|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Final Project Report  Leon Hayden / CSCI 59000 / 8-May-21 |  |
|  |  |  |  |
| Pipette dropping liquid in a petri dish | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | |
|  |  | TABLE OF CONTENTS |  |
|  |  | Introduction ..………....….….…………….….…….3  Method ..………..…..…..….….…….............…….4  Discussion …………..……..….….……..…….…….5  Results ..………..…....……..….….…..……….…….6 |  |



|  |  |  |
| --- | --- | --- |
| INTRODUCTION |  | |
| The lack of a viable simple to use stock predictor is hurting the middle class. It is leaving the opportunity to help build individual wealth left on the sideline. As either being to complex or out of the reach of a normal person by them feeling that they do not have the needed resources to compete. With the more established crowd, it is my intention to design a stock predictor that will help the common everyday person. Be able to help them to decide how much and what stick has the best possible return using previous stock price data. This stock price data will be from Nasdaq, I hope to enlighten and show one way to use previous data to help increase the chance of having a successful portfolio. | |  |
|  | |  |
|  | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Method | |  | |
| Recurrent Neural Networks (RNN)  As a human the ability to decipher words, pictures, logic is based on the ability to remember previous exposure to said information. A part of person ability figures out new experiences is based on previous information learned. To put this in layman’s terms if you wanted to classify an event happening at every point during a play. There would need to be a way to reason how every event flowed during the play. Since there would be no way for you to determine how the order of the scenes in the play was placed. To help with this a storage device called a neuron can be used to keep track of scene. This neuron which is the same as a human neuron. In that there are two inputs with a weighted and summed. If the weight is positive, would reflect excitatory connection. Negative values would lead to inhibitory connections, all weights are modified by a weight and summed. Which leads to a condition called linear combination an activation function controls the amplitude of the output.  RNN’s have a feature which is their internal state memory. This can lead to a finite and infinite impulse. Finite impulse network is a directed acyclic graph that can be broken down and placed in a feedforward neural network. Infinite impulse network is a directed cyclic graph that cannot be broken apart. Each does contain a storage feature that can be built that has another neural network or graph.  Figure : Basic Building Block Recurrent Neural Network  Representation of RNN both in folded and unfolded forms (Borges, 2018)  Legend:  The input, it can be a word in a sentence or some other type of sequential data.  The output for instance, what the network thinks the next word on a sentence should be given the previous words.  The main block of the RNN, it contains the weights and the activation functions of the network.  Represents the communication from one time-step to the other.  The above image represents a Fully Recurrent Neural Network in which all the neurons output is connected to the input of all neurons. This is a general topology of an RNN is some cases this can be misleading due to most RNN. Are based in layers as which is seen above. Even though the figure shows that there are layers, these in effect different steps in time of the same network. That is shown where t-1 and t+1 for time.  Long Short - Term Memory (LSTM)  Consists   * Cell: Entire Blue Area * Input Gate: Left hand side Ct -1, Ht -1 * Output Gate Right had side Ct, Ht * Forget Gate Xt   (IntelliPaat, 2020) | | |  |
| LTSM Advantage / Disadvantage |  | |  |
| * Acts like a human neuron, in that it maintains its state over time. (IntelliPaat, 2020) * Has feedback connections, that can process single data points, and entire sequences of data. It truly shows that it is worth with unsegmented, connected handwriting recognition, speech recognition as well as anomaly detection in network traffic. * Better predictor of time series events * Weights are shared across time. * Computation considers historical information | * Computation being slow. * Difficulty of accessing information from a long time ago. * Cannot consider any future input for the current state | |  |

What does the LSTM do that an RNN does not, it can keep a state so it can remember the past for instance?

If I have two sentences:

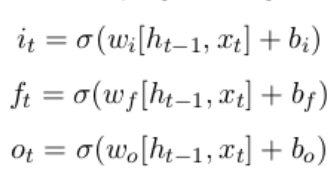
The cat which already ate [pause] was full.

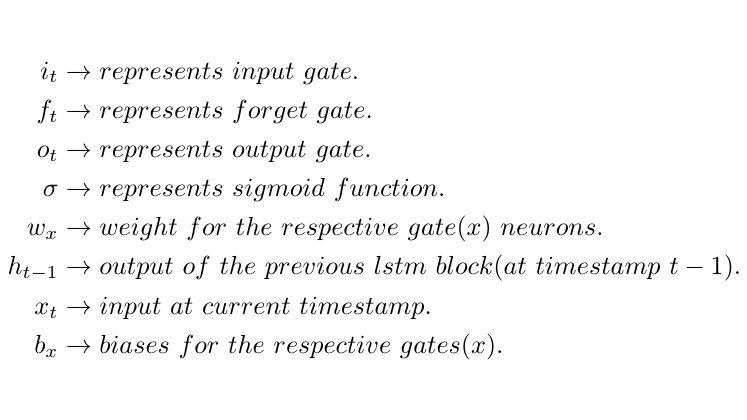
The cat which already ate [pause] were full.

The state of “was” and “were” needed to be remembered. “Was” state is a singularly possessive so it needs to be remembered, the “were” are used for the subject the cat. (Thakur, 2018)

Construction of LSTM consists of three gates:

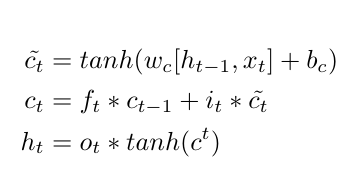
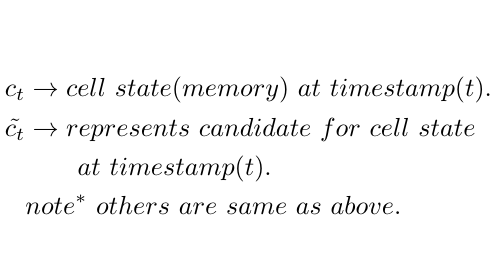
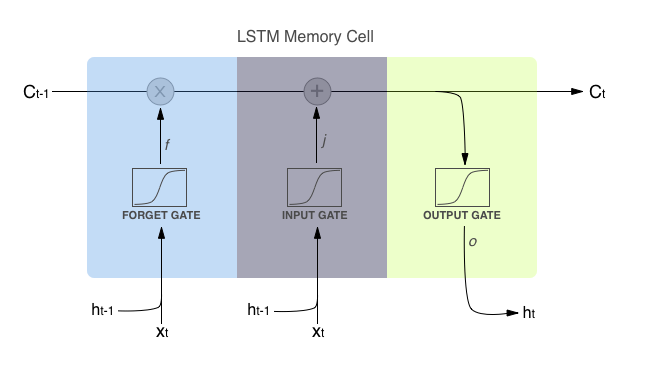
* Input gate = it This informs what new information that is going to store in the cell
* Forget gate =ft What information to throw away from the cell
* Output gate =ot Gate which is used to provide the activation for the final gate output at time stamp “t”

 Long Short - Term Memory (LSTM) Equations



(Thakur, 2018)

The equations for the cell state, candidate cell state and the final output:

 (Thakur, 2018)

To get a memory vector for the same timestamp (c\_{t}) calculates the candidate. From the equation any time stamp the state know what is needed to forget the previous state. [ i.e., f\_{t} \* c\_{t-1}] & what is needed from the current timestamp (i.e., i \_{t} \* c’{t}). As the cell state is filtered which passes through the activation that shows what portion should appear as the output of current LSTM at the timestamp.

Passing h\_{t} the output from current LSTM through the SoftMax layer to get the predicted output (y\_{t}) from the current block.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Discussion | | |  | |
| I was looking for a way that would allow for proper way to choose a stock on New York Stock Exchange, NASDAQ, or any listed security, option, currency, commodity. That could be understood by the common lay man and help them understand how to pick an asset. A confusing, difficult chore for most people, which unfortunately left many opportunities to increase wealth on the side, where they can help with increasing net worth along with helping in making the,” American Dream” a reality.  Developing it was decided that the assistant:   * Be easy to operate. * Use Yahoo API stock information. * Analyze one stock per request. * Variable date selector * Technical analysis of stock * Implementation of Neural net technology | | | |  |
| Long Short-Term Memory was chosen due to ability to hold information and help with the decision process. It does need a data frame for processing and is appropriate for prediction and time-series data. This model has a higher accuracy for a short time frame when working with larger data frames.  The API allowed for use of any data that was stored on Yahoo’s server so if there was information stored it could be used.  Stock ticker symbol can be gleam from an internet search. All libraries that were going to be used are listed are listed in cell one.  Cell 2: Stock symbol input, and price  Cell 3: Dates inputted  Cell 4: Established Data Frame, Data frame shape, output showed number of rows and columns.  Cell 5: Building of interactive graph, technical indicators.  Cell 6: Normalization of data frame with heat map to indicate that all relevant columns have been normalized.  Cell 7: Create training data frame from “close” price.  Cell 8: Scaling data for preprocessing  Cell 9: Dividing training database into independent [*x-train dataset*] dependent [*y-train dataset*], using 60-day past data for training.  Cell 10: Convert data into NumPy array.  Cell 11: Reshape *x-train* data.  Cell 12: Develop LSTM  Cell 13: Compile model and Mean square error for loss function and optimizer.  Cell 14: Building training dataset  Cell 15: Visualize training dataset and validation loss.  Cell 16: Developing new testing dataset new array that contains the scaled values.  Cell 17: Convert to NumPy array  Cell 18: Reshape data from 2D to 3D.  Cell 19: Get predicated stock price value.  Cell 20: Find Root Mean Square Error  Cell 21: Combining projection, validation, predictions.  Cell 22: Showing Close vs Predictions.  Cell 23: Showing final projected price for the next day |  | A confusing difficult chore for most people, which unfortunately left many opportunities for increased wealth on the side | |  |
| How things work  After the data has been gathered and the data frame has been created. The first analysis tool is the Bollinger Bands, this tool is mostly used for showing oversold and overbought signs. There are three lines that describe these technical signs. The upper and lower lines are based on two standard deviations based on moving average or intermediate-term trend. Since it is a moving average, it can be adjusted as needed to fit your analysis. (Hayes, 2021).  So, since the volatility is shown with standard deviation the more volatile the bands widen, during less volatile periods the bands shrink. (Hayes, 2021)  MACD Moving Average Convergence Divergence, is a trend following momentum indicator that shows the relationship between two moving averages of the stock price. This is calculated by subtracting 26 period from a 12-period exponential moving average. (Fernando, 2021) By calculation, the “MACD line’ this line is plotted on top of MACD, this can be used as a signal for buy and sell signals. Signals can be interpreted that when the line is crossed above this line a sell can be executed, or when it goes below a short can be executed. If there are several crossovers depending on the speed of the crossover, can be taken as a signal of market is overbought or oversold. (Fernando, 2021)  The primary usage for this indicator is that it can be used as an indicator of bullish or bearish price movement (price getting stronger, weaker).  Visualization, by using a heat map shows where the correlation took place between open, high, low, close, adj close fields. Volume field does indicate that there is no correlation to the previous fields and was left out of the program.  Spilt data in testing and training segments, where closing price of the data frame was scaled for 80% was set aside for model training and 20% for testing.  From this I created a data set that contains 60 days of training data with close price. This would give out a predication on the 61 day from the *x-train* independent *& y-train* dependent data set. For LSTM to work these data need to be converted to NumPy array. Once this has been been reshaped from 2D to 3D which allows the model to be created with two layers. It has 50 neurons with two dense layers (dense layers are non-linear functions that when stacked can be used to solve complex mathematical functions (Moawad, 2019).) so by building two dense layers this helps in developing the picker model by solving complex problems. To find error loss Mean Square Error formula was used.  Epoch value the value that is the optimizer program gets when it is figuring the training database. It was found that if the data set is kept around 20 years the epoch build time for 1 second for each of the 200. However, when data goes up to 40 years it can take 35 seconds to build each of the epoch. The lower the loss and the better the number and fit.is shown.  Training Validation Loss graph shows what the loss or error that was in the data. As shown in the graph after the training database runs the error rate goes to zero. While validation is jagged at first, and epochs run the line begins to smooth out which indicates that it is a good fit. At this point I would create a testing dataset with the last 60 days scaled. From here to get find the stock price by unscaling in the array. Root Mean Squared Error [RMSE] shows how if the model is good changes with the stock selected and the time frame, Longer time frame, means the RSME value that is,” good”. Can change depending on the length of the stock dates you are selecting.  Trained model graph, shows training, validation, predictions based with closing stock price. The amount of error is low, this is shown towards the end for training along with validation.  The closing price is and the projection price where close to each other which showed that the model is good, good enough for use as an everyday picker. To get a reasonable prediction the prediction price is ran through a normalizer. Helping it be smoothed out and making it give an,” truer to live response”. | | | |  |

|  |  |
| --- | --- |
| Results |  |

|  |  |
| --- | --- |
| This prediction picker works. I can say that the price that is shown will be present the following day with extended trading hours considered. Of course, no external factors would be present. External factors are events that cannot be forecasted or anticipated in advance. Examples, Insurrection on January 6, 2021, Covid-19 shutdown, September 11 attacks, certain bills getting passed into law, etc. I can say that I was surprised about how useful LSTM is, I can see many applications for this. To me this was the most important application I had learned this semester. |  |
|  |  |
|  |  |

# References

Borges, P. (2018, October 18). Deep Learning: Recurrent Neural Networks. *Medium*.

Fernando, J. (2021, April 29). *Moving Average Convergence Divergence (MACD)*. Retrieved from Investopedia: https://www.investopedia.com/terms/m/macd.asp

Hayes, A. (2021, April 30). *Bollinger Band Definition*. Retrieved from Investopedia: https://www.investopedia.com/terms/b/bollingerbands.asp

IntelliPaat. (2020, May 20). *What is LSTM*. Retrieved from IntelliPaat: https://intellipaat.com/blog/what-is-lstm/

Moawad, A. (2019, October 8). *Dense layers explained in a simple way*. Retrieved from Medium: https://medium.com/datathings/dense-layers-explained-in-a-simple-way-62fe1db0ed75

Thakur, D. (2018, July 6). *LSTM and it's Equations*. Retrieved from Medium : https://medium.com/@divyanshu132/lstm-and-its-equations-5ee9246d04af