Galal Karim Travail de Maturité

ARTIFICIAL INTELLIGENCE IN FINANCE



Maître(sse) - accompagnante(e) : M Morel

INSTITUT FLORIMONT 2020-21

Table of Contents

[**Introduction**](#_2d14n2kp3121) **3**

[**1 Risk Management**](#_nujxqbq2173v) **4**

[**1.1 Coronavirus Risk**](#_7julacf1t0xl) **4**

[1.1.1 Data](#_bqju8ri18bcd) 4

[1.1.2 Model](#_maink6m3g6qm) 5

[1.1.3 Results](#_vllebdclosj) 6

[**1.2 US Political Risk**](#_7sufqiclemq) **6**

[1.2.1 Data](#_28l6d99fildx) 7

[1.2.2 Model](#_1umzj7cov3a2) 7

[1.2.3 Results](#_16v23qynj8a1) 8

[**2 AI Investing**](#_wvvsi2s8qwur) **9**

[**2.1 Statistical-based Investment**](#_o82nzfcumpuu) **9**

[2.1.1 Data](#_5tijsj95xpig) 9

[2.1.2 Model](#_psvo9nio20pa) 11

[2.1.3 Results](#_b2je3mgtlivh) 11

[**2.2 Stock News Model**](#_9bscw9byhitu) **12**

[2.2.1 Data](#_pu1r2qlvuvts) 12

[2.2.2 Model](#_yxhk7vlxeaoo) 13

[2.2.3 Results](#_5mny4q5uxmyo) 13

[**Testing**](#_ysuonpqrp0dt) **14**

[**Conclusion**](#_uioh0zs8xpnu) **16**

[**Bibliography**](#_8vcw9w9e073r) **17**

[**Annexes**](#_oozc5un9nld0) **19**

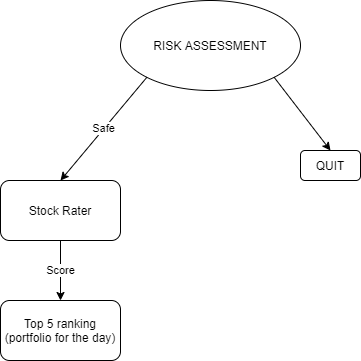
[Background](#_9x1iujqkht) 19

[Neural Network](#_y1o0tqw7dwu5) 20

# 

# Introduction

Artificial intelligence has not been around for very long. However, it has had immense effects on the world we currently live in, from elevators to our suggested movies on Netflix. One field that was significantly affected is finance. Big banks and investment firms worldwide have heavily engaged in the new technology in hopes of a model that will start turning significant profits, ultimately dropping the expensive requirement of unreliable traders. These models' functioning is based on the data fed into them and the models' architectures. Tweaking these is how the enterprises may start to be profitable. However, the challenge is finding the correct data to use and the best architecture to train. At the time of this paper, the circumstances are very precarious with social unrest and a once in a century event(COVID-19). Therefore, this leads us to our main question: How to build a reliable AI model for financial predictions during the massive uncertainties of the coronavirus and political instability? Firstly, to see if we can safely invest in the stock market(S&P 500), we must first have some risk measures in place; therefore, this will entail our model's first part. If the market passes this test, we move on to the second half of our solution that will be a more investment-based approach to the work where we will purely find out whether a stock is worth investing or not. Each stock will be given a rating between 0-1. They will then all be sorted and our portfolio for the following day will be built out of the top 5 stocks that our models predicted. We will be day trading. Therefore, our models will have all their predictions for the following day. Finally, we shall implement these models to live and see how they work in real-time.



# 1 Risk Management

## 1.1 Coronavirus Risk

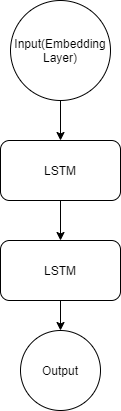
The stock market has been walloped by COVID-19 and has become unavoidable to account for in our risk assessment measures. Since the coronavirus has proliferated in the US, we have noticed high volatility in the market as trading is becoming more and more linked to the virus's progression[[1]](#footnote-0). Many have started selling out of fear, making markets decline. However, the US government and Federal Reserve Board(FED) have been making efforts to instil hope in the economy by pumping money into it, buying $560 billions of corporate bonds in April 2020 which is double the average level, a bill that gives businesses around America a $2 trillion lifeline which is double the stimulus package passed in 2009.[[2]](#footnote-1) This storm has ravaged the economy, but markets have been working esoterically with a seemingly heavy reliance on COVID-19[[3]](#footnote-2). I have found it imperative to consider it in our risk management. We shall build a model that predicts the effect that the virus's progression will have on the stock market to see whether it is safe enough to invest.

### 1.1.1 Data

As the coronavirus is a novel occurrence, there is not enough data to build a just model. Therefore, we equally have to look at past pandemics such as SARS 2003 and its effect on the economy then. Using the SARS virus as a starting point for our data is extremely helpful as they are similar in many ways: spread in the same way, both had significant death tolls, and both had severe effects on the economy[[4]](#footnote-3).We shall be using the “SARS 2003 Outbreak Dataset” from Kaggle by Devakumar KP (<https://bit.ly/35S0Rul>) for the SARS data. For the COVID-19 data, we shall be using thevirustracker live API (<https://bit.ly/2T0hSx8>), which will give us up-to-date information to renew our model once in a while with more data to get more accurate results. With the batches of data, we shall extract the percentage change of people infected as well as the percentage of deaths attributed to the particular virus so that we can not only see a holistic view of the consequences but also because we can normalise the data (as the percentages all remain between 1 and 0). We will be taking the percentage change of the cases and deaths of the past three days in the United States for the features. We will look at the standard deviation (volatility) of the S&P 500 for the next day using hourly data for the classes. If the standard deviation is over ten, we consider the stock market too volatile and risky to invest in. Whereas if the standard deviation is under ten, we shall consider the stock market tame enough to invest in to run the risk as little as possible. We split up our data into two batches, 80% being our training data, and 20% being our testing data.

### 1.1.2 Model

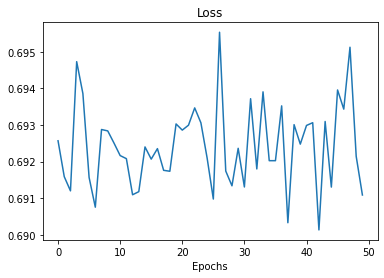
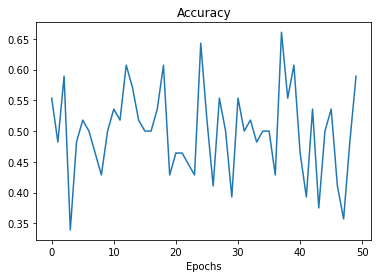
For our first model, we shall be looking back at the last three days to see how it will affect the next trading day's volatility as we do not want to invest when the stock market is being too unpredictable. We run our data through a recurrent LSTM neural network. This network type should work best for us as our data is based on time, and it is best for sequential pattern recognition.[[5]](#footnote-4) As we do not have much data, and we would like our model to generalise to the best of its ability, we should restrain the neurons and layers in our model. This is why I have chosen two layers with 16 neurons each. Each layer will have a dropout rate of 20%. For our optimiser we shall be using Nadam optimiser as it proves to get the best training and validation losses out of all optimisers[[6]](#footnote-5); to go along with it we choose a learning rate of 1e-10. Finally, for our loss function we shall be using sparse categorical cross entropy. We are dealing with a multi-class classification problem with data entries that can only belong to one class.



### 1.1.3 Results

After running our model for 50 epochs, we started getting favourable results. In the end, we hit an accuracy of around 57% and a loss of 0.691. To better understand our results, we must use our testing data to analyse our model. We received an accuracy of 60% in which it was able to predict non-volatile periods at a 40% accuracy whereas it was able to predict 70% a volatile market. This means that it is more likely to predict a non-volatile market as volatile rather than the inverse. This is somewhat serendipitous as this model is less likely to take risks itself which fits ideally for our risk assessment part of our financial prediction system.

**TRAINING GRAPHS**



## 1.2 US Political Risk

Outside of the coronavirus, the government plays a very considerable role in the economy and the markets, primarily the Trump's administration's erratic actions[[7]](#footnote-6), for example, the China-United States trade war which led to massive stock market instability as it brought hardships to American farmers thus higher prices for consumers[[8]](#footnote-7). Now that the coronavirus has become prevalent and a strong sense of social unrest with the killing of George Floyd, the USA's government responses have become ever more crucial to implement in our risk assessments.[[9]](#footnote-8) Considering US politics in our model will give us an insight into what traders may be thinking and whether enough confidence has been instilled in the markets that it is safe enough to trade safely.

### 1.2.1 Data

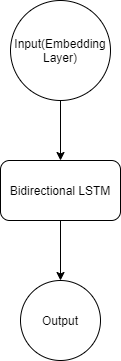
The data collection part of this model will be the toughest as there is no predefined dataset that we can use that collects all US political news articles. To create our dataset, we shall be looking at the Guardians’ US Politics section (https://www.theguardian.com/us-news/us-politics) and use a parser to collect all the titles of articles written on a particular day. We shall only be using titles to give us a better indication of what the story is talking about rather than the ambiguity that we can receive from reading the entire article.[[10]](#footnote-9) Much like in our coronavirus risk assessment, we will be looking at the next trading day’s volatility to correctly classify the articles. The next issue we face is transforming the data or titles into a data type that the neural network can use.

To make the text usable for my neural network, we have to convert the text into numerical values. To do so, I vectorised each word using google’s Word2Vec[[11]](#footnote-10). The model is trained to reconstruct the linguistic context of words. It does so by assigning each unique word with a corresponding vector in space positioned in the vector space, so words that share familiar contexts are located near one another. For example, “bad” and “terrible” have similar word vectors, which means that our vocabulary is limited to the words used in the dataset and the entire English dictionary.

Now that we have translated our text to word vectors, we can shape the data to fit into our network. We have to define the size of our input to be able to use our network. Therefore, we split the data into chunks of a specified sequence length; if we cannot achieve that length(150), we will pad the sequences with nulls. We split up our data into two batches, 80% being our training data, and 20% being our testing data.

### 1.2.2 Model

Once the data shaping is done, we can fit it into our embedding layer and start modelling our neural network. The architecture of the neural network I decided to use is recurrent as it is the best for text sentiment analysis because words follow a logical sequence. It will be composed of 1 embedding layer which stores a lookup table to map words represented by numeric indexes to their dense vector representations and one layer of 20 neurons Bidirectional Long Short Term Memory cells. Each layer will have a dropout rate of 20%. Bidirectional LSTMs train one LSTM layer on the input sequence and second LSTM layer on a reversed copy of the input sequences. This is extremely helpful in our case as unlike a vanilla LSTM, our model not only provides the capability of learning long term dependencies, but it also provides more context for learning sequences. Our output will be a basic dense layer with softmax activation function as we are doing binary classification. We will be using Nadam optimizer with a learning rate of 1e-5, and it will be using sparse categorical entropy as our loss function. The model will determine whether there is a considerable amount of political risk (by classing the input data with 1) or if the political climate is safe enough to invest (by classing the input data with 0).



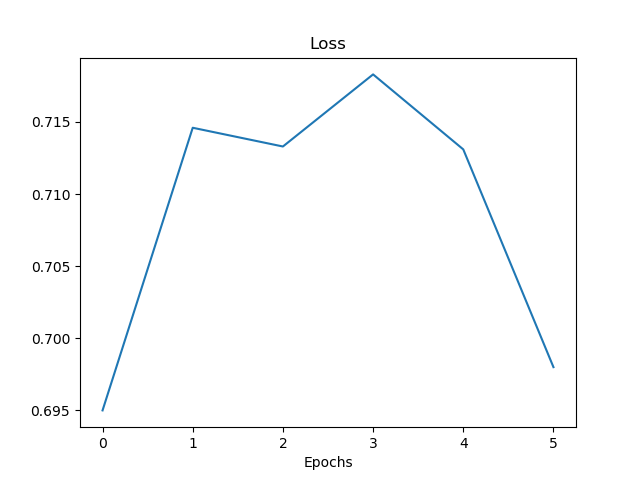
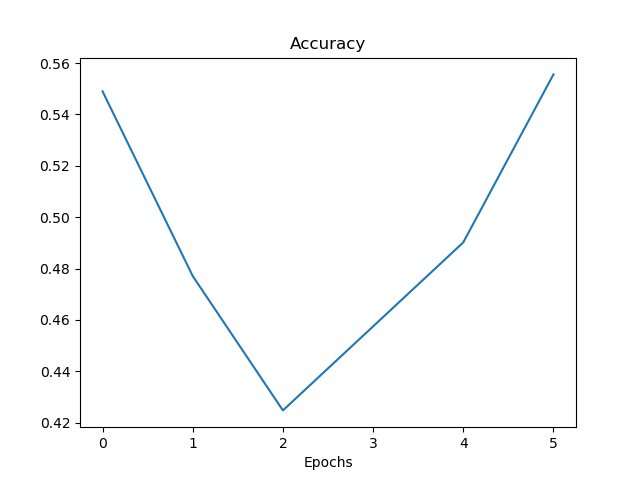


### 

### 1.2.3 Results

After running the model for six epochs(due to the depreciation of accuracy in the following epochs), we received satisfying results. We had hit accuracy of around 55% and a loss of approximately 0.698. To see whether the model works or not, we had to evaluate it with our training data which gave us an accuracy of around 56% and a loss of 0.688. For a deeper dive into our testing data evaluation, we noticed that it predicted correctly that the market would be calm after reading US political news 57% of the time. In contrast, it predicted correctly that the market would be volatile 55% of the time. So it shows us that it is a relatively reliable model to use in our risk assessments.

**TRAINING GRAPHS**



# 2 AI Investing

Now that the model has passed our risk assessment test and the stock market is deemed stable enough to invest in, we can start taking a look at how we can invest properly in the markets.

## 2.1 Statistical-based Investment

Our first step in our investment techniques is the statistical-based investment method. Much like human investors, our model will have to look at specific stocks and understand their movement to predict how the prices will evolve throughout the following days. Investors usually look at stock charts that present various information about a given stock: its price evolution and an array of crucial indicators that present the data in a way that traders can use to their advantage to make more educated choices. Our model will have to figure out what these different signals mean and how to use them to maximise profits.

### 2.1.1 Data

It would be frivolous and too challenging to make our model analyse pictures of stock charts to predict future price movement. Therefore, instead, we will take the OHLCV data from the yahoo finance API to manipulate the data ourselves and then inject them into our model. However, we first have to look at what technical indicators could be valuable to read the stock market.[[12]](#footnote-11) For our purposes, we shall go for the most popular and effective ones: 7 and 21-day moving average, MACD, Bollinger bands, 7 and 21-day exponential moving average and RSI.

The n day moving average is the unweighted mean of the previous n data Following the same line of logic, the ema (or exponential moving average) is a specific type of moving average that places more importance on newer data; it does this by continuously weighting newer data, to make our model adaptable we must take the percentage change of these moving averages as they purely depend on a given stock's price. This tool is not only useful by itself but also it can be used to calculate the MACD. The MACD (or moving average convergence divergence) is an indicator which allows a trader to access a stock's momentum and strength which is calculated by subtracting the 26 ema by the 12 ema. Nevertheless, we also have to consider volatility indicators, Bollinger bands. Constituted out of three bands, the middle one being a simple 20 day MA(Moving Average). The lower line is two standard deviations below the 20 days MA, and the upper line is two standard deviations above. The Bollinger Bands indicate that if the stock exits these bands' limit, then it is highly volatile and will snap back into its bands soon. A highly volatile stock is also less favourable, as it implies a high risk. Our last technical indicator is the RSI(or relative strength index) which like MACD is a momentum indicator. It is used to measure the significance of recent price changes to determine whether a stock is being overbought or oversold; much like the MACD will use the raw values of the RSI for our model as it is not stock specific. Finally, we shall also be using the percentage changes of OHLCV prices to give us more information about a stock.

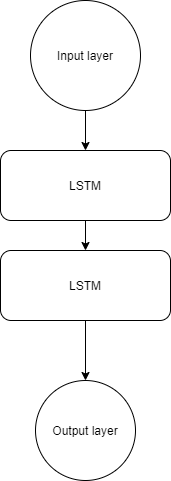
* 



However, after having collected this data, we must first clean it up to become more favourable to our neural network. Therefore, we must first normalise the data using the minmaxscaler function, which will put all the values between 1 and 0. Next, we must balance our dataset, to not make our model instinctively favour a specific output. We split up our data into two batches, 80% being our training data, and 20% being our testing data.

### 2.1.2 Model

After having amassed all the vast and insightful data, we must try to find a model that best fits it. Our model will collect the stock data from the past 60 days (counting all the 18 indicators) and predict what the stock will look like in 3 days. Again, as we are looking at sequential data, we recommend using an LSTM recurrent neural network. After several tests, I found the best model to have 2 LSTM with 60 and 180 neurons. Each layer will have a dropout rate of 20%. For our optimiser, we shall be using Nadam with a learning rate of 1e-20. Finally, we shall be using sparse categorical cross-entropy as our loss function since our output layer will have two neurons (sell or buy).

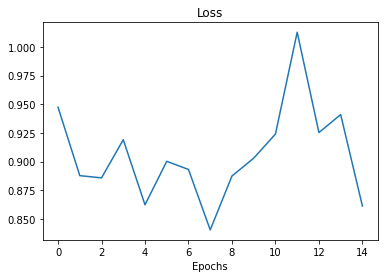
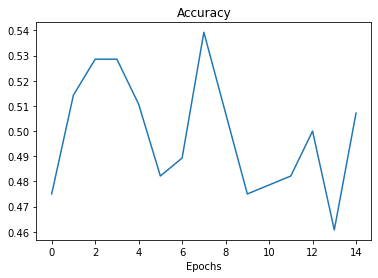
****



### 2.1.3 Results

After running the model for 15 epochs with a batch size of 6, we hit accuracy of approximately 51% and a loss of around 0.86. To properly analyse the model, we must evaluate it to see whether it is indeed reliable. On the test data, we got an accuracy of 52% and a loss of 0.73, which performed better than during training. To dive deeper into our testing, we must look if there are any unreasonable biases or overfitting taking place. Our model predicted a sell correctly 50% of the time, whereas it predicted a buy with a success rate of 54%. Therefore, we can conclude that our model tends to predict buys instead of sells which is not ideal; the AI does not seem to have any significant default or bias. So in the end, albeit the model is not perfect, it mostly performs better than random and is thus adequate for us. However, the 51% accuracy is not statistically significant to us to prove that it works. Even though it is one percent over random, it is too close to 50% to pull any real conclusions.

**TRAINING GRAPHS**



## 2.2 Stock News Model

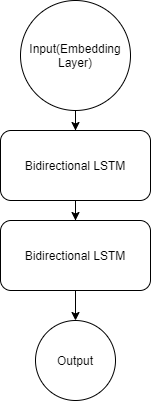
To predict a stock's movement, it is not enough to purely look at its data; we must equally consider outside events/circumstances that might affect it. We can use the news to analyse how a company is performing and whether anything notable could take a toll on their evolution. More specifically, the articles may cover quarterly earnings reports, new product releases, scandals within the company, etc. These would give us fascinating insight into the company and understand how these external circumstances may affect it. Many investors indeed prefer basing their predictions over the news articles' sentiment written about a ticker they would like to invest in.[[13]](#footnote-12)Therefore, it is crucial that our model can with enough confidence predict the sentiment of a given stock and how the news articles will affect it.

### 2.2.1 Data

Much like our previous model in the risk assessment section where we used the guardian's political news articles to predict volatility, we shall be using Reuter's stock news to predict the outcome on a stock price. To collect the data, we created a parser that went through every company's stock in the S&P 500 to collect all the news articles available. After having amassed approximately 7000 news articles, we extracted the title and blurb, preprocessed the words (using the Word2Vec model) and padded the titles until we reached a sequence length of 150. To give our model some context of the stock, instead of just using the titles of only the day before, we will be feeding it headlines from the past three days, giving us a more comprehensive insight into how the stock will react. To classify the data, we look at how the stock evolved the next day. If there was a rise, we labelled it as one whereas if the stock dropped, it received a label of 0. Finally, we split up the training data (80%) and testing (20%).

### 2.2.2 Model

Now that we have the data adequately cleaned and set up for neural network use, we can build a model that fits our data type. If we look at it, we will need a useful model with sequential predictions that is aware of context. Therefore, similarly to our US Politics model, we shall be using Bidirectional LSTM recurrent network. Thus, our model will encapsulate an embedding layer and two bidirectional LSTM cells with 80 and 100 neurons with a dropout of 30%. We can allow ourselves to raise the number of layers and neurons this much due to the more substantial amount of data which means we can use the extra weights to make our model generalise more accurately. For our optimiser, we shall be using Nadam with a learning rate of 1e-12 and sparse categorical cross-entropy as our loss function.

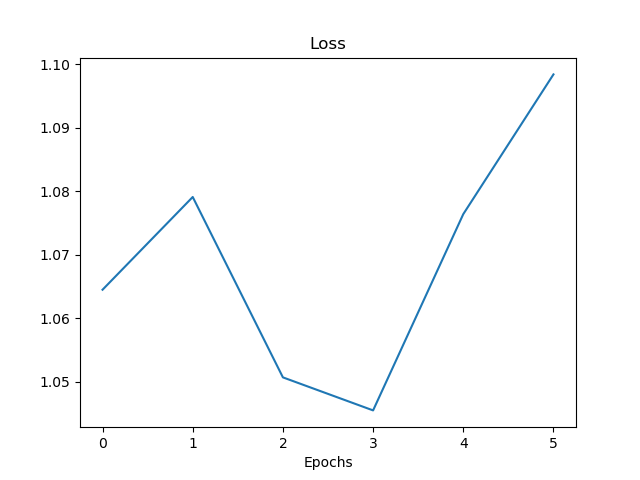
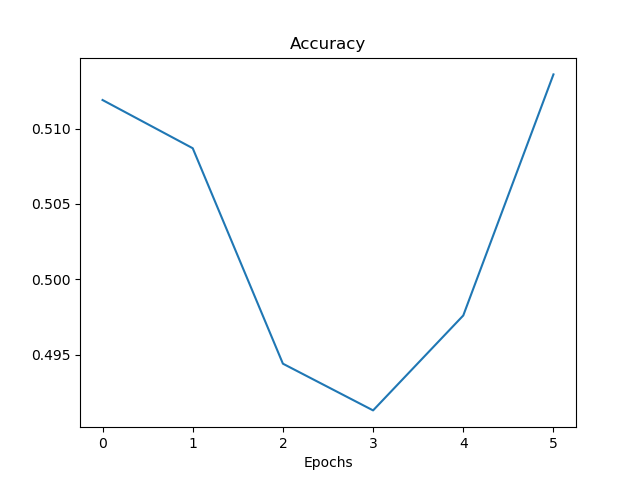




### 2.2.3 Results

After running our model for six epochs with a batch size of 42, we started to get pleasing results. We hit accuracy of around 51% and a loss of approximately 1.09. However, the training results do not give a valuable insight into what the model can do in the real world; therefore, we must run it through our test data. After running a model evaluation, we were able to reach an accuracy of 52%. Once we take a deeper dive into the results, we can see that the model slightly holds up. It was able to predict a sell 52% of the time and same with a buy. Therefore, we can say that it is a genuinely balanced(no biases) and more or less reliable model for our uses. This part of our stock movement prediction concludes that the AI seems to lean on the random side more than anything due to the statistically insignificant results we get on our training and testing data.

**TRAINING GRAPHS**

****

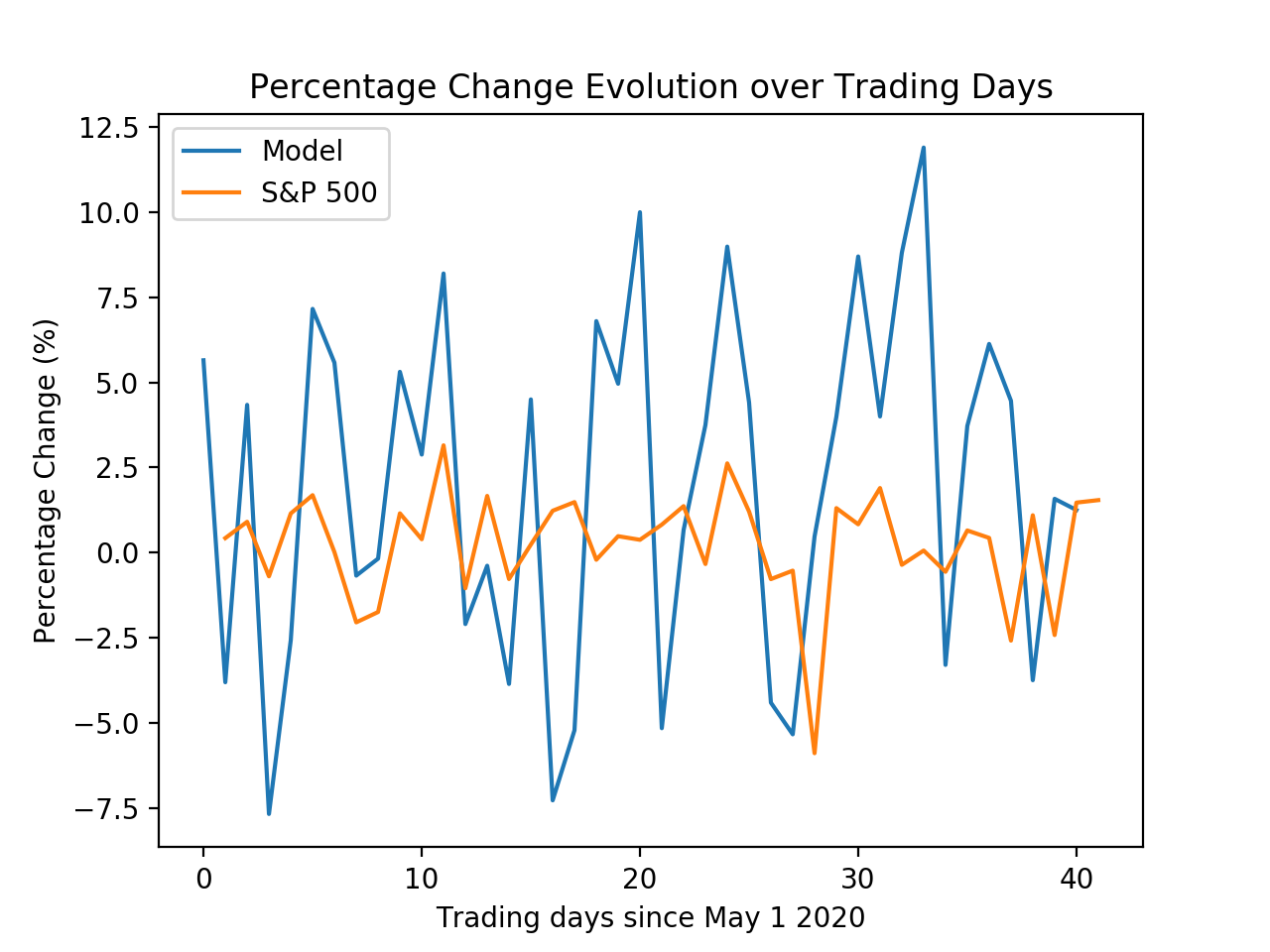
# Testing

After completing and compiling all our AI models, we now have to test in real-time to see whether they would be profitable in uncharted environments. We first have to concatenate the risk models, and next, we shall have to join the stock prediction models. The module that encompasses all will predict the following day's risk and price movement based on the real-time data that we feed it. I shall briefly explain and recap how the models will compile and work together; next, I shall move onto the trading module's execution.

Before I run the stock prediction, I must get the risk assessment results so that I know whether it is worth putting my money in the market. For this model, we have chosen two significant risk factors: US political risk and Coronavirus risk. Both models will give me a number between 0 and 1 to indicate whether the risk is predominant or negligible. We are taking the confidence that the model believes that the market is volatile. To figure out the overall risk assessment, the program will average the two resulting figures. If the result is over 0.5, then the risk is too high, whereas under 0.5. Once passed the risk assessment test, we can move on to the stock prediction.

Now that we have established that the market environment is safe enough for us to engage in, we can permit ourselves to analyse our options and figure out which stocks would build the most profitable portfolio. Much like our risk model, we have to split this one into multiple parts that we have previously discussed: statistical-based analysis and news sentiment analysis. Both of these models will provide us with a result between 0 and 1. After going through every stock individually, each ticker will have a rating assigned to it by averaging the models' values, which means that we can rank the stocks by rating. Finally, we will take the top 5 stocks chosen by the module and see how they perform during the trading day compared to the S&P 500.

Finally, to fairly assess our model, I will run the program for two months (May through mid-July). At the end of this trading period, we received the results. On the first of July, the S&P 500 was up 0.2348% whereas our model was up 1.669%, which confirms that our model beat the market by 1.434% during the May-June period. In the end, our work proved roughly successful.



We can see that our model often selected a portfolio that had substantial volatility: often either having very high gains, e.g. reaching 12.5% or very high losses, e.g. -7.5%. However, we do not necessarily want our portfolios to be too volatile, leading to massive losses. However, we seemed lucky at the end of the day, as we were able to average more significant gains than losses. Moreover, we can also notice that occasionally when the market went up, our model would equally go up by much more; however, e.g. on the 25th day that we traded on. The same can also be seen with losses, e.g. on the 13th day we traded on. So the model seems to have approximately learnt how the market moves, but there are still some evident inconsistencies and uncertainties that lead the model to make considerable losses.

# Conclusion

To conclude, we have included possible political insecurities as well as those posed by the coronavirus. Our model analyses these factors to see whether it is safe to invest or not and how to invest the money which answers what we are looking for. At the end of this project, I can conclusively say that I underestimated this undertaking's difficulty. The module requires much more computing power and brainpower to reach consistently positive results due to the uncountable amount of parameters that affect the market. However, we were still able to get satisfactory results due to an optimistic economy and the tech and pharmaceutical bubble[[14]](#footnote-13) expanding during our trading period. Despite the positive results, our model will not be usable in the future once the coronavirus is not as much of a significant threat to the global economy. However, the same organisation can be kept: risk assessment and stock classification with different ways to assess risk.

Nevertheless, this project was the toughest one I have ever undertaken. I entered it without much background knowledge which forced me to study finance, economics, etc. behind the stock market. The research allowed me to expand my knowledge of various previously esoteric concepts and delve deeper into the AI world. Furthermore, it took several failures and iterations to finally come to my final module due to several disturbing factors such as the Black Lives Matter riots[[15]](#footnote-14) and the COVID pandemic wreaking havoc in America and turning the trading environment very fragile. All the skills I acquired through this process will significantly aid me in my future endeavours in both university and my future professional career.

# 

# 

# 

# Bibliography

Ian B., FitchRatings, *Deep Global Recession in 2020 as Coronavirus Crisis Escalates,* {Online}, <https://www.fitchratings.com/research/sovereigns/deep-global-recession-in-2020-as-coronavirus-crisis-escalates-02-04-2020>

Seungho B., Sunil M. and Mina G., *COVID-19 and stock market: An industry level analysis*, {Online}, (last visit: 5 january 2021), <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7467874/#:~:text=The%20recent%20Coronavirus%20>(COVID%2D19,cent%20on%2016th%20March%202020.

Vitaly B., Towards Data Science, *Adam - latest trends in deep learning optimization,* {Online}, (last visit: 5 january 2020), <https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c?gi=1d0fd241d146>

Dhruvil K., Towards Data Science, *Introduction to Word Embedding and Word2Vec,* {Online}, (last visited: 5 january 2021), <https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa>

Ravindra K., Towards Data Science, *Using LSTMs to forecast time-series,* {Online}, (last visit: 5 january 2020) [Using LSTMs to forecast time-series - Towards Data Sciencetowardsdatascience.com › using-lstms-to-forecast-time-series-4ab6883…](https://towardsdatascience.com/using-lstms-to-forecast-time-series-4ab688386b1f)

Samer M., Simon A. and Michael B., *The Economic Impact of political Instability and Mass Civil Protest,* {Online}, (last visited: 5 january 2021) <https://www.nottingham.ac.uk/credit/documents/papers/2017/17-03.pdf>

Max N., Quartz, *Forget the tech bubble. It’s the biotech bubble you should worry about,* {Online}, (last visited: 5 january 2021), <https://qz.com/324939/biotech-valuation-bubble/>

Jeremy O., MarketWatch, *What investors can learn from how investors read the news,* {Online}, (last visited: 5 january 2021), <https://www.marketwatch.com/story/what-investors-can-learn-from-how-they-read-the-news-2016-03-09>

Matt P., The New York Times, *Trump Volatility in Markets is Back,* {Online}, (last visited: 5 january 2021) <https://www.nytimes.com/2019/12/03/business/stocks-fall-trump-china-trade-deal.html>

PITTS Walter and MCCULLOGH Warren, *A Logical Calculus of the Ideas Immanent in Nervous Activity*, {Online}, (last visited: 5 january 2021), <https://web.csulb.edu/~cwallis/382/readings/482/mccolloch.logical.calculus.ideas.1943.pdf>

Jill SS., healthline, *COVID-19 vs. SARS: How Do They Differ?,* {Online}, (last visit: 5 january 2021), <https://www.healthline.com/health/coronavirus-vs-sars#symptoms>

Antony S. and John M., *News Sentiment Analysis,* {Online}, (last visited: 5 january 2021), <https://arxiv.org/pdf/2007.02238.pdf>

Erika S., Los Angeles Times, *2020 was the year America embrace Black Lives Matter as a movement not just a moment,* {Online}, (last visited 5 january 2021), <https://www.latimes.com/california/story/2020-12-16/black-lives-matter-protests-george-floyd-coronavirus-covid-2020>

Jim T., The New York Times, *A $2 Trillion Lifeline Will Help, but More May Be Needed*, {Online}, (last visit: 5 january 2021), <https://www.nytimes.com/2020/03/25/business/2-trillion-stimulus-coronavirus-bill.html>

Unknown, UNCTAD, *Trade war leaves both US and China worse off,* {Online}, (last visited: 5 january 2021) <https://unctad.org/news/trade-war-leaves-both-us-and-china-worse#:~:text=US%20tariffs%20on%20China%20have,or%20captured%20by%20US%20producers>.

Unknown, Investopedia, *7 Technical Indicators to Build a Trading Toolkit,* {Online}, (last visited: 5 january 2021), <https://www.investopedia.com/top-7-technical-analysis-tools-4773275#:~:text=Technical%20indicators%20are%20used%20by,a%20price%20move%20will%20continue>.

Unknown, *MarketMedia, Banks to Invest $300 Billion in AI by 2030*, {Online}, (last visited: 5 january 2021), <https://www.marketsmedia.com/banks-to-invest-300-billion-in-ai-by-2030>

**Finance(Sites)**

* <https://en.wikipedia.org/wiki/Moving_average#Simple_moving_average>
* <https://www.investopedia.com/terms/e/ema.asp>
* <https://www.investopedia.com/terms/m/macd.asp>
* <http://www.traderpedia.it/wiki/index.php/Bollinger_bands#:~:text=La%20costruzione%20delle%20Bollinger%20bands,la%20std%20a%2020%20gg.>
* <https://www.investopedia.com/terms/r/rsi.asp>
* <https://www.investopedia.com/articles/technical/02/010702.asp>
* <https://en.wikipedia.org/wiki/Algorithmic_trading>
* <https://www.investopedia.com/articles/active-trading/101014/basics-algorithmic-trading-concepts-and-examples.asp>

**Neural Networks(Sites)**

* <https://en.wikipedia.org/wiki/Long_short-term_memory>
* <https://en.wikipedia.org/wiki/Artificial_neural_network#Learning>
* <https://towardsdatascience.com/understanding-different-loss-functions-for-neural-networks-dd1ed0274718>
* <https://en.wikipedia.org/wiki/Cross_entropy>
* <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>
* <https://stackoverflow.com/questions/34518656/how-to-interpret-loss-and-accuracy-for-a-machine-learning-model>
* <https://en.wikipedia.org/wiki/Activation_function>
* <https://d4datascience.wordpress.com/2016/09/29/fbf/>
* <https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history2.html>

# 

# 

# Annexes

## Background

Professionals have been trying to crack the financial markets for more than a century with no real or constant success. This process, albeit not completed yet, has taken much brain power and time. Financial analysts usually spend half a decade studying finance to acquire the knowledge in university to be able to judge financial markets correctly or at least understand them a bit more. The traders made predictions to try to get as much profit as possible based on data like news, stock prices, public sentiment, etc. Over these years and thanks to machines this process of data collection has become easier due to the ubiquity of information on the internet.

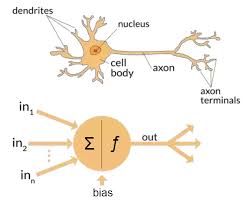
Moreover, stock purchases were then made via phone calls to holders which would then sell the stock for the market price. Thanks to the internet revolution this process has disappeared and instead computers are being used to make every transaction.

Nevertheless, these aren’t the only way computers can be helpful in this process. In the mid 20th century, a new area of study was added to computer science: machine learning, the theory and development of computer systems able to perform tasks typically requiring human intelligence. This field wasn’t much regarded at the time but thanks to a snowball effect it started slowly increasing in importance. More recently, significant leaps were taken in this field which allowed artificial intelligence to exceed human ability, for example in 2016 Google’s AI DeepMind system beat the Go world champion, Ke Jie, in a round of 5 or even beat the rubix cube world record by solving it in 0.38 seconds compared to 4.69 seconds scored by a human. This leads us to our main subject of interest with models called neural networks. One of the most considerable leaps in machine learning is the development of artificial neural networks which are going to be the architecture that we will be using in this paper. ANNs vaguely try to copy the functioning of the human brain(neurons). ANN’s grasp can exceed purely scientific fields and start reaching and dominate entirely dissimilar fields like finance. The opposition underlined is essential as machine learning is regarded as mathematical analysis of situations whereas financial markets are often seen as a game of chance. So by implementing a machine learning model to the stock market, we extract the mathematical and statistical part of the markets to give us reasonable predictions. This method, of course, seems appealing, and it is seen as the inevitable future for finance with large institutions investments in AI “projecting to reach $300 billion by 2030”[[16]](#footnote-15) .

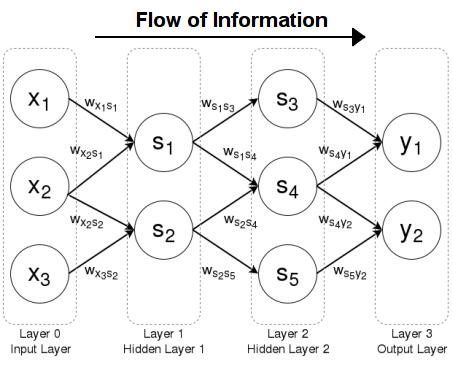
## Neural Network

The first publication that alluded to neural networks was written in 1943 by Warren McCulloch, neurologist, and Walter Pitts, mathematician, in which they attempted to explain how neurons in the brain work and using that logic they were able to model the first simple neural network with an electrical circuit (A Logical Calculus of the Ideas Immanent in Nervous Activity[[17]](#footnote-16)). After seemingly having struck gold in the study of computer science, progress was slow. There were a couple of leaps in the sector, e.g. Stanford making echo-cancelling models in the 1960s. However, soon after came a cold winter for neural networks during which many unfulfilled promises had been made. The future seemed bleak. Nevertheless, a US-JAPAN conference on neural networks in 1982 set a fire under American researchers due to all of the significant advancements made by the Japanese researchers. Thanks to this spike of interest as well as the advancement in computer technology and most importantly, the accessibility of large amounts of data led to the creation of multiple-layered and backpropagation networks in 1975. This new architecture is the basic artificial neural network.

Modelled similarly to a brain, a neural network consists of nodes(neurons), links(axons) which lead to a specific output/result. More concisely, the network analyses training data which it fits throughout its layers of neurons(all densely interconnected). Each node will give a specific number or weight to each of its incoming links. For every different piece of data, a different weight will be assigned. The data first enters through an input layer which will then feed it through hidden layers. The input layer takes in a specific amount of information of features that are important to the final result. After having been flown throughout the main body of the network, the values that the neurons will return from the hidden layer will be sent to the output layer which will determine the probability of a data item being of a specific class. Depending on this probability, it will be able to classify our data.

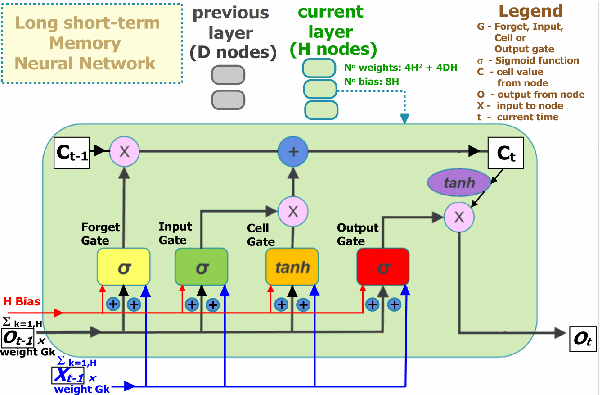


There can be various types of neurons in a neural network, the most common being the Dense cell which returns a weighted sum of the output of the previous cell which gives us a specific value using a nonlinear activation function to keep the returning value between 0 and 1. The nonlinearity of the activation functions allows the neural network to successfully predict the class of a function which is divided by a nonlinear decision boundary. The value returned from a neuron’s activation function is valuable to us as it helps us determine whether a feature of the data is essential to the decision making. If the value is over a specific threshold value, the node fires/sends the number to other outgoing connections; otherwise, the node is not activated and will not pass any data. Initially, all the weights and thresholds are set randomly but are altered during training to fit the data best until predicted results start becoming closer and closer to expected ones. This process is best known as “feed-forward” as the data only moves in one direction.

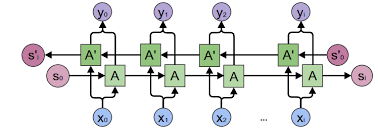


However, these conventional neural networks aren’t very effective on sequential data. They are indifferent to context or what has previously been done to predict or classify objects. This is problematic as all the data we will be handling relies on what happened the previous days. Therefore, we must use recurrent neural networks to address this issue which are simply looping networks that allow information to persist. Each cell within relies on the state of the previous one for its output forming a chain of cells. More specifically we shall be using Long Short Term Memory networks (LSTM) which are basically recurrent networks that allow the network to retain long-term dependencies.

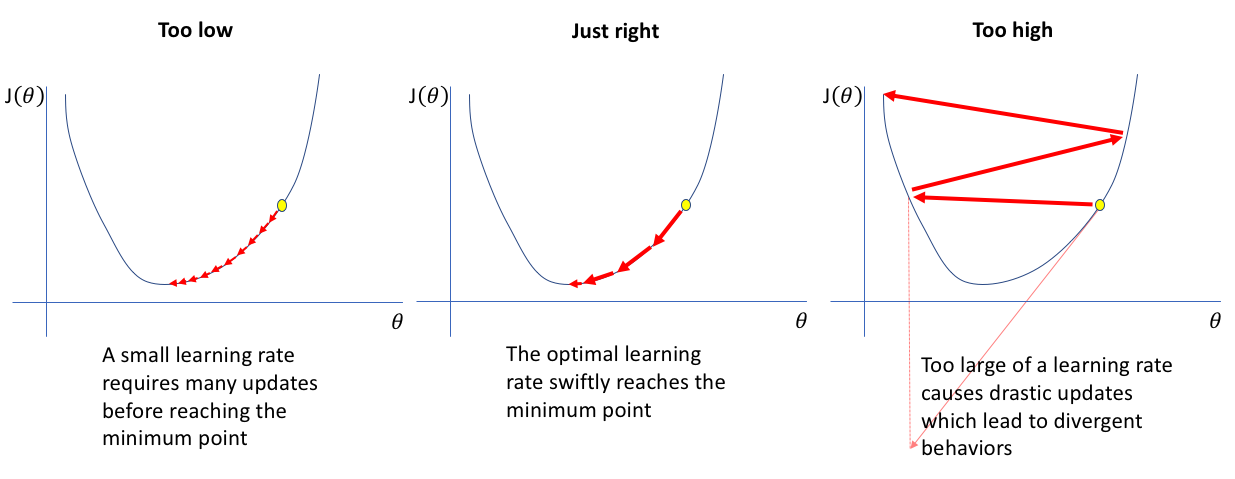
Being a revolutionary discovery for recurrent networks, LSTMs have a very complicated anatomy. The easiest way to explain it is that a conveyor belt (cell state) carries the data through the cell passing it through gates which will evaluate the importance of parts of the data and tweak it to get a valuable output for the next cell. There are, all in all, 3 main gates: forget, input and output gates. The entrance of the first gate or forget gate looks at the previous and current state and decides what information we should keep or throw away. Next, this selection we move on to the input gate which decides what values we should update. After this classification, the tanh layer creates a new vector of novel candidates that will be added to the old state creating the newly updated cell state. Finally, once we get to the output gate, we will decide what our output should be and hand over a filtered version of our cell state or what we find relevant to pass on for our prediction.



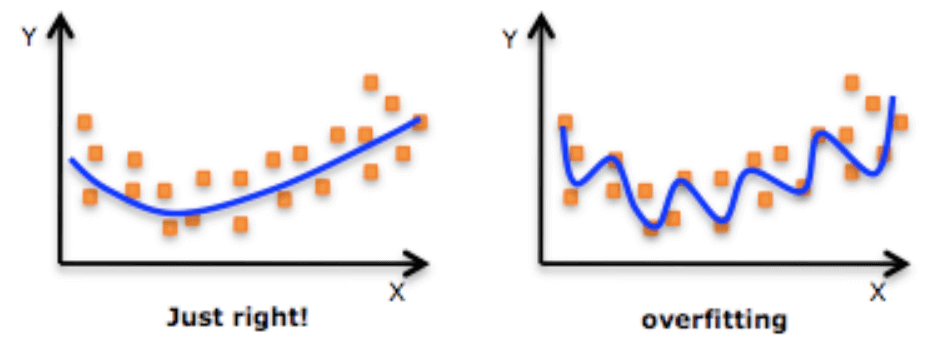
What we’ve presented so far is a vanilla LSTM, however there is a variant of the LSTM which can be helpful when the past as well as the future context of the data is important. We shall be looking at Bidirectional LSTM recurrent neural networks. Instead of just having a forward recurring component, it has a backwards one as well. After having passed through the forward feed(positive time direction), the information then passes through the backwards recurring component(negative time direction). Therefore, the networks are trained to predict the positive and negative directions of time simultaneously. By using both directions of time, the data from the past and future can be used to calculate the same output. This varies from the standard LSTM RNN as we added an extra layer that includes the future information.



Now that we cleared up the forward feed of the model we must go over an essential part of neural network training which is the backpropagation. After the forward processes are done our network evaluates itself. To do this we use a loss function which calculates the error marge between the predicted and the expected result. Loss functions differ in use and output. Thus are important to consider when building a model. The calculated error depends on the network’s internal parameters such as weights and biases. Therefore we must find a way to minimise the error by optimizing these parameters. We achieve this by propagating the error backwards to the previous layer where we will optimize the parameters using an optimizer. There are many different optimizers that are used and differ in efficiency depending on the problem. Yet optimizers will not always find the minimum of the loss function depending on how we set its hyperparameters. The most significant one is the learning rate as it designs how big of a step it should take towards a minimum of the loss function. So if the step is too big it shall never converge and if the step size is too small it will take more and more time to reach the minimum.



The optimizer might not instantly find the global minimum for the error after having iterated through the whole training data. To clear this up we repeat the process until we are satisfied with the results. These repetitions are called epochs. Another problem we might find is that our model might overfit on our training data which in general terms means that the neural network has memorised our data and won’t be applicable to real life problems. A way to avoid this is to add a dropout. A dropout refers to a probability of certain neurons on different layers not being used. This prevents neurons from creating dependencies on each other and encourages the individual power of a neuron. To check if there is any overfitting it is always important to have validation data which the model tests itself on after each epoch to see where it is as well as test data to holistically evaluate the model at the end of training which will not only give us an idea of the loss but also the accuracy(or probability of getting the prediction) of the model.



Nevertheless, the idea of a neural that we have now is incomplete. Therefore, we must introduce the concept of batch size which will help use not only alleviate the computational work but also improve our model. The batch size is a hyperparameter which defines the number of samples that the network should work through before updating its internal parameters. The higher the batch size the greater the computational power required and vice versa. However, a greater batch size (to a certain limit) will help our model generalise more to give us a more promising and applicable model.

1. Seungho B., Sunil M. and Mina G., *COVID-19 and stock market: An industry level analysis*, {Online}, (last visit: 5 january 2021), <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7467874/#:~:text=The%20recent%20Coronavirus%20>(COVID%2D19,cent%20on%2016th%20March%202020. [↑](#footnote-ref-0)
2. Jim T., The New York Times, *A $2 Trillion Lifeline Will Help, but More May Be Needed*, {Online}, (last visit: 5 january 2021), <https://www.nytimes.com/2020/03/25/business/2-trillion-stimulus-coronavirus-bill.html> [↑](#footnote-ref-1)
3. Ian B., FitchRatings, *Deep Global Recession in 2020 as Coronavirus Crisis Escalates,* {Online}, <https://www.fitchratings.com/research/sovereigns/deep-global-recession-in-2020-as-coronavirus-crisis-escalates-02-04-2020> [↑](#footnote-ref-2)
4. Jill SS., healthline, *COVID-19 vs. SARS: How Do They Differ?,* {Online}, (last visit: 5 january 2021), <https://www.healthline.com/health/coronavirus-vs-sars#symptoms> [↑](#footnote-ref-3)
5. Ravindra K., Towards Data Science, *Using LSTMs to forecast time-series,* {Online}, (last visit: 5 january 2020) [Using LSTMs to forecast time-series - Towards Data Sciencetowardsdatascience.com › using-lstms-to-forecast-time-series-4ab6883…](https://towardsdatascience.com/using-lstms-to-forecast-time-series-4ab688386b1f) [↑](#footnote-ref-4)
6. Vitaly B., Towards Data Science, *Adam - latest trends in deep learning optimization,* {Online}, (last visit: 5 january 2020), <https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c?gi=1d0fd241d146> [↑](#footnote-ref-5)
7. Matt P., The New York Times, *Trump Volatility in Markets is Back,* {Online}, (last visited: 5 january 2021) <https://www.nytimes.com/2019/12/03/business/stocks-fall-trump-china-trade-deal.html> [↑](#footnote-ref-6)
8. Unknown, UNCTAD, *Trade war leaves both US and China worse off,* {Online}, (last visited: 5 january 2021) <https://unctad.org/news/trade-war-leaves-both-us-and-china-worse#:~:text=US%20tariffs%20on%20China%20have,or%20captured%20by%20US%20producers>. [↑](#footnote-ref-7)
9. Samer M., Simon A. and Michael B., *The Economic Impact of political Instability and Mass Civil Protest,* {Online}, (last visited: 5 january 2021) <https://www.nottingham.ac.uk/credit/documents/papers/2017/17-03.pdf> [↑](#footnote-ref-8)
10. Antony S. and John M., *News Sentiment Analysis,* {Online}, (last visited: 5 january 2021), <https://arxiv.org/pdf/2007.02238.pdf> [↑](#footnote-ref-9)
11. Dhruvil K., Towards Data Science, *Introduction to Word Embedding and Word2Vec,* {Online}, (last visited: 5 january 2021), <https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa> [↑](#footnote-ref-10)
12. Unknown, Investopedia, *7 Technical Indicators to Build a Trading Toolkit,* {Online}, (last visited: 5 january 2021), <https://www.investopedia.com/top-7-technical-analysis-tools-4773275#:~:text=Technical%20indicators%20are%20used%20by,a%20price%20move%20will%20continue>. [↑](#footnote-ref-11)
13. Jeremy O., MarketWatch, *What investors can learn from how investors read the news,* {Online}, (last visited: 5 january 2021), <https://www.marketwatch.com/story/what-investors-can-learn-from-how-they-read-the-news-2016-03-09> [↑](#footnote-ref-12)
14. Max N., Quartz, *Forget the tech bubble. It’s the biotech bubble you should worry about,* {Online}, (last visited: 5 january 2021), <https://qz.com/324939/biotech-valuation-bubble/> [↑](#footnote-ref-13)
15. Erika S., Los Angeles Times, *2020 was the year America embrace Black Lives Matter as a movement not just a moment,* {Online}, (last visited 5 january 2021), <https://www.latimes.com/california/story/2020-12-16/black-lives-matter-protests-george-floyd-coronavirus-covid-2020> [↑](#footnote-ref-14)
16. Unknown, *MarketMedia, Banks to Invest $300 Billion in AI by 2030*, {Online}, (last visited: 5 january 2021), <https://www.marketsmedia.com/banks-to-invest-300-billion-in-ai-by-2030> [↑](#footnote-ref-15)
17. Walter Pitts and McCulloch Warren, *A Logical Calculus of the Ideas Immanent in Nervous Activity*, {Online}, (last visited: 5 january 2021), <https://web.csulb.edu/~cwallis/382/readings/482/mccolloch.logical.calculus.ideas.1943.pdf> [↑](#footnote-ref-16)