one tanh layer. Gates have been introduced in order to limit the information that is passed through the cell. They determine which part of the information will be needed by the next cell and which part is to be discarded. The output is usually in the range of 0-1 where '0' means 'reject all' and '1' means 'include all'.



### 4. Working

1. Take input the current input, the previous hidden state, and the previous internal cell state.

2. Calculate the values of the four different gates by following the below steps:-
  - For each gate, calculate the parameterized vectors for the current input and the previous hidden state by element-wise multiplication with the concerned vector with the respective weights for each gate.
  - Apply the respective activation function for each gate element-wise on the parameterized vectors. Below given is the list of the gates with the activation function to be applied for the gate.
3. Calculate the current internal cell state by first calculating the element-wise multiplication vector of the input gate and the input modulation gate, then calculate the element-wise multiplication vector of the forget gate and the previous internal cell state and then adding the two vectors.
4. Calculate the current hidden state by first taking the element-wise hyperbolic tangent of the current internal cell state vector and then performing element-wise multiplication with the output gate.

## 5. Code

[https://github.com/poojasinojiya02/DL\\_CaseStudy/blob/main/Pooja\\_Sinojiya\\_18012021078\\_DL2\\_Case%20Study\\_Report.docx](https://github.com/poojasinojiya02/DL_CaseStudy/blob/main/Pooja_Sinojiya_18012021078_DL2_Case%20Study_Report.docx)

## 6. Output

