Stock Price Forecaster Web Application In Python

Dean McGowan

G00369547

B.Sc.(Hons) in Software Development

MARCH 15, 2022

Final Year Project

Advised by: Joseph Corr

Department of Computer Science and Applied Physics
Galway-Mayo Institute of Technology (GMIT)

Contents

About this project

Abstract This project is a stock price forecaster. It is a web application designed using Streamlit which is an open-source python framework specifically for creating web apps using data science, machine learning etc. The web app allows the user to select a stock from a list and select the number of years the price should be forecast. The current state of the stock is displayed in a table and the recent history is displayed in a graph. The app will then use machine learning to predict the closing price of the chosen stock in n number of years chosen by the user. The predicted data will then be presented in a manner similar to the current data with a table and graph indication the predicted price with a legend for the users ease of reading.

Authors There is a single author for this project, myself, Dean McGowan.

Chapter 1

Introduction

The stock market is a platform where people can buy and sell shares of companies to gain ownership in a company, diversify assets, or make a profit. Currently 55% of American's invest in the stock market [1]. In recent years the ease of access has improved significantly with the release or trading brokerage apps such as Robinhood developed by Baiju Bhatt. Traders can purchase stocks of companies they believe will be successful and grow. If they are correct the profits are only limited by how much they initially invest. Of course the same goes for losses also. The pandemic saw a tremendous fall and subsequent mighty rise in the stock market that shook traders globally. It was this series of unexpected market crash and rise that made me interested in the question 'Is it possible to predict the market's behaviour?'.

Artificial intelligence is simply the discipline of attempting to bring human intelligence to machines through of use of pre-defined algorithms.

Machine learning is one of the fastest growing sectors in the technology word right now. Between 2016 and 2022 the overall machine learning market has grown from $ 1 billion in 2016 to $ 8.81 billion in 2022 [2]. Machine learning is a field of artificial intelligence that focuses on using data and algorithms to mimic the way humans use their brains. This allows the machine to learn without specifically programming the rules it must follow. The models become more accurate over time as they 'learn' gradually. Machine learning typically requires vast amounts of data and pre-processing. There are a few different types of machine learning like supervised learning in

which all data given to the model is labelled, unsupervised learning where none of the inputs are labelled and the model must decide upon its own classifications or rules and reinforcement learning where the model is 'rewarded' when correct as to steer it in the right direction.

This project utilises a type of neural network called a 'recurrent neural network' or an 'RNN'. This relates to the way human intelligence learns and understands. Humans do not look at a sentence and take in each word as an individual item, they take in each word remembering the word before it to put it into context. The network has a persistence to it.
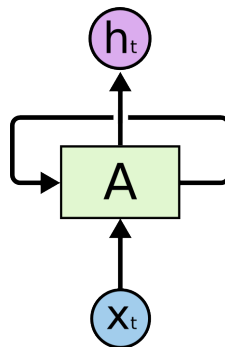
The figure above shows a neural network, labelled 'A'. Label 'Xt' is the input and label 'ht' is the output. As you can see the input is processed by the neural network and then an output is given, while simultaneously the output information is fed back into the neural network.
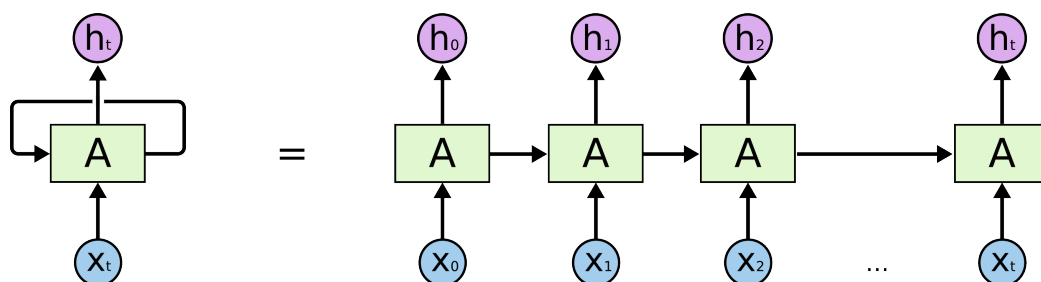

The figure above shows an RNN expanded out to visualise what is happening. It depicts that each neural network is passing on data to it's successor. You may note the visual similarity to a linked

list. This is why RNNs are commonly used with sequential data. An RNN was the most effective choice for my work as the stock data I am working with is sequential.

However, I did not use a standard RNN for my work. I used a very specific type of RNN called a 'Long Short Term Memory' network. To understand why I used this, I will describe the shortcomings of an RNN. An RNN is very efficient in using recent past data to learn and help process the current data. But a problem arises when they have to use data rather far back in the queue to process the current data. As an example, finishing the sentence "I went to the zoo and I saw a lot of...". An RNN will struggle to pull context from multiple words back to complete the sentence. Obviously the answer is 'animals' or a specific species. But this is only obvious to us and not the RNN. For my work I needed the network to recall more than just the past few days of stock data as they may not represent the true current performance, the stock may just have dipped for example, something frequent and completely normal for a stock to do.

A 'Long Short Term Memory' network or an 'LSTM' is designed specifically to fix the long-term dependency problem present in a standard RNN. The LSTM design was first introduced in 1997 by two computer scientists Hochreiter and Schmidhuber. This type of network was created to remember data for long periods of time as it's natural behaviour.
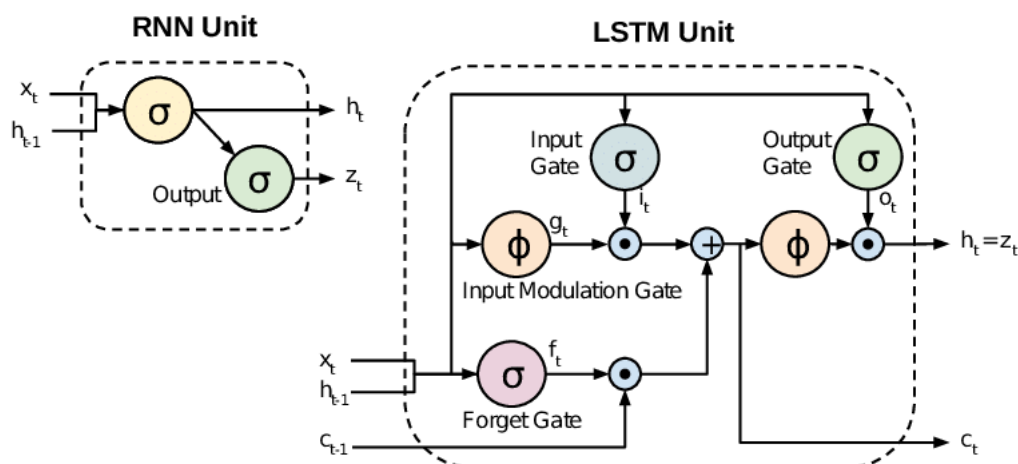
All RNNs have the same topology. A chain of repeating neural networks as shown in figure 2. Each of these neural networks are quite simple as they will typically have a single tanh layer. A tanh layer is an activation function used to get the output of a node. In the case of a tanh function it will map the output somewhere between -1 and 1.
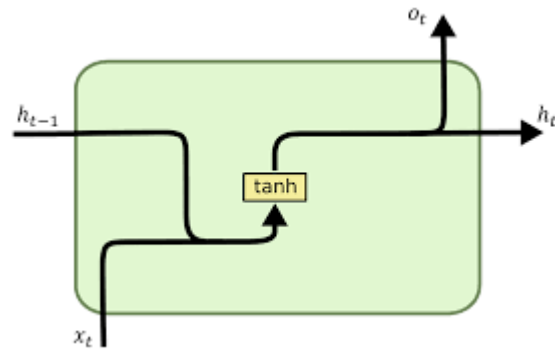
LSTMs have a similar topology to RNNs in that they share the chain-like structure, however, each module of the chain in an LSTM network is more complex than an RNN with each module containing four layers working together.
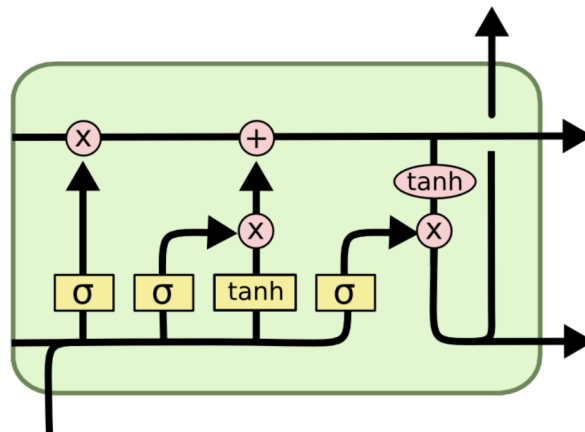
In the above figure, the yellow boxes are neural network layers, the red circles are pointwise operations, and each line carries a vector. Where the lines split into two is copying and where thy come together in concatenation.

Chapter 2

Context

2.0.1   Objectives

The key objectives of this project are as follows:

- Create an easy to use and well designed front-end utilising a new technology for me, Streamlit.

- Learn about machine learning algorithms and techniques to further my knowledge in artificial intelligence.

- Improve upon my project management and time management skills.

- Learn how to build and implement a user authentication system into my web application.

- Sharpen my research skills and thinking critically about the content I find.

2.0.2   Following Chapters

Methodology: Describes the methods in which I went about my project. The development approach, testing, why I chose the technologies used.

Technology Review: I will give a review of each technology I used to create the application. I will discuss the technologies at a conceptual level, their standards, database models etc.

System Design: I will cover the components of the application and how they work together.

System Evaluation:

My GitHub can be found at: https://github.com/DeanMcG/4thYearProject

Chapter 3

Methodology

I chose to to develop this project using an agile approach. I broke down all the tasks needing to be done and wrote them out into a list of deliverables in a chronological order of when they needed to be completed. I then divided these deliverables into separate sprints. Each sprint lasted two weeks. At the end of each sprint I would make sure everything that was supposed to be completed was done and working well. If a deliverable wasn't complete or had bugs, that became my first issue in the following sprint.

I used pytest to test my code. Pytest is a third-party testing framework. I used unit tests in my project. Unit testing allowed me to be sure that all code was of a good standard before deploying it to the project. It also allowed me to find any bugs very early in the process and pinpoint where these bugs were before implementation.

I chose to develop this project using python. I wanted to use python as it is a great language for working with machine learning. It is also a language I have little experience with and I wanted to know more about it as it is used widely in the industry. Being ranked second most in demand in 2021 by Berkely Unversity. [3] About one to two pages. Describe the way you went about your project:

- Agile / incremental and iterative approach to development. Planning, meetings.

- What about validation and testing? Junit or some other framework.

- If team based, did you use GitHub during the development process.
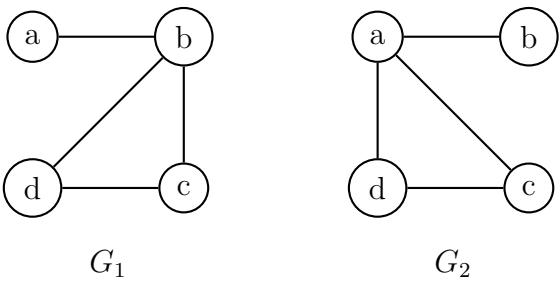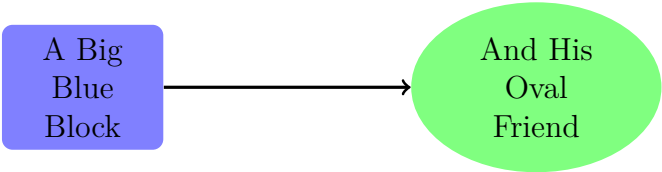
Figure 3.1: Nice pictures

Figure 3.2: Nice pictures

- Selection criteria for algorithms, languages, platforms and technologies.

Check out the nice graphs in Figure 3.2, and the nice diagram in Figure ??.

Chapter 4

Technology Review

About seven to ten pages.

- Describe each of the technologies you used at a conceptual level. Standards, Database Model (e.g. MongoDB, CouchDB), XMl, WSDL, JSON, JAXP.

- Use references (IEEE format, e.g. [1]), Books, Papers, URLs (timestamp) – sources should be authoritative.

4.1 XML

Here's some nicely formatted XML:

```xml
<this>
  <looks lookswhat="good">
    Good
  </looks>
</this>
```

Chapter 5

System Design

As many pages as needed.

- Architecture, UML etc. An overview of the different components of the system. Diagrams etc... Screen shots etc.

Column 1	Column 2
Rows 2.1	Row 2.2

Table 5.1: A table.

Chapter 6

System Evaluation

As many pages as needed.

- Prove that your software is robust. How? Testing etc.

- Use performance benchmarks (space and time) if algorithmic.

- Measure the outcomes / outputs of your system / software against the objectives from the Introduction.

- Highlight any limitations or opportuni-ties in your approach or tech-nologies used.

Chapter 7

Conclusion

About three pages.

- Briefly summarise your context and ob-jectives (a few lines).

- Highlight your findings from the evalua-tion section / chapter and any opportuni-ties identified.

Bibliography

[1] F. Norrestad, "Share of adults investing money in the stock market in the united states from 1999 to 2020."

[2] S. Lazzaro, "Machine learning's rise, applications, and challenges."

[3] B. University, "11 most in-demand programming languages in 2021."