الجمهورية الشعبية الديمقراطية الجزائرية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
المدرسة العليا للإعلام الآلي ٠80 ماي 5491٠ بسيدي بلعباس
École Supérieure en Informatique
-08 Mai 1945- Sidi Bel Abbès

THESIS

To obtain the diploma of **Master**
Field: **Computer Science**
Specialty: **Ingénierie des Systèmes Informatiques (ISI)**

Theme

Trading using Reinforcement Learning

Presented by: BENAHMED Djawed

Submission Date: **June, 2022**
In front of the jury composed of:

Mr. CHAIB Souleymane	President
Mr. KHALDI Belkacem	Supervisor
Ms. DIF Nassima	Examiner

Academic Year : 2021/2022

Acknowledgments

God, the Almighty, is to be praised and thanked first and foremost for showering me with blessings during my research to enable me to accomplish it successfully.

Words cannot express how grateful I am to my parents for their love, prayers, care, and sacrifices in educating and preparing me for my future; the level of support we had and continue to receive is indescribable.

I would like to express my deep and sincere gratitude to my research supervisor and teacher, **Mr. KHALDI Belkacem** for providing valuable guidance and advice throughout this research.

I cannot thank **Mr. ALAOUI MDAGHRI Abdellah** of Swissdigilab enough for providing me the opportunity to work on this project, from which I have acquired a huge amount of fresh new information. I also would like to thank him for his guidance and continuous support.

A special thanks to the entire working team of my school, including the School director **Dr. Benslimane Sidi Mohammed** and **Dr. Amar Bensaber Djamel**, as well as all teachers, without whom I could not have accomplished my goals. Finally, a big thank you to our friends and colleagues for their assistance along this trip.

BENAHMED Djawed.

Abstract

Artificial intelligence has been increasingly popular in financial sectors such as stock and currency trading in recent years. Reinforcement learning is one of AI widely used branches in financial market problems.

In this thesis, we begin with a quick overview of DL, RL, and deep RL methods in various economic applications. Furthermore, providing an in-depth insight into the problem of ”automated trading” in different markets and exploring different approaches covered in the literature.

Keywords

Stock Market, Forex Market, Crypto Market, Asset trading, Automated Trading, Portfolio management, Reinforcement Learning, Deep Learning, Deep Q-Network.

Résumé

L'intelligence artificielle est de plus en plus populaire dans les secteurs financiers tels que le commerce des actions et des devises ces dernières années. L'apprentissage par renforcement est l'une des branches de l'IA largement utilisées dans les problèmes de marché financier.

Dans cette thèse, nous commençons par un aperçu rapide des méthodes DL, RL et RL profondes dans diverses applications économiques. En outre, fournir un aperçu approfondi du problème du "trading automatisé" sur différents marchés et explorer différentes approches couvertes par la littérature.

Keywords

Marché boursier, marché Forex, marché de la cryptographie, trading d'actifs, trading automatisé, gestion de portefeuille, apprentissage par renforcement, apprentissage en profondeur, Deep Q-Network.

الملخص

أصبح الذكاء الاصطناعي شائعًا بشكل متزايد في القطاعات المالية مثل تداول الأسهم والعملات في السنوات الأخيرة. التعلم المعزز هو أحد فروع الذكاء الاصطناعي المستخدمة على نطاق واسع في مشاكل السوق المالية.

في هذه الأطروحة ، نبدأ بنظرة عامة سريعة على أساليب DL و RL و RL العميقة في مختلف التطبيقات الاقتصادية. علاوة على ذلك ، توفير نظرة متعمقة حول مشكلة "التداول الآلي" في الأسواق المختلفة واستكشاف الأساليب المختلفة التي تم تناولها في الأدبيات.

Keywords

سوق الأسهم ، سوق الفوركس ، سوق العملات المشفرة ، تداول الأصول ، التداول الآلي ، إدارة المحافظ ، التعلم المعزز ، التعلم العميق ، شبكة Q-Network العميقة.

Contents

List of Figures

List of Tables

List of Acronymes

A2C Advantage Actor Critic. 59

ADX Average Directional Index. 18

AI Artificial intelligence. 12, 13, 23, 24, 27

BH Buy and Hold. 7, 46, 48, 49

BPTT Backpropagation Through Time. 57

CCI Commodity Channel Index. 18

CNN Convolutional Neural Network. 37, 61

DDPG Deep Deterministic Policy Gradient. 35, 61

DNN Deep Neural Network. 17

DPG Deterministic Policy Gradient. 34, 52

DQN Deep Q-Network. 7, 34, 48, 49, 51

DRL Deep Reinforcement Learning. 13, 33, 48, 59, 60

DRQN Deep Recurrent Q-network. 7, 48–50

EMA Exponential Moving Average. 51

GDPG Gated Deterministic Policy Gradient. 59, 60

GDQN Gated Deep Q-network. 52, 59, 60

GRU Gated Recurrent Units. 51, 59

LSTM Long Short-Term Memory. 37, 48, 49, 61

MA Moving Average. 51, 60

MACD Moving Average Convergence Divergence. 7, 16–18, 51

OBV On-Balance Volume. 18, 51

OHLCV Open, Hight, Low, Close, Volume. 17, 42, 50, 51, 59

PG Policy Gradient. 34, 59

PPO Proximal Policy Optimization. 35

RCNN Recurrent Convolutional Neural Network. 60, 61

RL Reinforcement Learning. 13, 26, 57, 61

RNN Recurrent Neural Network. 37, 60, 61

RSI Relative Strength Index. 17

SR Sortino Ratio. 51

VaR Value at Risk. 44

General Introduction

0.1 Introduction and motivation

Artificial Intelligence, like many other technical achievements, emerged from the pages of fairy tales and science fiction stories. People fantasized about machines that might fix issues, speak and make decisions. Fast forward to the 21 century, Artificial intelligence has become part of our daily lives, from phone voice assistants to smart homes and self-driving cars, Artificial intelligence (AI) solutions are all over the place.

The financial industry is one of the first adopters of artificial intelligence, from banking fraud detection to automated investment and trading. Anthony Antenucci, vice president of global business development at Intelenet Global Services, recently said “Machine learning is evolving at an even quicker pace, and financial institutions are one of the first adaptors,”

Financial markets, such as stocks, fiat money, cryptocurrencies, and even physical assets such as gold and silver, are notoriously complicated and chaotic, with highly intricate systems that often make prediction very difficult for a human trader. A quote by William Gallaher ”When you think the market can’t possibly go any higher (or lower), it almost invariably will, and whenever you think the market “must” go in one direction, nine times out of ten it will go in the opposite direction. Be forever skeptical of thinking that you know what the market is going to do”. The complexity of the financial markets created a need for a sophisticated intelligent solution.

Stiff trading strategies designed by experts in the field are one of the intelligent solutions, but these strategies often fail to achieve profitable returns in all market conditions due to their incapability of adapting to new trends and changes in the markets [60]. Hence, Machine learning methods such as RL/DRL are used to address these problems.

Thanks to artificial intelligence, trading nowadays is brought to a whole new level – more professional and advanced strategies are applied easily and comfortably even by beginners. "Artificial intelligence is to trading, what fire was to the cavemen." That's how one industry player described the impact of disruptive technology on a staid industry. In other (less creative) words, AI is a game-changer for the financial markets [75].

While humans remain an important element of the trading equation, AI is becoming increasingly important, and human interventions becoming less and less needed, to the point where the whole trading process can be automated. Which is the goal of this thesis, exploring different approaches and methods used in the literature to automate the process of trading using AI Reinforcement Learning.

0.2 Plan of the thesis

- **Chapter 2: Background**

  In this chapter, we will present the necessary material to enable the reader to comprehend certain basic topics that will be explored throughout this thesis. such as Trading, Stock market indicators, and AI (Machine learning, Reinforcement learning, ..)

- **Chapter 3: Deep reinforcement learning and trading**

  In this chapter, we will discuss different problems in trading tackled with reinforcement learning such as prediction problems, portfolio management, and automated trading.

- **Chapter 4: Automated trading with deep reinforcement learning**

  In this chapter, we will go deeper into the topic of automated trading, examining and comparing various techniques and methodologies used in the literature

- **Conclusion**

  Conclusion of the overall study of trading using deep reinforcement learning

CHAPTER 1

Background

In this chapter, we provide the required information to allow the reader to grasp certain basic concepts that will be discussed throughout this thesis

1.1 Trading

Trading is the process of purchasing and selling different assets or financial instruments (such as stocks, currencies, commodities, companies, etc) that are listed on the market. The idea is to maximize the capital's return on investment by using recurring buy and sell orders to take advantage of market volatility. Profit is generated when one buys at a lower price than it sells afterward. Trading is a short-term and volatile activity as compared to long-term transactions such as mutual fund or bond deals. In a nutshell, the fundamental rule of trading is to purchase when the price is low and sell when the price is high to make profits **??**.

1.1.1 Trading VS Investing

There is often a misunderstanding between the concepts of investing and trading, and this must be clarified. When it comes to investing, an investor is someone who stays on to their position or security for a longer amount of time and is a long-term participant, while a trader is someone who is impacted by the ups and downs of the market's securities. Investing is a long-term method in which the objective is to accumulate wealth gradually over time via the use of investment schemes such as mutual funds, and the purchase and sale of a portfolio of stocks, bonds, and other securities. When compared to trading, investment is kept for years or even decades, and it comes with a variety of benefits such as interest, dividends, stock splits, and many more [55]. The difference between trading and investing is summarized in the table 1.1.

Investing	**Trading**
Long period (Years, decades, or even longer periods)	Short period (Depending on the type of trading, it might be for a week, a day, or hours)
Art of creating wealth (accumulate wealth gradually over time)	Skill of timing the market (buying when the price is lowest and selling when the price is the highest)
Comparatively lower risk and lower returns in the short run but might deliver higher returns in the long run	Involves a high risk-reward ratio (since the price might swing either way, high or low, in a short period)
Profit from stability and long-term predictions of the markets	Profits from volatile trends in the market

Table 1.1: Difference between Trading and Investing

1.1.2 Types of trading

There are different types of trading that we are going to mention below [1]

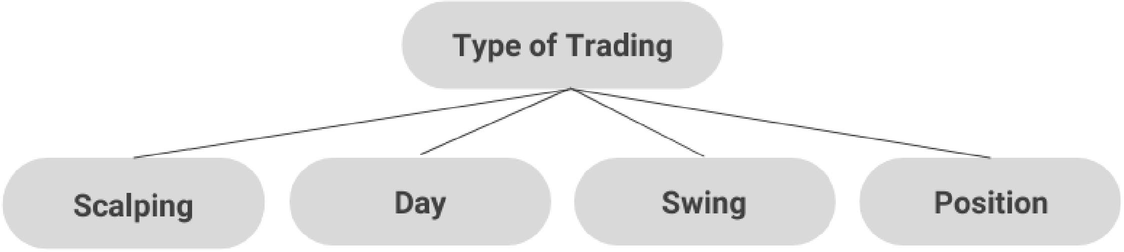

Figure 1.1: Types of trading

1.1.2.1 Scalping

Scalping is the most short-term form of trading. Scalp traders only hold positions open for seconds or minutes at most. These short-lived trades target

[1] https://www.capitalindex.com/bs/eng/pages/trading-guides/different-types-of-trading-strategies

small intraday price movements. The purpose is to make lots of quick trades with smaller profit gains, but let profits accumulate throughout the day due to the sheer number of trades being executed in each trading session [13].

1.1.2.2 Day trading

Day traders take positions and exit them on the same day, eliminating the risk of big overnight movements. All trades will be closed with a profit or a loss at the end of the day. Because trades are typically held for minutes or hours, there is enough time to evaluate the markets and monitor positions periodically throughout the day [13].

Day traders pay particularly close attention to fundamental and technical analysis, using technical indicators such as Moving Average Convergence Divergence (MACD), the Relative Strength Index, and the Stochastic Oscillator, to help identify trends and market conditions.

1.1.2.3 Swing trading

Swing traders usually hold positions for several days, sometimes even weeks. Traders do not need to sit constantly monitoring the charts and their trades throughout the day because positions are held over some time [13].

1.1.2.4 Position trading

Position traders are interested in long-term price movement, hoping to benefit as much as possible from large price shifts. As a result, trades typically last for weeks, months, or even years. Position traders typically analyze and evaluate markets using weekly and monthly price charts, using technical indicators and fundamental analysis to find suitable entry and exit levels [13].

Scalping	Day	Swing	Position
Make multiple trades per hour	Make multiple trades per day	Make several trades per week	Make fewer trades per months
Positions last from seconds to minutes	Positions last from hours to days	Positions last from days to weeks	Positions last from weeks to months
Full-time job	Full-time job	Part-time job	Part-time job
Uses short-term buy and sell signals	Uses short-term buy and sell signals	Utilizes trends and momentum indicators	Utilizes some investing strategies
Smaller relatively stable gains	Multiple, smaller gains or losses	Fewer, but more substantial gains or losses	Fewer, long-term gains

Table 1.2: Comparison between diffrent type of trading

1.1.3 Technical analysis and indicators in trading

The trading data is generally represented with a sequence (at regular intervals of time) of open, close, height, low, and volume of the entity traded (stocks, currencies, bonds, etc) and it is called OHLCV representation [80]. Even Deep Neural Network (DNN) find it challenging to grasp this data in this manner due to the high degree of noise and non-stationary nature of the data [60]. Technical indicators were created to describe market behavior and make the patterns easier to interpret. Technical indicators are used by traders to gain insight into the supply and demand of securities and market psychology. Together, these indicators form the basis of technical analysis. Metrics, such as trading volume, provide clues as to whether a price move will continue [73]. The most popular indicators are:

- **Moving Average Convergence Divergence (MACD)** MACD is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price [6].
- **Relative Strength Index (RSI)** The relative strength index (RSI) is a momentum indicator used in technical analysis that measures the magnitude of recent price changes to evaluate overbought or oversold conditions in the price of a stock or other asset [78]

- **Average Directional Index (ADX)** ADX is used to quantify trend strength. ADX calculations are based on a moving average of price range expansion over a given period of time [32]

- **Commodity Channel Index (CCI)** The CCI was originally developed to spot long-term trend changes but has been adapted by traders for use on all markets or timeframes [48].

- **On-Balance Volume (OBV)** OBV is a technical trading momentum indicator that uses volume flow to predict changes in stock price [76]

There are plenty more indicators, and the encyclopedia [21] book fully describes each one.

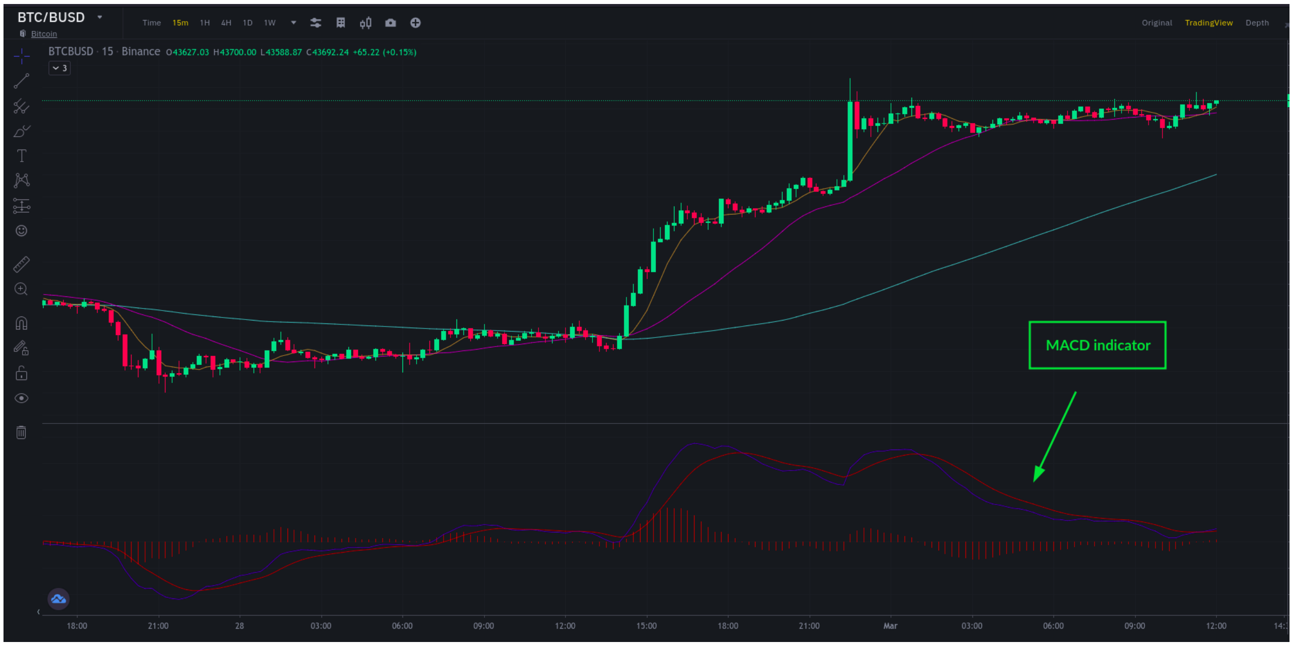

Figure 1.2: MACD indicator in binance platform

1.1.4 Financial markets

Financial markets are a kind of marketplace in which assets such as bonds, shares, and foreign currency are bought and sold (traded). They go under many names, like ”Wall Street” and ”Capital Market” but all refer to the same thing [23]. financial markets are used by:

- **Businesses** to raise capital for expansion and growth.

- **Investors & Traders** to invest or benefit from market instability to expand the capital.

Several financial markets could be used for trading; we will explore a few of them:

1.1.4.1 Crypto currency market

Cryptocurrency refers to a kind of digital or virtual money that is protected by cryptography. This kind of protection makes it impossible to counterfeit or double-spend the cash.

Bitcoin is by far the most common and lucrative kind of virtual money. It was developed by an unknown individual who went by the name Satoshi Nakamoto, and it was presented to the public in the form of a white paper in the year 2008 [54]. Since that time, hundreds of other cryptocurrencies have appeared, such as Ethereum, Litecoin, Solana, and many more.

One of the most distinguishing characteristics of cryptocurrencies is the fact that they are not issued by any central authority. Because of this, cryptocurrencies are considered to be resistant to the influence or meddling of governments or any other parties.

What makes cryptocurrencies possible is the revolutionary blockchain technology which was also introduced in Satoshi Nakamoto's white paper [54]. A blockchain may be thought of as an electronic database distributed, and shared among nodes of computers in a network, in crypto-currencies blockchain securely stores information about transactions.

A cryptocurrency exchange, also known as digital currency exchange (DCE), is an online marketplace that gives consumers the option to acquire, sell, and trade cryptocurrencies or digital currencies for other assets, such as traditional fiat money or other digital currencies.

Trading cryptocurrency with fiat currency such as USD can be expensive (height transaction cost) which led to the creation of stable coins. Stablecoins are digital currencies that are designed to keep their value constant regardless of market conditions. Tether (USDT) and Binance Dollar (BUSD) are two examples of cryptocurrencies that have a fixed value of one dollar. Essentially forming a pair of BTC-USDT (Bitcoin and Tether) or BTC-BSDT (Bitcoin and Binance USD) is effectively the same as making a pair of BTC-USD (Bitcoin USD)

1.1.4.2 Stocks market

Stocks (also called shares or a company's equity), are a financial instrument that shows that you own a piece of a company or business and that you have a percentage of its assets [33]. Stock ownership entails that the shareholder owns a piece of the business in proportion to the number of shares owned. For example, if someone owns 100,000 shares of a company with one million shares, that person or group would own 10% of the company [33].

Why Do Companies Issue Shares? Starting a company (or growing an existing one) needs a large amount of capital to cover the costs of getting the business up and running (expenses such as raw materials, hiring employees, equipment, etc). To get the necessary cash, the business asks for funds from investors, who in turn receive several shares based on the amount invested.

Figure 1.3: Shares holder illustration [14].

The stock market, also known as the stock exchange or bourse in Europe, is the most well-known marketplace. It is a regulated market for the acquisition and selling of securities such as shares and stocks. In simplified terms, the stock market is where shares of publicly traded companies are bought and sold. The stock market is also a measure of the general state of the economy and it allows the discovery of the price of corporate shares [74].

Stock trading is the practice of purchasing and selling shares of a publicly-traded business in an attempt to profit from the variation in their prices. This short-term perspective distinguishes stock traders from more conventional stock market investors, who often invest for the long term [24].

1.1.4.3 Bonds market

The bond market (sometimes referred to as the debt market or credit market) is a financial market in which participants may either issue new debt (referred to as the primary market) or acquire and sell existing debt instruments (referred to as the secondary market)[2].

The primary market is commonly referred to as the ”new issues” market since transactions take place solely between bond issuers and bond purchasers. In short, the primary market results in the production of entirely new debt securities that did not exist before. Securities that have previously been sold in the primary market are then acquired and sold at a later period in the secondary market. Numerous of online brokers provide this kind of bond for trading[3].

1.1.4.4 Foreign Exchange (Forex)

The forex market enables parties, including banks and individuals, to purchase, sell, and exchange currencies for hedging and speculating reasons. The forex market is the world’s biggest and most active financial market, with billions of dollars changing hands daily. The forex market does not have a central location; rather, it is an electronic network of banks, institutions, brokers, and traders [18]. Forex brokers operate as market makers, they are online platforms allowing individuals to execute exchanges easily with simple clicks, most famous forex brokers[4].

Foreign exchange can be as simple as changing (trading) one currency for another at a local bank. For example, one can swap the U.S. dollar for the euro. When trading on the forex market, currencies are listed in pairs, for example, USD/CAD means changing the United States dollar against the Canadian dollar, and USD/JPY means changing the United States dollar against the Japanese yen [17].

[2] https://en.wikipedia.org/wiki/Bond_market

[3] https://www.schwab.com/
https://www.tdameritrade.com/
https://us.etrade.com/

[4] https://www.ig.com/
https://www.home.saxo/
https://www.cmcmarkets.com/
https://www.forex.com/

1.1.4.5 Physical Assets

Investment in physical assets means the acquisition of tangible assets, such as metals, jewels, real estate, livestock, and anything else that has a physical form and a recognized value. A physical asset is essentially anything tangible that you own that can be sold or exchanged other than cash. Trader's aim in this market is that the price at which they may sell an item will be higher than the price at which they purchased it initially [34].

1.2 Artificial intelligence

Artificial intelligence (AI), the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings[5]. The term is frequently applied to the project of developing systems endowed with the intellectual processes characteristic of humans, such as the ability to reason, discover meaning, generalize, or learn from experience [22]

There are many forms and types of AI, In this thesis, we will mainly focus on one category, Machine learning which in turn can be divided into other categories

Machine learning according to Arthur Samuel (American pioneer in the field of computer gaming and artificial intelligence) is "the field of study that gives computers the ability to learn without being explicitly programmed" another definition by Tom Mitchell (research scientist and data science pioneer) "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E"

The majority of machine learning algorithms are exposed to a dataset. There are several ways in which a dataset may be described. In all cases, a dataset is a collection of examples, which are themselves collections of features [29].

The learning process starts with observations or data, such as examples, direct experience, or instruction, so that we can seek patterns in data and make better decisions in the future based on the examples we provide [65].

1.2.1 Types of machine learning

Machine learning is divided into four main categories as shown in Figure 1.4

1.2.1.1 Supervised learning

Supervised learning is an approach to creating artificial intelligence (AI), where a computer algorithm is trained on input data that has been labeled for a particular output. The model is trained until it can detect the underlying patterns and relationships between the input data and the output labels, en-

[5] https://www.britannica.com/technology/artificial-intelligence

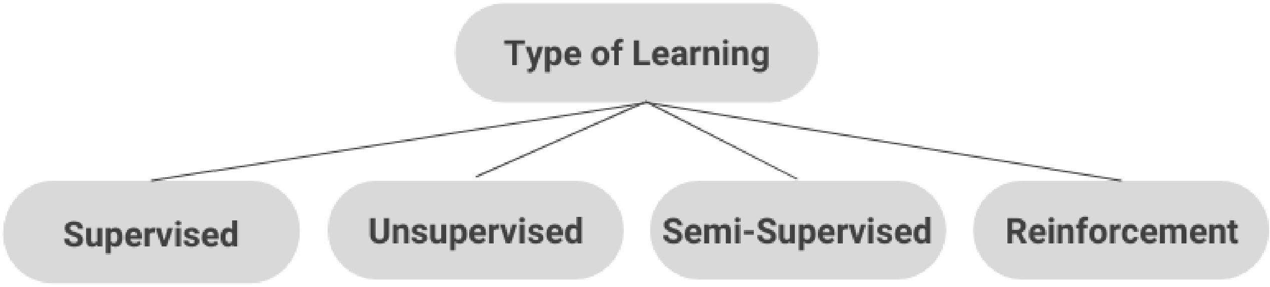

Figure 1.4: types of machine learning

abling it to yield accurate labeling results when presented with never-before-seen data [58].

When learning to categorize handwritten digits, for example, a supervised learning system takes thousands of images of handwritten digits along with labels containing the correct number each image represents, the algorithm will then learn the relationship between the images and the numbers associated with them, and use that knowledge to identify entirely new images (without labels) that it has never seen before, as shown in Figure 1.5.

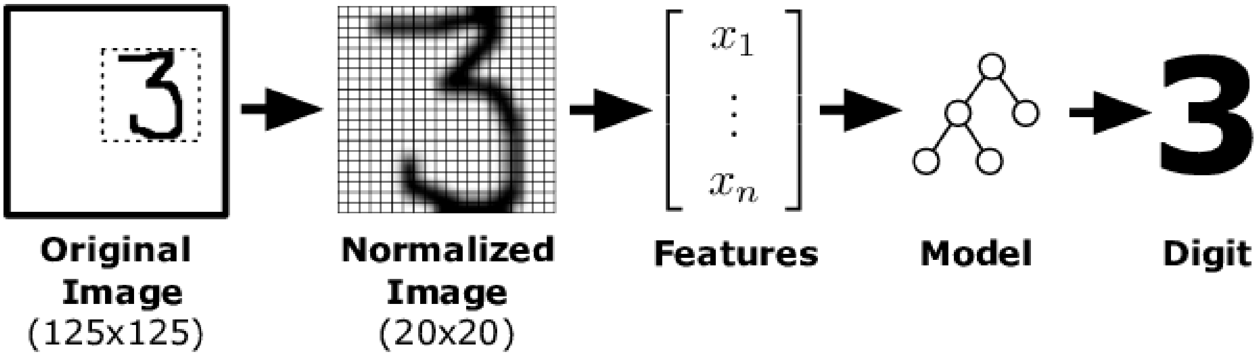

Figure 1.5: Handwritten digits classification model [57]

1.2.1.2 Unsupervised learning

Unsupervised learning refers to the use of artificial intelligence (AI) algorithms to identify patterns in data sets containing data points that are neither classified nor labeled. The algorithms are thus allowed to classify, label, and/or group the data points contained within the data sets without having any external guidance in performing that task. In other words, unsupervised learning allows the system to identify patterns within data sets on its own[6] [59].

How do you find the underlying structure of a dataset? How do you summarize it and group it most usefully? These are the goals of unsupervised learning. Examples of unsupervised learning applications are Customer segmentation or understanding, of different customer groups, around to build an effective marketing strategy as shown in Figure 1.6.

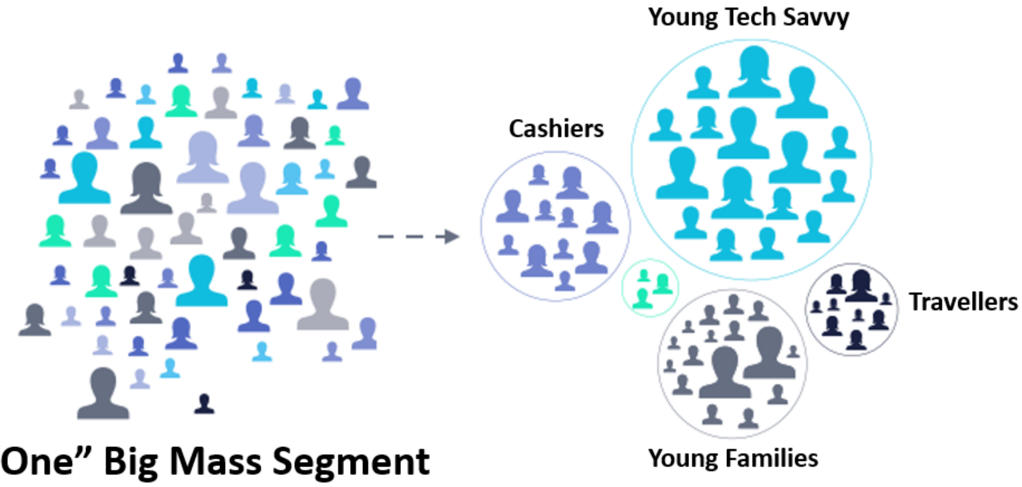

Figure 1.6: Unsupervised learning Segmentation [69]

1.2.1.3 Semi-Supervised learning

Semi-supervised learning is a learning problem that involves a small number of labeled examples and a large number of unlabeled examples. Semi-supervised learning is a type of machine learning that sits between supervised and unsupervised learning

1.2.1.4 Reinforcement learning

”Reinforcement learning is learning what to do—how to map situations to actions so as to maximize a numerical reward signal. The learner is not told which actions to take but instead must discover which actions yield the most reward by trying them. In the most interesting and challenging cases, actions may affect not only the immediate reward but also the next situation and, through that, all subsequent rewards. These two characteristics—trial-and-error search and delayed reward—are the two most important distinguishing features of reinforcement learning” [70].

[6] https://www.techtarget.com/searchenterpriseai/definition/unsupervised-learning

In a typical RL a learner and a decision-maker called “agent” interact with the environment. As a result of the agent’s actions, the environment provides rewards and transit to a new state. So, in reinforcement learning, we do not instruct an agent on how to do a task, but rather give it with rewards, which may be either positive or negative, depending on its actions [1]. And the agent will learn to choose actions that maximize the current and future rewards. This method of modeling the environment is called Markov-Decision Process (MDP) formalization

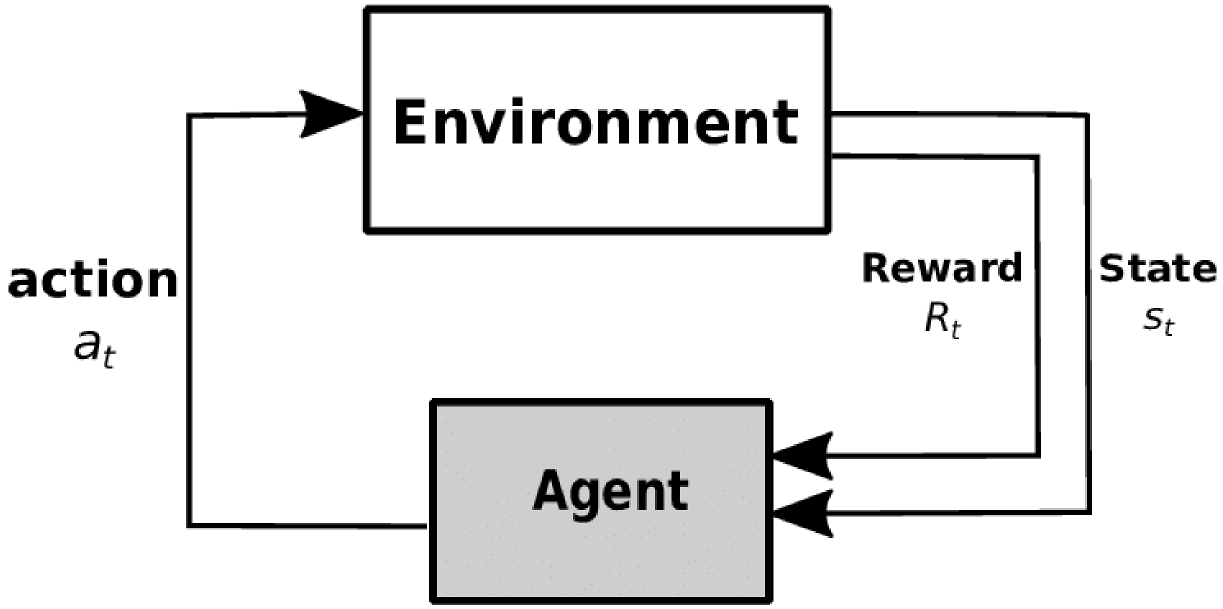

Figure 1.7: Reinforcement Learning, Agent and Environment [3]

- **Policy function** policy (π) describes the decision-making process of the agent. In general, a policy assigns probabilities to every action in every state, for example $\pi(s1|a1) = 0.3$ probability of taking action ”a1” when state is ”s1” is 30%. The policy thus represents a probability distribution for every state over all possible actions [26]. The policy function is commonly represented with the symbol (π)

- **Value function** The Value Function represents the value for the agent to be in a certain state. In simplified words, the value function represents how good to be in a specific state/ how good to take an action in this specific state. There are two main important value functions:

    - **State-Value function** The state value function tells us the value for being in some state when following some policy [26]. The state value function is commonly represented with the symbol (V)
    - **Action-Value function** The action value function tells us the value of taking an action in some state when following a certain

policy. The action value function is commonly represented with the symbol (Q)

1.2.2 Deep learning

The simple machine learning algorithms perform excellently on a wide range of significant problems, despite their simplicity. However, they were not successful in resolving the fundamental challenges of artificial intelligence, such as speech recognition and object detection, deep learning was developed in part as a result of the failures of standard algorithms to generalize effectively on AI challenges of this kind [29].

Understanding deep learning begins with an understanding of artificial neural networks since deep learning is just a sophisticated collection of tools for learning in neural networks.

Artificial Neural networks are simply an interconnected collection of simple processing pieces, units, or nodes whose operation is inspired by the animal neuron in some way. The interunit connection weights, store the network's processing capability, obtained by learning from a dataset [31]. The artificial neural networks are inspired by brain neurons. There are around 10 billion neurons in the human brain like the one shown in this figure 1.8.

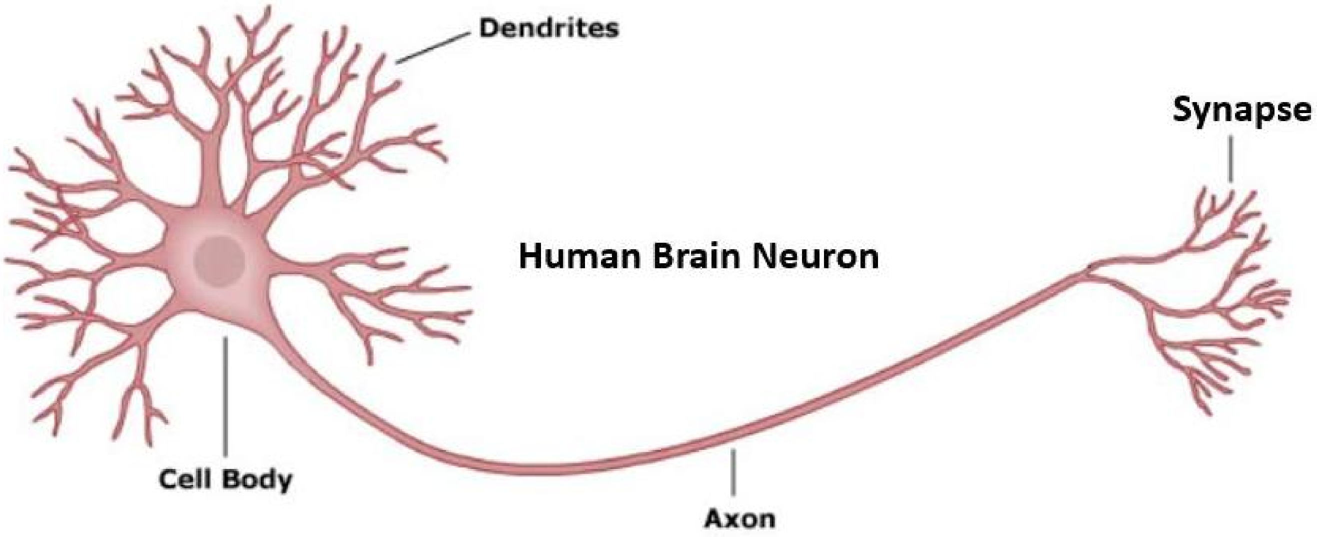

Figure 1.8: Illustration of a neuron [9].

A single artificial neuron (or perceptron) can be imagined as a single Logistic Regression model. This Figure 1.9 shows the artificial equivalents of biological neurons.

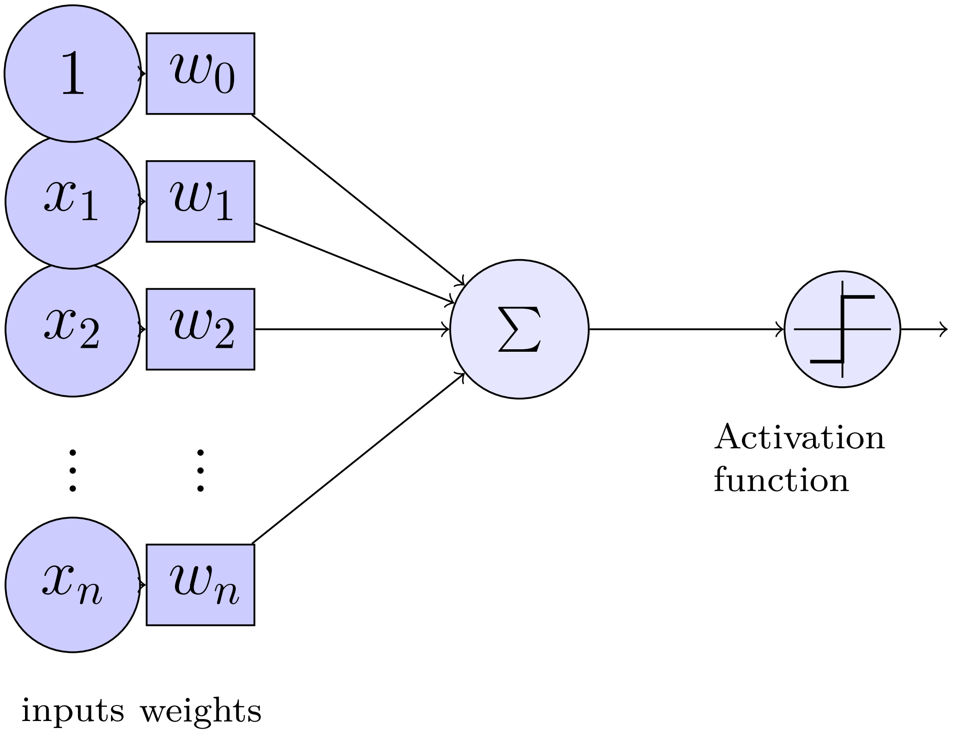

Figure 1.9: Artificial Neuron

A neural network is created by layering together neurons in several different configurations. There are three levels of layers: the input layer, which accepts the input of the model, single or multiple hidden layers, each of which accepts inputs from the previously hidden layer or the input layer, and an output layer, which accepts the input from the last hidden layer and as his name suggests will output the result (prediction, action, etc.) of the model as shown in this figure 1.10. The strength of neural networks lies in their capacity to approximate and map inputs to outputs by learning from examples in a complex nonlinear manner.

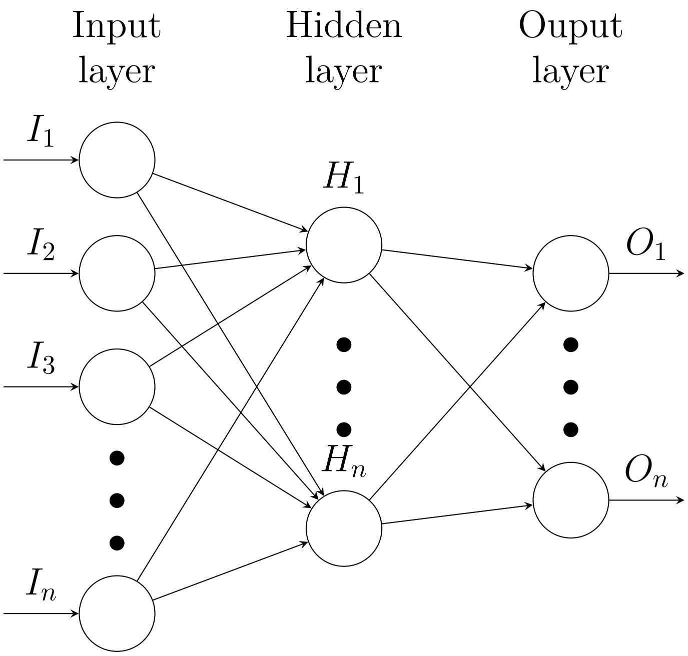

Figure 1.10: Artificial Neural Network ANN (Feedforward)

There are many different kinds of neural networks, and we will go through three important types.

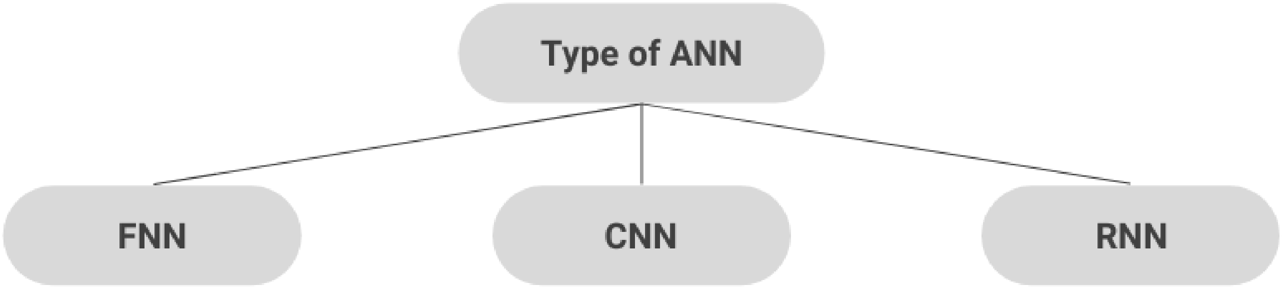

Figure 1.11: Types of Artificial Neural Network

1.2.2.1 Feedforward Neural Network (FNN)

Feedforward neural networks, also often called multilayer perceptrons (MLP) is the standard and simplest models of deep learning. Consists of multi perceptrons (Artificial neurons) stacked in multiple layers, with each layer's output serving as the input for the following layer. It is referred to as Feedforward since information is only processed in a single direction without cycles or recursion as shown in the figure 1.12. These networks are often referred to as "Universal Function Approximators" because they have the capability of learning weights that approximate any input to an output with a high degree of precision.

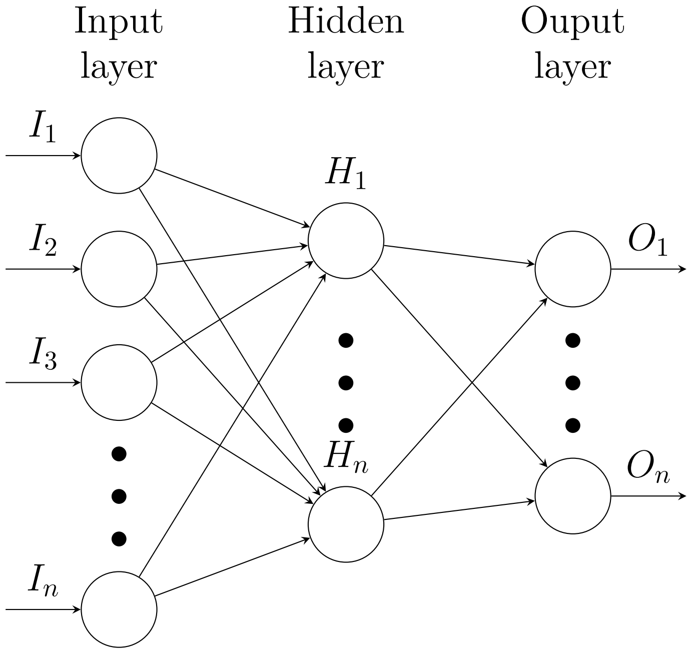

Figure 1.12: Feedforward Neural Network

1.2.2.2 Convolutional Neural Network (CNN)

Recent years have seen a significant increase in interest in convolutional neural networks within the deep learning community, due to their efficiency and performance in processing two-dimensional data with grid-like topologies, such as images and videos. When constructing CNNs, the general matrix multiplication operation used in standard Feedforward neural networks is replaced with a convolution operation, which helps in the detection of features in 2d and 3d matrices while simultaneously reducing the number of weights used and, as a result, the complexity of the network. Furthermore, the raw images can be imported directly to the network without the need for feature extraction to be performed beforehand. With the fast growth of computer hardware such as graphics processing units (GPUs) and tensor processing units (TPUs), training CNNs became easier and more efficient, opening the door to deeper networks with hundreds of layers. Currently, CNNs have been effectively used for a variety of tasks such as handwriting recognition, face detection, speech recognition, recommender systems, image classification, and natural language processing (NLP) [44]. As shown in figure 1.13, CNNs are composed of three distinct types of layers. Convolution layers (L=1, L=3), subsampling or pooling layers (L=2, L=4), (Typically, each convolutional layer is followed by a pooling layer) and finally fully connected or Dense layers, which are the same as in Feedforward networks.

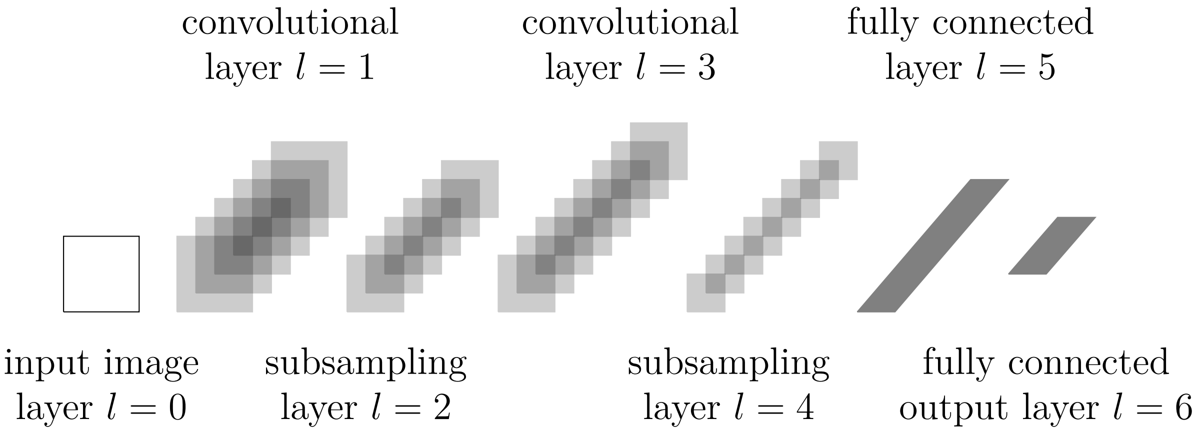
convolutional layer $l = 1$
convolutional layer $l = 3$
fully connected layer $l = 5$
input image layer $l = 0$
subsampling layer $l = 2$
subsampling layer $l = 4$
fully connected output layer $l = 6$

Figure 1.13: High-level illustration of a simple convolutional neural network indicating convolutional layers, pooling layers and fully-connected layers.

1.2.2.3 Recurrent Neural Networks (RNN)

All of the networks that have been presented so far deal with multidimensional data that has independent properties (every instance of the dataset is independent of other instances). However, some data kinds, such as text and time-series, as well as biological data, are, on the other hand, sequential and dependent on one another [2]. Examples of such dependencies:

- Predicting future stock prices based on historical data is an example of a time series. Without knowledge of past time steps, it is impossible to determine whether prices have increased or decreased at a given point in time. By treating every time step independently, we would lose this valuable information.

- By processing text with a bag of words technique the ordering of words in the document will be ignored this approach works fine in certain areas such as sentiment analysis, However in certain areas where the order of words is important such as voice command this approach will be inadequate [2].

To deal with sequential data Recurrent Neural Network (RNN) architecture is introduced. They can do this because they have loops that let data be stored inside the network (hence the name recurrent). This way, RNNs can pick up on sequential information in the data. It is possible to think of a recurrent neural network as a collection of numerous copies of the same network, each of which produces a signal to its successor as shown in the figure 1.14.

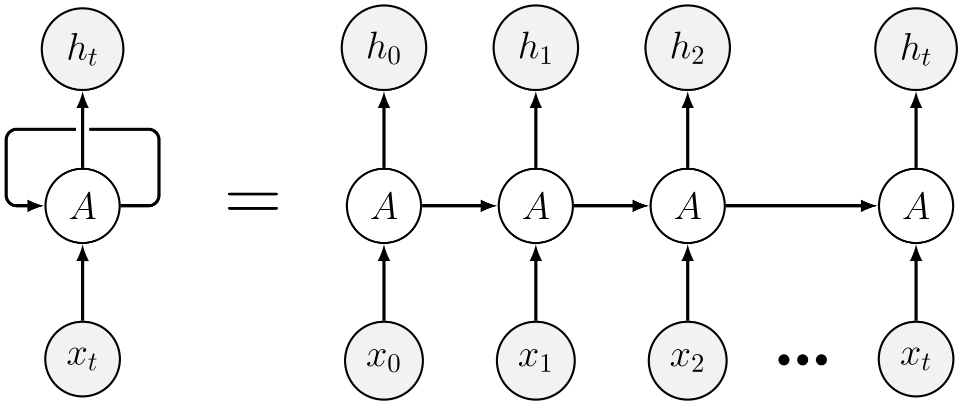

Figure 1.14: High-level illustration of a simple Recurrent Neural Network (RNN)

When working with RNN networks, there are a few complicating factors. One of them is the vanishing and the exploding gradient problem [2]. The vanishing happens during the backpropagation phase when the algorithm moves backward from output to input layer, gradients get smaller and smaller slowly approaching zero, leaving the weights of the first and lower layers almost unchanged which will cause the optimizing function (gradient descent, adam, etc) to never converge. In contrast, exploding gradients happen when gradients get larger and larger during the backpropagation step which eventually will cause the optimizer to diverge [10]. There is also the issue of RNN networks losing track of outdated data that may have been fed into the system. It is theoretically possible for RNNs to handle ”long-term dependencies,” but in practice, RNNs seem incapable of doing so [56], this problem was explored in-depth in this paper [8]. To overcome these issues, variants of RNNs were introduced, LSTMs and GRUs.

- **LSTM** Long Short Term Memory networks – often referred to as ”LSTMs” are a type of Recurrent neural network that is capable of learning long-term relationships between variables. LSTM networks were invented by Hochreiter and Schmidhuber [35]. And have since been refined and popularized by many other researchers. LSTMs are specifically intended to prevent the issue of long-term dependency in the first place. The ability to retain information over extended periods is their default mode of operation [56].

- **GRU** Gated Recurrent Neural Network was introduced in 2014 by Kyunghyun Cho, Bart van Merrienboer and Caglar Gulcehre [20]. GRU is a simpler version of LSMT with the same goal of addressing the issues of long-term dependency and vanishing/exploding gradient.

1.2.3 Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) is a relatively new branch of machine learning that combines two powerful techniques (Deep learning + Reinforcement learning). Reinforcement learning took birth in early 1980 as dynamic programming descendent Sutton and Barto explain the history and principles of reinforcement learning and the birth of DRL in their book [70]. In brief, reinforcement learning was used where the data were limited and few and the requirement were complex, however as a result of the emergence and advent of deep networks Reinforcement learning has gained more power to tackle more complex problems. More details about Reinforcement learning in this reviews [42, 4]

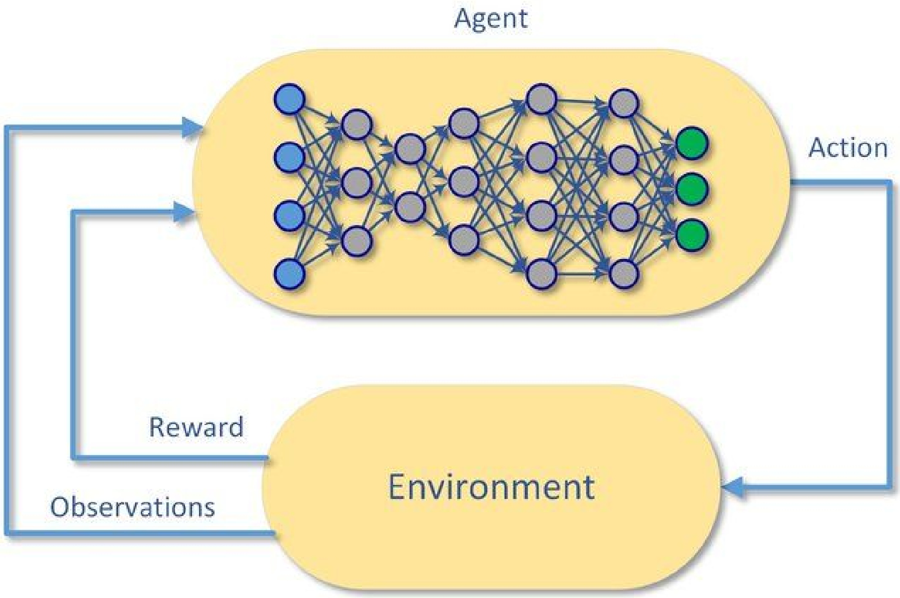

Figure 1.15: Deep reinforcement learning illustration [39].

In the literature, there are three main types of deep reinforcement learning

1.2.3.1 Critic-only

These methods are based on estimating the value function (State value-function/ Action value-function) of being in a given state. The learning (estimating the value function) early on was done by methods called dynamic programming but since the emergence of deep learning, the learning process was much more effective, fast, and efficient by using deep learning algorithms [4]. To take an action in the critic-only, we just use the action-value function. We can choose to be greedy and constantly choose the action that provides the highest possible future reward, or we can add an exploration factor so the agent will continue learning from new experiences [60]. One of the Critic-only disadvantages is it can't be applied on a continuous space of actions, works only on a discreet limited set of actions the agent can perform.

- **Deep Q-Network (DQN)** were originally proposed by DeepMind back in 2013 [53] and it was the first algorithm that brought the capabilities of deep learning to reinforcement learning. The only difference between Q-learning and DQN is that Q-learning employs temporal difference to estimate the Q function while Deep Q-learning uses a neural network to approximate it.

- **Deep SARSA** algorithm is very similar to Q-learning, the difference is in the way the value function is updated, update is done using the value of the future state and the action of the current policy [47].

1.2.3.2 Actor-only

If we consider the policy to be a probability distribution over the whole action space, Policy Gradient techniques may be used to optimize that distribution such that the agent picks the action with the highest probability, out of all the potential actions, that yields the highest reward [68]. A major advantage of actor-only methods over critic-only methods is that they allow the policy to generate actions in the complete continuous action space.

- **Deterministic Policy Gradient (DPG)** This is the first algorithm that deals with continuous action space, it was first introduced in this paper [68].

- **Augmented Random Search (ARS)** The ARS is an improved version of BRS (Basic Random Search), it was first introduced in 2018 in this paper [49]. The authors state that they have created an algorithm that would be at least 15 times more effective than the fastest competing model-free approaches.

1.2.3.3 Actor-critic

In the Actor-Critic method, the policy is referred to as the actor that proposes a set of possible actions given a state, and the estimated value function is referred to as the critic, which evaluates actions taken by the actor based on the given policy. Actor-critic methods combine the advantages of actor-only and critic-only methods. While the parameterized actor brings the advantage of computing continuous actions the critic's estimate of the expected return allows for the actor to update with gradients that have lower variance [4]

- **Proximal Policy Optimization (PPO)** Introduced by the OpenAI team in this paper [63] and it has become the primary reinforcement learning algorithm at OpenAI due to its simplicity of use and high performance. The objective was to create an algorithm with data efficiency and consistent performance that perform comparable to or better than state-of-the-art techniques while being considerably easier to develop and tune.

- **Asynchronous Advantage Actor-Critic (A2C or A3C)** This algorithm was first mentioned in 2016 in a research paper appropriately named Asynchronous Methods for Deep Learning [52] by Google's DeepMind team which is the Artificial Intelligence branch of Google. And it was the first algorithm that uses two neural networks (policy and value function network). A3C is an asynchronous version of A2C.

- **Deep Deterministic Policy Gradient (DDPG)** is a model-free off-policy algorithm for learning continuous actions It combines ideas from DPG (Deterministic Policy Gradient) and DQN (Deep Q-Network), it uses DPG for policy network and DQN for value function network [43].

CHAPTER 2

Deep reinforcement learning for trading

Reinforcement learning has shown to be one of the most effective Machine Learning approaches over time, and it has been used and tested in a variety of areas, including the financial industry. In this chapter, we will discuss the application of reinforcement learning in the financial world (more specifically in trading), problems that have been solved using reinforcement learning, and some fundamental concepts such as risk measurements and benchmarks that will help to fully comprehend what will be covered in the following chapter chapter 3.

2.1 Application of Reinforcement Learning in trading

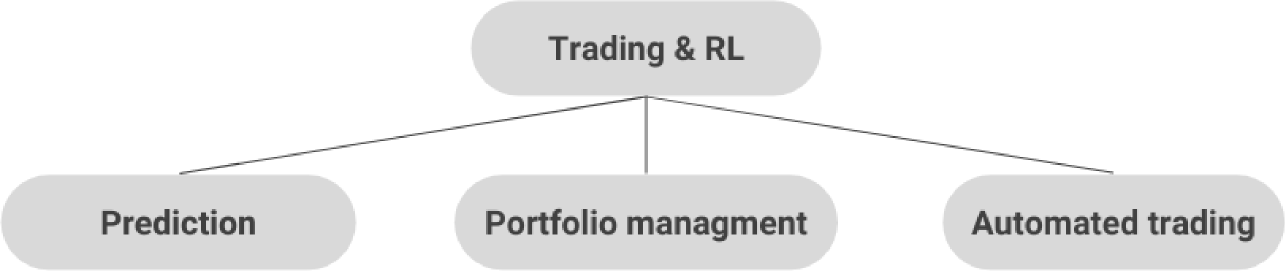

Figure 2.1: Application of RL in trading

2.1.1 Prediction

Prediction is the process of estimating what will happen in the future based on historical data (what we've seen so far), in our case "trading", predicting is the act of forecasting the prices of the assets (whether the price will go up or go down and by how much). And generally, the predictions serve as prior information for the agent who is trading.

The predictions can be quite useful information in many markets with low volatility and noise, such as the stock market, fiat currencies, and other markets like gold and silver, but when it comes to high-volatility markets, such as crypto-currency, where prices can rise or fall by more than 1000 percent in a single day the predicted information has a large margin of error and can mislead the trading agent resulting in a big loss of money. In the coinmarketcap [1] website we can see almost every day some currencies rising or falling more than thousands of percent overnight.

Many publications in the literature use deep-learning approaches to address the problem of price prediction, this survey explains most of the work done in predicting forex prices [36].

This paper compares the three approaches CNN, LSTM and RNN in predicting stock prices [64]. As shown in the figure 2.2, all approaches (CNN, RNN, LSTM) perform well in predicting the price of the TCS (Tata Consultancy Services) company shares. However the RNN and LSTM model didn't identify the pattern at the beginning where the CNN captures the trend more accurately. And according to the author the CNN model performs the best and it is (tested on three different company shares INFOSYS, TCS, and CIPLA) because it makes predictions based on the information available at a certain time. Despite the fact that the other two models (RNN, and LSTM) are utilized in a variety of different time-dependent data processing applications, they do not outperform the CNN architecture in this instance. This is due to the rapid shifts that occur in the stock market regularly. Stock market changes may not necessarily occur predictably or follow the same rhythm.

When it comes to Reinforcement learning the environment is simulated with historical data, and the agent will try to predict the next price point. There are fewer works using Reinforcement learning like this paper [40], there are even some works that combine CNN and LSTM with Reinforcement learning [67].

[1] https://coinmarketcap.com/gainers-losers

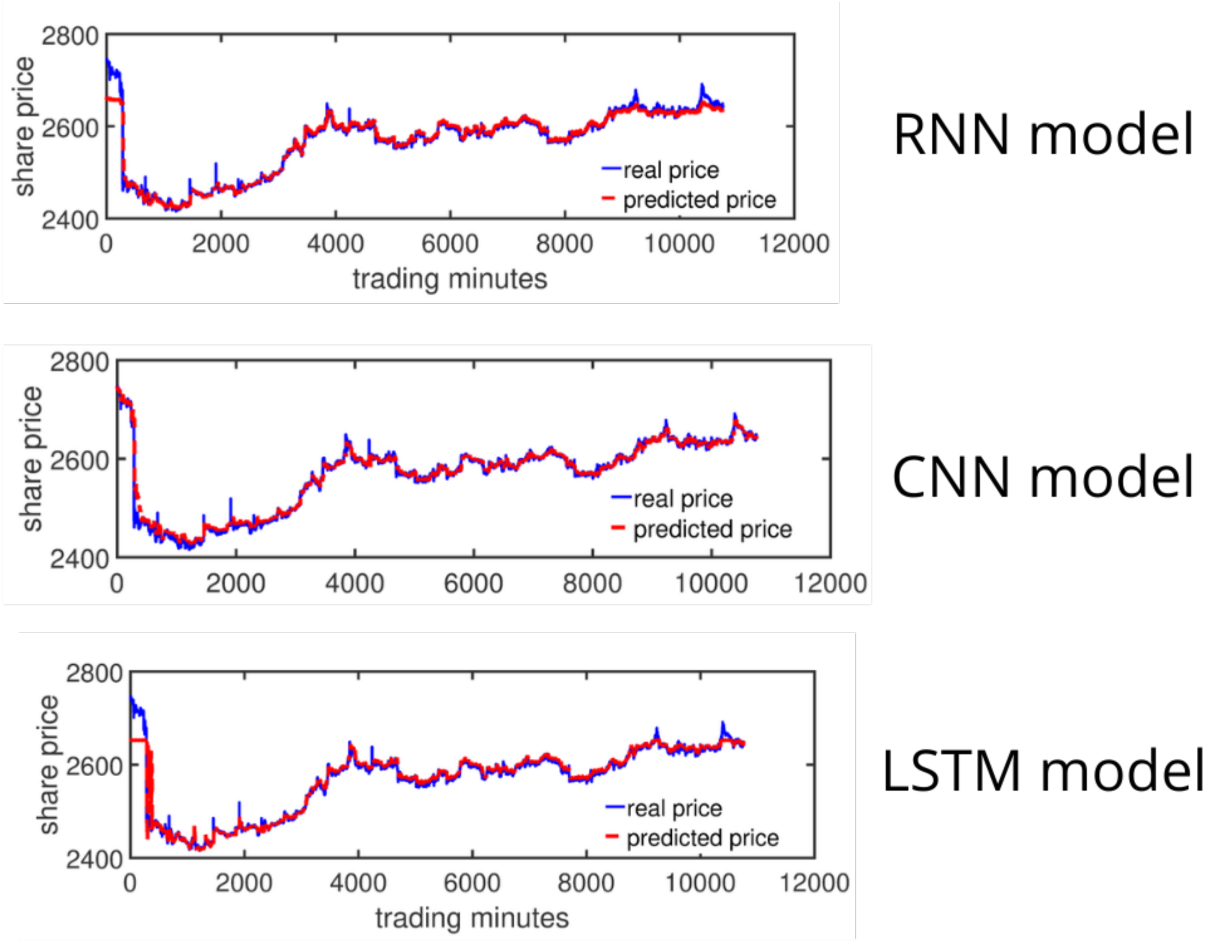

Figure 2.2: Plot for Real value vs Predicted value for TCS using RNN, CNN and LSTM [64]

2.1.2 Portfolio management

In the financial world, it's advisable to diversify investments. This means that instead of placing all of the resources in one area, we should split them around to reduce the risk of losing everything. But, how to divide resources into separate assets ?. And how much should we put into each asset? In a nutshell, this is the problem of portfolio management. Usually, experts in the field of investment will be employed to manage the portfolio and make all essential decisions to maximize the value of the portfolio. However, with the advancement of deep learning and reinforcement learning, this process might be effectively automated, with the potential to beat human specialists.

In the crypto-currency world for example there are hundreds of currencies on the Binance platform (the largest platform for trading cryptocurrencies), each crypto-currency can form a pair with other stable coins like USDT. As a result, there are many options to consider, and making a decision might be difficult even for expert people. Hence there is an increasing need for an intelligent automatic solution for the portfolio management problem. We can formulate the problem of portfolio management as follow

$$w = \{w_1, w_2, ..., w_n\} \qquad \sum_{i=1}^{n} w_i = 1 \tag{2.1}$$

$$\tag{2.2}$$

Where ”n” is the number of assets we want to invest in

Essentially the problem of portfolio management narrows down to finding the weights “wi” that maximize the overall value of the portfolio [51]

There is a lot of work in the literature addressing the problem of portfolio management, for a comprehensive review see [62]. Works that address portfolio management with reinforcement learning [81, 46, 77, 28, 38]

2.1.3 Automated trading

Automated trading essentially means making trade decisions such as buying, selling, or holding assets automatically without human interventions to maximize profit.

The ability to digest data, chart, and anticipate market states to predict future ones is what makes humans a successful traders, and most of this ability comes from experience (by seeing a lot of market movements). However, no human being can surpass machine learning algorithms in terms of data processing and learning from experience, thus machine learning algorithms are most likely to outperform humans in trading. Unfortunately, because this is a competitive profession, most research and height accuracy datasets have been kept secret, nevertheless, there are a few that are available to the public. More details about public research in this area are in chapter 3.

2.2 Financial market as an environment for RL

In this section, we will try to find the specifics (modeling, choices) of the trading problem in the context of Deep Reinforcement Learning. When we talk about solving a problem with reinforcement learning three things need to be done.

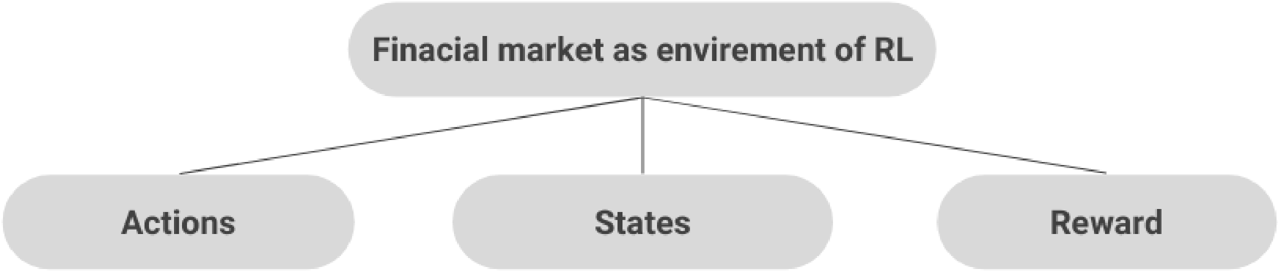

Figure 2.3: Financial market as environment for RL

- Defining the actions our agent[2] can make
- Defining the reward function
- Defining the environment and states

In each of the three aspects, we will discuss the options and choices presented in the literature.

2.2.1 Action space

In reinforcement learning, actions are what the agent can do in the environment to receive a reward or a penalty. In the context of trading, actions are essentially buying and selling signals. These actions can be represented in different ways: a continuous or limited set of actions, a fixed or a variant amount of assets are bought or sold each time, and many other possible choices.

The decision between a continuous or restricted set action space is determined by the type of algorithm employed to build the agent (critic-only, actor-only, and actor-critic see this subsection 1.2.3 for more information about different approaches). However this paper [41] presented an innovative solution to the discrete action space issue in critic-only methods (more specifically in Deep Q-learning), the goal is to create an expert model of each action that matches the behavior of investors. Essentially, the reinforcement learning agent generates three signals (actions): BUY, SELL, and HOLD, and an expert system is picked for each action. The expert system is responsible for the quantity utilized (bought or sold), as well as for generating the reinforcement learning agent's reward.

[2] The reinforcement learning model who is responsible for yielding the actions and decisions is generally referred to as Agent

There is another option to deal with discrete action space in cretic-only methods, which is to have several actions for each BUY and SELL signal, where each action buys or sells a defined quantity of the currently available resource. For example $\{HOLD, BUY{-}1, BUY{-}5, , SELL{-}1, SELL{-}5, ...\}$. Action BUY-1 will buy only one share the action BUY-5 will buy 5 shares and the same things for the sell signals. Therefore, the agent has more control over what we should do (sell or buy) and how much resources we should use.

2.2.2 State representations

In Reinforcement Learning, state refers to what the agent can see and observe from the environment, and it is this state that determines what action the agent will take. In the case of a robot that is trying to learn to walk, the state is the placement of its legs. A Go game's state is where the pieces on the board are placed. In the case of trading, there are many ways to represent the state from OHLCV historical prices to simple technical indicators. We will look at several representations found in the literature.

2.2.2.1 Raw OHLCV data

OHLCV is a market data aggregation that stands for Open, High, Low, Close and Volume. The OHLCV data consists of five data points: the Open and Close, which reflect the initial and latest price levels reached over a specific time, respectively. The terms High and Low refer to the highest and lowest prices that were obtained during that period. The entire quantity of assets transacted within that period is referred to as volume.

Many papers used multiple adjacent rows of OHLCV data as a single state. For example, when employing a window size of five, the state at a time "t" is the five preceding OHLCVs starting from time "t", this paper [19] (reviewed in details in chapter 3) used this state representation.

The OHLCV representation is known to be noisy and difficult to interpret, some techniques are employed to mitigate this problem. For example, this paper [79] presented preprocessing techniques based on GRUs to extract simpler features from the raw OHLCV window (this paper is reviewed in detail in chapter 3).

2.2.2.2 Technical indicators

Technical indicators are mathematical patterns generated from OHLCV data that may reveal hidden patterns in noisy data. Many traders rely on these indicators to make trading decisions, thus including them as features in our trading agent makes sense. More about technical indicators in subsection 1.1.3, The encyclopedia book [21] covers most if not all technical indicators.

The purpose of this article [16], was to investigate the use of basic mathematical adjustments (technical indicators) in minimizing the complexity of the state space and simplifying the raw data OHLCV. And according to the author and the experiment findings, employing basic mathematical procedures (technical indicators) on the price data can effectively decrease state-space complexity while retaining information robustness.

2.2.2.3 Convolutions

CNN has been used before in time series encoding and forecasting such as stock price prediction [64]. This means we can use CNNs for encoding asset historical prices and one way to do that is to concatenate the time-series into a tensor, as shown in Figure 2.4 from [12], the author used a 3D-blocks named as history block where the X-axis is the window size “w” (Typically, multiple adjacent time-steps are provided as a state, such as the latest 5 or 10 steps) and Y-axis representing each coin (USDT, BTC, etc) and the Z-axis representing the features from the dataset in this case a simple Open, Hight, Low, Close, Volume (OHLCV).

Another convolutional representation is used in this paper [38]. The author called this representation “Price Tensor” as shown in Figure 2.5, the state is of shape (f, n, m) where “m” is the number of assets, ”n” is the number of features (OHLCV) and ”f” is the window size.

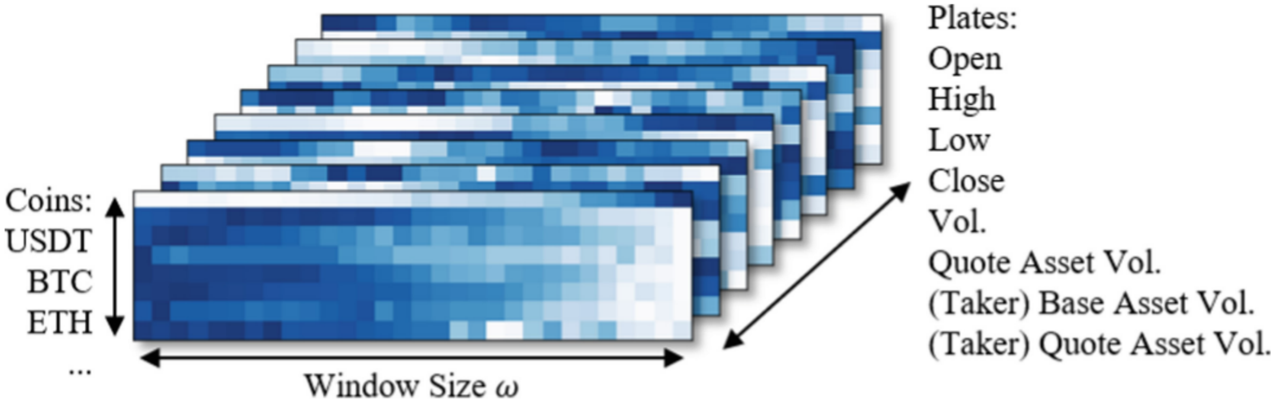

Figure 2.4: Preprocessed 3D input state of market history [12].

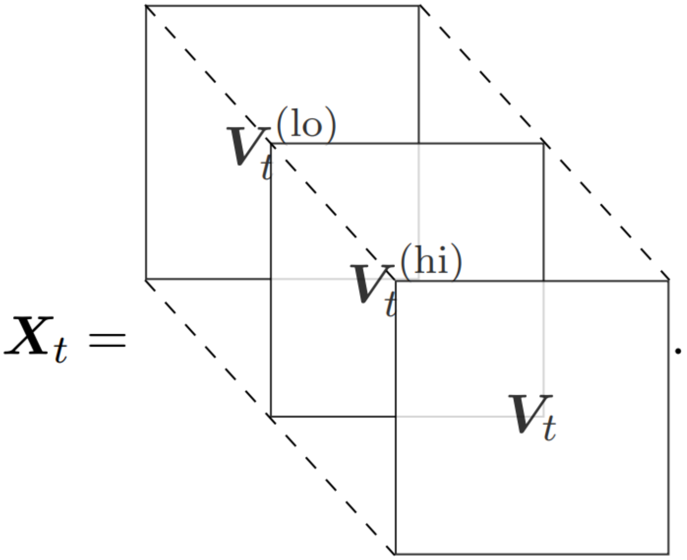
$\boldsymbol{X}_t =$
$\boldsymbol{V}_t^{(\text{lo})}$
$\boldsymbol{V}_t^{(\text{hi})}$
$\boldsymbol{V}_t$

Figure 2.5: Price tensor representation [38].

2.2.3 Reward functions

The reward function is responsible for informing the agent of what is right and what is wrong via the use of rewards and punishments (reward functions represent the feedback to the agent). Moreover, the primary objective of the reinforcement learning agent is to maximize the reward. This implies that the agent's behavior is entirely dictated by the reward function. As a result, it is vital to have a good reward function in place for a good performance. From risk-based measurements to profitability or cumulative return, there are many different types of rewards functions we can choose from, see this [61] article for a comprehensive review about reinforcement learning reward functions in trading.

2.2.3.1 Risk based reward:

Risk quantification is the process of measuring and evaluating the risk of a particular investment or trade. This is useful information that might be included in the reinforcement learning reward function to make the agent aware of the dangers involved in each trade. Some different techniques and formulas can be used to measure the risks we will list some of them:

Sharp ratio/Sortino ratio

The Sharpe ratio [66] is also called the reward-to-variability ratio and is the most common portfolio management metric that indicates how well an equity investment is performing compared to a risk-free investment, taking into consideration the additional risk level involved with holding the equity investment, The Sortino ratio is a variation of the Sharpe ratio measures the performance of the investment relative to the downward deviation. Unlike Sharpe, the Sortino ratio does not consider the total volatility of the investment [50]. Sharp/Sortino ratio measures were used successfully in this paper [80, 45].

The author of this paper [45] tested both cumulative and risk-based reward functions on the same models, and the results revealed that the model with the risk measurement reward function outperformed the cumulative reward function. (More about this experiment in chapter 3).

$$S_a = \frac{E[R_a - R_b]}{\sigma_a} \tag{2.3}$$

where:

S_a = Sharpe ratio
E = expected value
R_a = asset return
R_b = risk free return
σ_a = standard deviation of the asset excess return

Value at Risk (VaR)

VaR is a statistic that quantifies the extent of possible financial losses within a portfolio over a specific time frame. This metric is most commonly used by investors to determine the extent and probabilities of potential losses in their portfolios [11]. VaR was used successfully in reinforcement learning trading algorithm in this papers [72, 30, 81].

$$V_r = V_m * \frac{V_i}{V_{i-1}} \tag{2.4}$$

where:

V_r = Value at risk
V_i = the number of variables on interval i
V_m = m is the number of intervals from which historical data is taken

2.2.3.2 Profit and Loss based rewards (PnL):

"Profit and Loss" is a financial statement that summarizes the total revenues, costs, and expenses in a period, in companies it's usually a quarter or a half year. In our case, it's the period between the current time and the last time the agent took an action (buy or sell). There are many mathematical formulas for expressing Profits and Losses we will go through some of them:

Profit & Loss (P&L)
Revenues
Expenses
Profit / Loss
Statement of Operations

Figure 2.6: Profit and Loss (PnL) [27].

Simple profit

This is the simplest reward function, it calculates the ratio between current net worth and the previous net worth. The formula for simple profit reward function is described in Equation 2.5.

$$Reward = \frac{current\_networth}{previous\_networth} - 1 \tag{2.5}$$

Realized PnL

The realized PnL is determined based on the closing price and entry price. The realized PnL refers to the profit or loss that comes from closed positions, not the market price. It only has a direct connection to the price of the orders that were put in.

$$RPnl = \sum_{i=1}^{Bi} Q_b(i) * P(i) - \sum_{i=1}^{Si} Q_s(i) * P(i) \tag{2.6}$$

where:

$RPnl$ = Realized PnL
Bi = Total number of buy operation
Si = Total number of sell operation
$Q_b(i)$ = Quantity bought at time i
$Q_s(i)$ = Quantity sold at time i
$P(i)$ = Price at time i

Unrealized PnL

On the other hand, the unrealized PnL is always changing because it uses the price of the market rather than the price when the order was placed in.

$$RPnl = \sum_{i=1}^{Bi} Q_b(i) * P - \sum_{i=1}^{Si} Q_s(i) * P \tag{2.7}$$

where:

$UnRPnl$ = Unrealized PnL
Bi = Total number of buy operation
Si = Total number of sell operation
$Q_b(i)$ = Quantity bought at time i
$Q_s(i)$ = Quantity sold at time i
P = Current price of the asset

2.3 Benchmarks

Some stiff trading strategies could be used as a baseline to compare and evaluate the performance of reinforcement learning trading agent

2.3.1 Buy and Hold (BH)

BH is a traditional passive investment approach, in which an investor purchases stocks or other assets and holds them for a long period. Because some assets, such as stock shares for big companies, tend to increase in value over time (years, centuries), buying and holding for a long period often result in a profit. An investor who uses a BH strategy actively selects investments but has no concern for short-term price movements [7]. As the author of this study did [19], we can compare the return of this method to the returns of reinforcement learning agents to evaluate our agent.

2.3.2 Turtle strategy

The turtle strategy is a well-known trend-following strategy first created by Richard Dennis in 1979 and still used as a valid benchmark, it consists of set rules to follow when trading and can be automated. The general idea is to buy breakouts and close the trade when prices start consolidating or reverse. Short trades must be made according to the same principles because a market experiences both up-trends and down-trends. While any time frame can be used for the entry signal, the exit signal needs to be significantly shorter to maximize profitable trades [15]. This strategy is used as a benchmark in this paper [80].

2.3.3 Simple Moving Average (SMA) crossover

Simple Moving Average (SMA) crossover is a technique based on moving average indicators; it employs SMA indicators of different lengths (most often SMA 50 and SMA 200) and uses their crosses as a signal for entering or exiting trades. When the SMA 50 crosses above the SMA 200, it generates a buy signal (golden cross), and when the SMA 50 crosses beneath the SMA 200, it generates a sell signal (death cross).

2.3.4 Relative Strength Index (RSI) divergence

Relative Strength Index (RSI) divergence is a strategy based on RSI indicators. Generally, an RSI divergence means that the RSI indicator is moving in the opposite direction compared to the price. Therefore, while the price is moving, the RSI is telling us in advance to anticipate a change in the direction. There are two types of rsi divergence.

- **Positive RSI Divergence** When the price movement reflects lower lows and lower highs, but the RSI indicator is indicating the contrary, the price action is bearish. This divergence may function as a buy signal.

- **Negative RSI Divergence** It applies to positive trends characterized by higher closing highs and lows. This divergence could be used as a Sell signal.

CHAPTER 3

Automated trading with reinforcement learning

In this chapter, we will examine and analyze various methods of automating trading using reinforcement learning that has been proposed in the literature

3.1 Critic-only DRL

Recalling that critic-only methods have one big disadvantage over other methods, it works only on a discreet limited set of action (it can't be applied on a continuous space of action), This is the most published approach in the literature, the majority of them use DQN or an enhanced version of it.

Inspired by the work done in gaming fields and the results that showed DRL agent can outperform human beings in games like atari [53], the author of **"Application of Deep Reinforcement Learning on Automated Stock Trading"** paper [19], tries to solve the problem of auto-trading by applying DQN and DRQN in stock trading. The author replaced the fully connected layer in DQN with a recurrent LSTM layer of the same size to provide the agent with the latest continuous daily stock data (avoid the trading process to be Partially Observable Markov Decision Process). The dataset used is 19 years of S&P500 ETF price history (daily closing prices), obtained through Yahoo Finance, 5 years used for training, and 11 years was used for testing, BH and Random action-selected DQN trader benchmarks were used to evaluate the agent. Agent action space is composed only of three actions (buy a share, sell a share, do nothing). Depending on which action was taken, the reward is calculated as the difference between the next day's adjusted close and the current day's adjusted closing price. As we can see in the figure 3.1 and table 3.1 both models managed to make profit but the model with LSTM layer DRQN outperform the DQN model. The DQN method generates a 22.33% annualized return while the DRQN generates 23.48%.

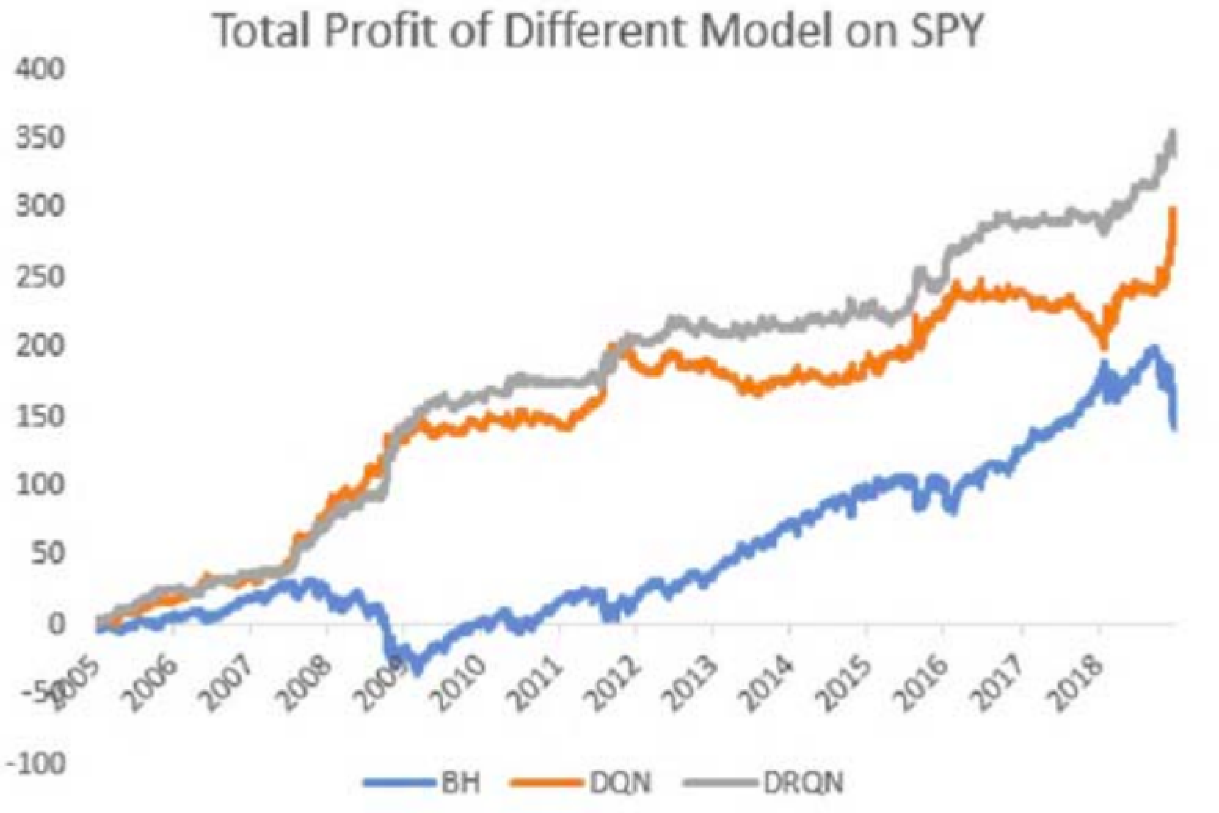

Figure 3.1: Total Profit Curves of BH, DQN and DRQN on the SPY Test Dataset [19]

Table 3.1: Average accumulated reward of different models.

	Performance Comparison			
	BH	**Random DQN**	**DQN**	**DRQN**
Average Total Profits	159.4	5.0	285.8	338.8

Limitations:

- The agent was evaluated only on one stock. Which makes it difficult to make a general conclusion about the algorithm since some stocks tend to be more profitable than others.

- The reward function didn't use any risk measurement which makes it dangerous to use this model on live data.
- Actions are limited to buying or selling one share at a time.
- The model was trained on a noisy dataset without using any technical analysis indicator or any sort of prepossessing to remove the noise from the data.
- Author didn't include transaction costs.

The paper titled **"Financial Trading as a Game: A Deep Reinforcement Learning Approach"** [37], is strongly related to the previous one "Application of Deep Reinforcement Learning on Automated Stock Trading" [19] but with major improvements, the author used an enhanced version of DRQN based agent more suited for the financial trading task by using smaller replay memory and sampling a longer sequence for training. The agent is trained on 5 years of 15-minute time frame forex dataset (from 2012 to 2017) of 12 currency pairs, The author extracted 16 features from the raw OHLCV data. This paper introduced a novel action augmentation technique to mitigate the need for random exploration in the financial trading environment and according to the table 3.2 this technique resulted in 6.4% increase in annual return compared to using ϵ-greedy for exploration, the reward function used here is the portfolio log returns. The agent was able to obtain an average of 10% annual increase with all pairs being positive, pair CHF-JPY managed to make 60% annual increase which is impressive

- Rather than using raw OHLCV as in the previous paper [19], the agent was trained on a normalized and cleaned dataset.
- The author used portfolio log returns as a reward function which is an improvement over the difference between days adjusted close price used in the first paper [19]. But still, no risk measurement is included.
- This paper was more generalized as it tries 12 different pairs of currencies and the results were positive on all pairs.

	ϵ-greedy	Action Augmentation	Gain
GBP-USD	13.7%	16.2%	2.5%
EUR-USD	7.1%	9.5%	2.4%
AUD-USD	6.4%	14.8%	8.4%
NZD-USD	9.5%	17.1%	7.6%
USD-CAD	-4.1%	12.2%	16.3%
EUR-GBP	7.1%	12.8%	5.8%
AUD-NZD	28.1%	34.4%	6.3%
CAD-JPY	17.9%	20.4%	2.5%
AUD-JPY	20.3%	25.0%	4.8%
CHF-JPY	57.0%	60.8%	3.8%
EUR-JPY	15.0%	23.6%	8.6%
GBP-JPY	30.9%	39.0%	8.1%
	17.4%	23.8%	6.4%

Table 3.2: Annualized returns with and without action augmentation [37]

Limitations:

- Same as first paper [19] reward function didn't address risk.
- Action space is limited to only 3 actions.
- Author included transactions costs.

The main difference in the paper titled **”Adaptive stock trading strategies with deep reinforcement learning methods”** [79], is the deep learning technique Gated Recurrent Units (GRU) used to extract features from stock markets raw data and technical indicators for the DQN model, as shown in the figure 3.2, this method is mainly used to improve the accuracy and robustness for the representation of stock market conditions [79]. The agent is trained on 10 years (from 2008 to 2018) of OHLCV stock dataset of 15 companies from the United States, United Kingdom, and China as shown in table 3.3. New reward function is used and it includes the Sortino Ratio (SR) risk measurement, something all reviewed models so far lack, the state is the usual OHLCV plus technical indicators like MACD, MA, EMA, OBV. Action space is also limited to only 3 actions (buy a share, sell a share, hold). The architecture of the model is summarized in the table 3.4. The results of this model are fascinating, managing over 171% return on a Chinese stock, only

2 stocks out of 15 were negative. This paper also introduced an actor-critic method DPG (will be reviewed in the next section) which turns out to be more stable and performant than the critic-only method GDQN.

Limitations:

- Same as other studies action space is limited to only one share.
- Author didn't include transaction costs.

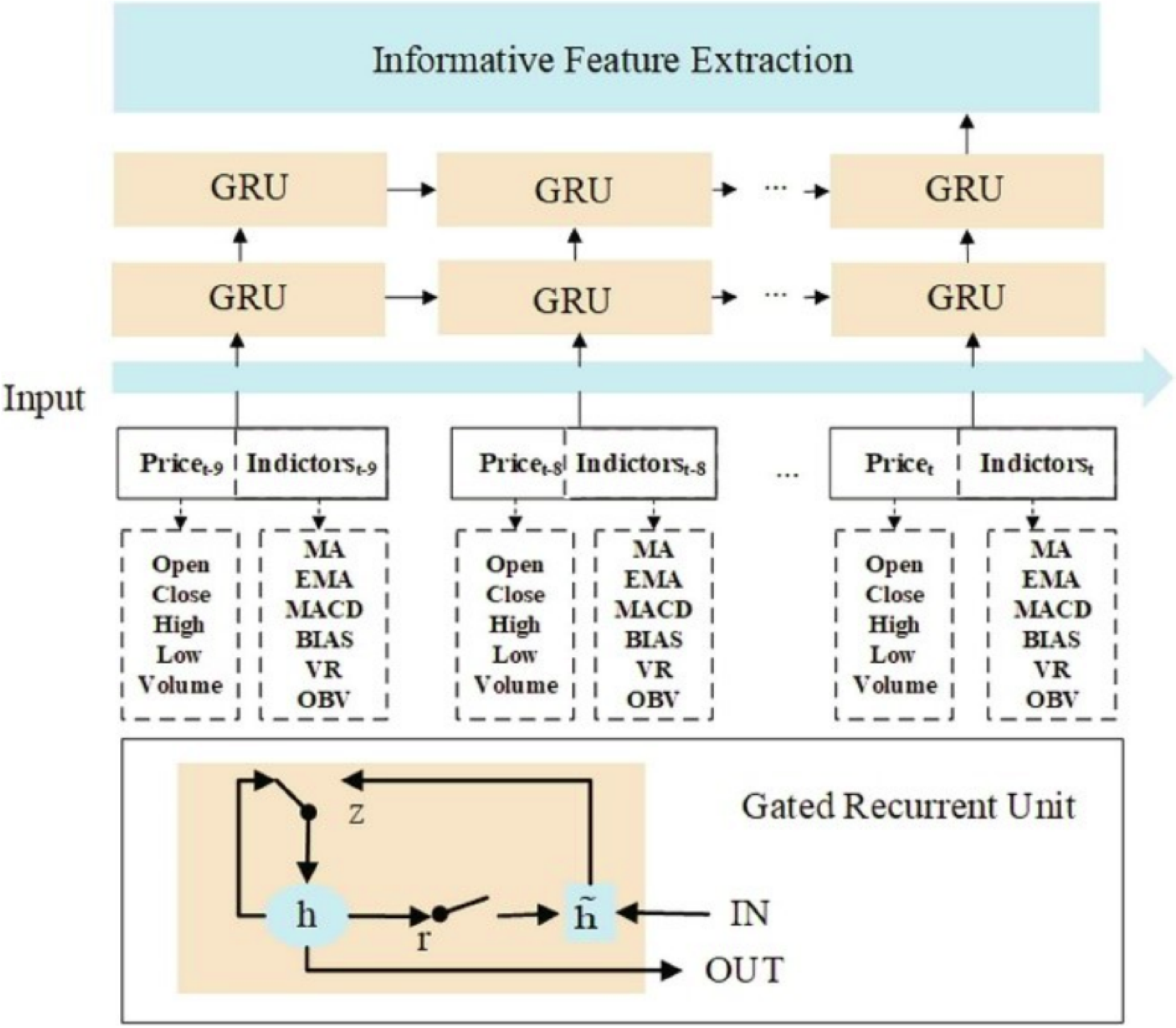

Figure 3.2: Gated Recurrent Unit (GRU) architecture [79]

Market	Symbol	Company
The U.S. stock	AAPL	Apple, Inc.
	AXP	American Express
	IBM	International Business
The U.K. stock	RDSB	Royal Dutch Shell
	ULVR	Unilever
	BATS	British American Tobacco
The Chinese stock	600519	Kweichow Moutai
	601288	Agricultural Bank of China Bank
	601398	ICBC

Table 3.3: Sample stocks in three stock markets [79]

Layer	Outout Shape	Parameter
GRU	(10, 32)	5088
Dropout	(10, 32)	0
GRU	32	6240
Dropout	32	0
Dense	3	99

Table 3.4: The parameter settings of the Q network of GDQN [79]

The paper titled **”A Deep Reinforcement Learning Approach for Automated Cryptocurrency Trading”** [45], is the only paper in the thesis that deals with the cryptocurrency market. The author proposed a system of Q-learning based on:

- Deep Q-learning Network (DQN)
- Double Deep Q-learning Network (D-DQN)
- Double Dueling Deep Q-Learning Network (DD-DQN)

And with two different reward functions:

- Sharp-ratio function (see more about Sharp-ration in this subsubsection 2.2.3.1)
- Simple profit function

The dataset used for training and testing is historical OHLCV data of bitcoin prices in US dollars from December 1, 2014, to June 27, 2018, collected at 1-minute intervals. To make the experiment more realistic, the author set the transaction fee to 0.3 percent of the transaction price, and to promote greater exploration at the start of training, the probability of exploring is set to 1 for the first 300 iterations and set to 0.13 after that (epsilon greedy strategy), this means the agent will be continually learning and adapting to new market trends even after the training phase is over. The DQN system is made up of two CNN layers with 120 neurons each, and both epochs and batch sizes are set to 40. The Mean Squared Error is applied as a loss function, and the ADAM algorithm is used as an optimizer. In all layers, the Leaky ReLU is used as an activation function. The architecture of the DD-DQN is explained in the 3.3.

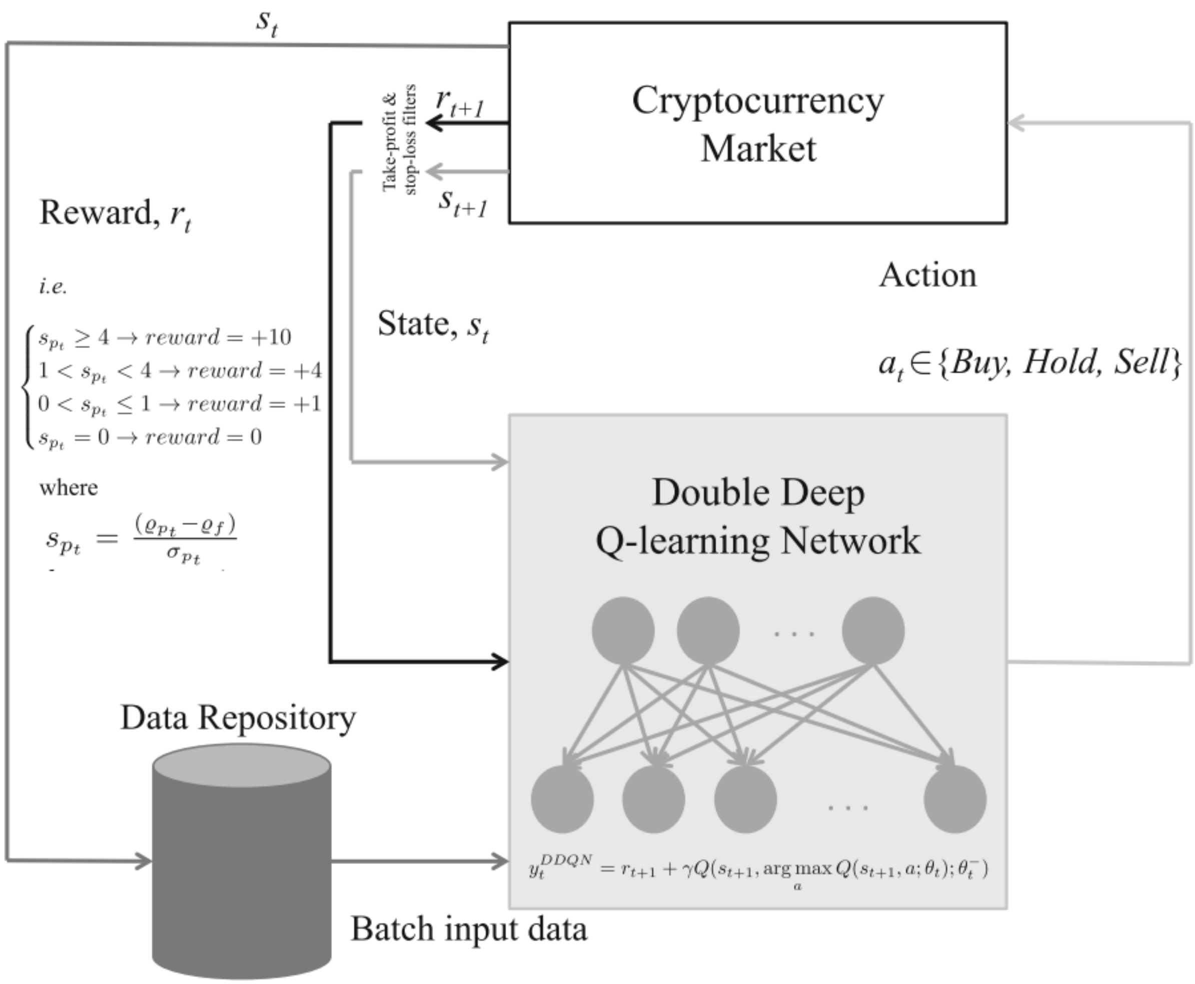
s_t
Cryptocurrency Market
Take-profit & stop-loss filters
r_{t+1}
s_{t+1}
Reward, r_t
i.e.
$\begin{cases} s_{p_t} \geq 4 \rightarrow reward = +10 \\ 1 < s_{p_t} < 4 \rightarrow reward = +4 \\ 0 < s_{p_t} \leq 1 \rightarrow reward = +1 \\ s_{p_t} = 0 \rightarrow reward = 0 \end{cases}$
where
$s_{p_t} = \frac{(\varrho_{p_t} - \varrho_f)}{\sigma_{p_t}}$
State, s_t
Action
$a_t \in \{Buy, Hold, Sell\}$
Double Deep Q-learning Network
Data Repository
Batch input data
$y_t^{DDQN} = r_{t+1} + \gamma Q(s_{t+1}, \arg\max_a Q(s_{t+1}, a; \theta_t); \theta_t^-)$

Figure 3.3: Double Deep Q-learning trading system with Sharpe reward function [45].

From Figure 3.4 and Table 3.5, we can see that DD-DQN and D-DQN trading systems outperform the simpler DQN system in both sharp and profit reward functions. These results also demonstrated that the Sharpe D-DQN is the best Q-learning trading system with an average cumulative return of 5.81% and 26.14% maximum and −5.64% minimum.

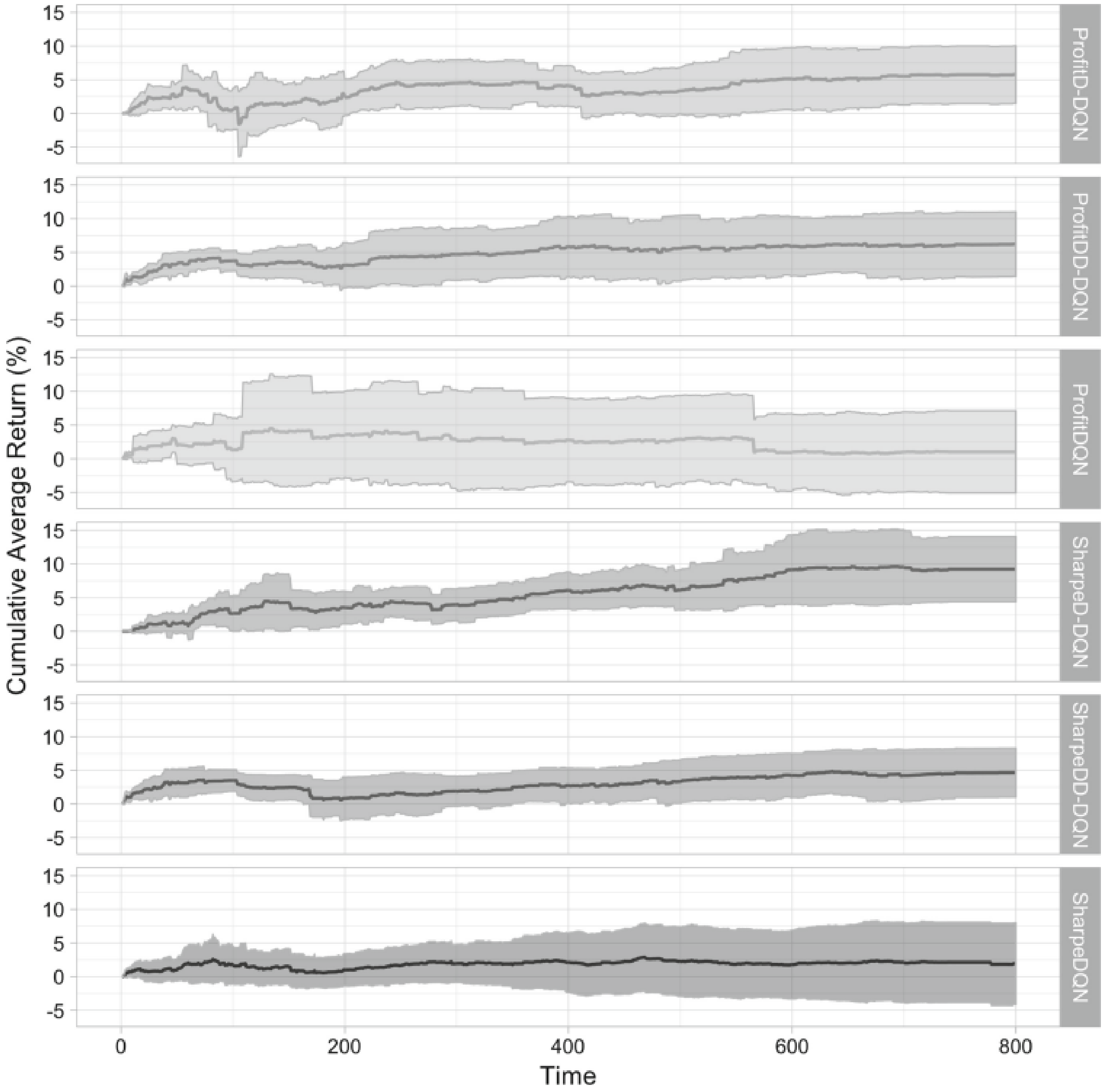

Figure 3.4: Average percentage returns over the 10 trading periods, i.e. different combinations of start and end dates for the trading activity [45].

Trading system	Avg. return (%)	Max. return (%)	Min. return (%)
Profit D-DQN	3.74	21.31	−10.74
Profit DD-DQN	4.85	17.34	−8.49
Profit DQN	2.32	22.59	−17.97
Sharpe D-DQN	5.81	26.14	−5.64
Sharpe DD-DQN	3.04	13.03	−8.49
Sharpe DQN	1.83	15.80	−9.29

Table 3.5: Average performance over the 10 trading periods [45].

- The author used two different reward functions: the risk measurement sharp-ration and a simple Profit function. The experiments revealed that sharp-ration was more performant, indicating that employing or integrating risk measurement in a reinforcement learning reward function may result in better performance.
- The author did include transaction costs which makes the trading environment more realistic.

Limitations:

- The agent was evaluated only on bitcoin currency, we can't make assumptions about the whole crypto-currency markets.
- No technical analysis indicators or any sort of preprocessing were used to eliminate the noise from the dataset (which has been shown in previous studies to significantly improve performance).

3.2 Action-only DRL

In this approach, the policy function is learned directly without the need for a value function which allows us to work on a continuous action space. Some disadvantage of this approach is the long time it takes the agent to learn to compare to critic-only methods as shown in this paper [82], and the reason behind this is that action-only methods require a big dataset to avoid the problem of bad actions considered good as long as the sum of all rewards is positive.

The first thing mentioned in the paper titled **”Deep Direct Reinforcement Learning for Financial Signal Representation and Trading”** [25], is this question: Can we train the computer to beat experienced traders for financial asset trading?

From the start, it was evident that the author’s goal was to outperform humans in trading. For that the author enumerated two main challenges to overcome:

- The difficulties of financial markets representation and summarization, the author also highlighted the fact that the financial dataset contains a lot of noise that makes it difficult to grasp and extract useful information from it.

- Second challenge is the dynamic behavior of trading action execution, the fact that some markets are so volatile that the price at the time of decision is not the same as the price at the time of transaction execution. Sometimes transaction expenses can wipe out all gains or, worse, result in a loss of money, so the challenge is how to model all this in a reinforcement learning environment.

To address the first challenge the author proposed the use of Deep learning for feature extraction and reducing the noise of the data by using fuzzy logic (assign linguistic values to input data) which in turn helps make the dataset more readable and easy to interpret by the RL agent as shown in the figure 3.5. Instead of using technical indicators, the author prefers to use the aforementioned deep learning model to extract features from the dataset.

To address the second challenge the author used a Recurrent Neural Network remodeled for an online environment and recurrent decision making, and he used an enhanced version of the backpropagation algorithm called task-aware BPTT to address the vanishing gradient problem. The agent was trained and tested on both the stock index (stock-IF contract) and commodity contracts the silver (AG) and sugar (SU) of 1-min time frame over a year. The agent was also tested on S&P-500 from 1990 to 2015 (the first 8 years for training and the rest 7 years for testing) obtained through yahoo finance. Transaction cost was set to 0.1% of the index value. The agent managed to make a profit but not so much considering the long period it was tested on (8 years) as shown in the figure 3.6.

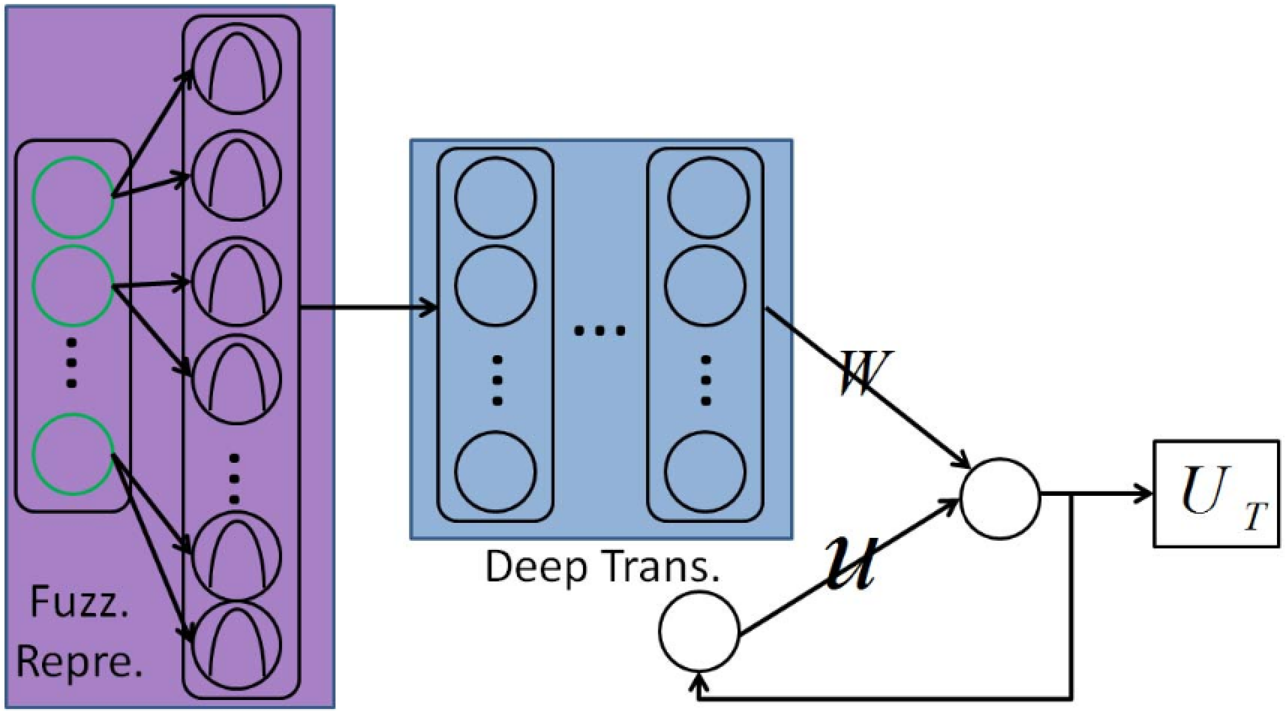

Figure 3.5: Overview of fuzzy DRNNs for robust feature learning and self-taught trading [25].

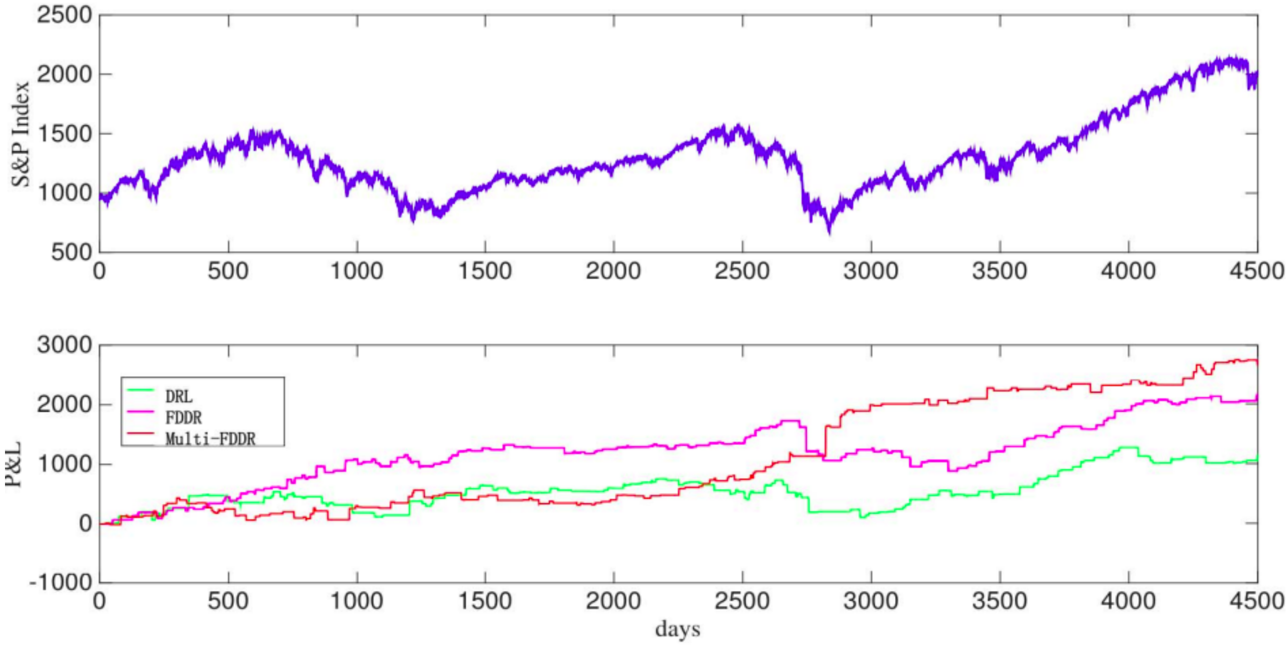

Figure 3.6: Testing S&P data and the P&L curves of different trading systems [25].

This was an interesting paper because it introduced a novel approach to financial data prepossessing, by combining deep-learning and fuzzy logic. The only drawback of this paper is there is no comparison with human performance in trading, despite stating at the beginning that the goal was to outperform humans in trading.

3.3 Actor-critic DRL

As we mentioned in the background chapter, actor-critic DRL combines both action-only and critic-only together. The actor learns the policy for taking action in a given state and the critic measures how good the action was. actor-critic algorithms like PG [71] or A2C [63] proved to be powerful and more effective than traditional reinforcement learning algorithms.

The paper titled **"Adaptive stock trading strategies with deep reinforcement learning methods"** [79], was reviewed in the critic-only section, as we stated before the paper contains two approaches: critic-only GDQN and action-critic GDPG which makes it a perfect candidate for comparing the two approaches. Same as before, the author used GRU to extract features from the raw dataset (OHLCV + technical indicators) and then train an agent using Policy gradient algorithms. The agent was trained and tested on the same dataset as before see table 3.3 for more information about the dataset. The architecture of the Deep learning models used is summarized in table 3.6.

Network	Layer	Outout Shape	Parameter
Actor network	GRU	(10, 32)	3240
	Dropout	(10,24)	0
	GRU	24	3528
	Dropout	24	0
	Dense	3	75
Critic network	Concatenate	23	0
	Dense	64	1536
	Dense	16	1040
	Dense	1	17

Table 3.6: The parameter settings of the actor network and critic network of GDPG [79].

The two algorithms (GDQN, GDPG) were tested and compared against Turtle trading Strategy. As shown in table 3.6 in most cases GDQN algorithm outperforms the turtle strategy and in turn GDPG was more performant and stable than GDQN algorithm.

		GDQN		GDPG		Turtle	
	Symbol	SR	R(%)	SR	R(%)	SR	R(%)
U.S. stock market.	AAPL	1.02	77.7	1.30	82	1.49	69.5
	GE	-0.13	-10.8	-0.22	-6.39	-0.64	-17.0
	AXP	0.39	20.0	0.51	24.3	0.67	25.6
	CSCO	0.31	20.6	0.57	13.6	0.12	-1.41
	IBM	0.07	4.63	0.05	2.55	-0.29	-11.7
U.K. stock market.	RDSB	0.35	26.8	0.79	22.7	0.36	10.9
	GSK	-0.04	-5.2	0.06	1.10	-0.26	-9.9
	ULVR	0.41	21.0	0.21	11.1	-0.16	-7.07
	HSBA	0.76	46.0	0.99	45.2	0.90	24.8
	BATS	0.14	19.1	0.03	37.0	0.51	18.5
Chinese stock market.	600519	1.79	171.0	1.90	215.9	0.4	15.1
	000001	0.07	2.46	0.07	1.99	-0.37	-10.5
	601288	0.86	36.8	0.96	37.1	1.20	30.4
	601988	0.62	40.1	1.01	41.6	0.30	6.62
	601398	3.47	96.6	2.60	103.3	2.45	94.1

Table 3.7: The GDQN, GDPG and Turtle Strategies in the U.S, U.K and Chinese stock market. [79].

.

The paper titled **"Stock Trading Bot Using Deep Reinforcement Learning"** [5], is a unique paper in this thesis because it is the only one that combines both news sentiment analysis with RCNN network and DRL agent for trading together. The agent is focused on swing trading (see more about swing trading in the background chapter). The architecture is fairly simple, composed of a two-part RCNN model predicting if the prices are going up or down and the output of this model is fed to the second part (DRL agent) as an input, DRL receives as an input the predicted price signal along with market closing price, technical indicator MA, the capital and the number of stocks held. The architecture is explained in greater detail in Figure 3.7.

For the first part (RCNN network), the author argues that recurrent layer is needed to captures the contextual information to a greater extent but RNN,

unlike CNN networks with max-pooling layers, are unable to recognize discriminating phrases in a text. Thus RCNN networks were used to benefit from both network types (CNN and RNN).
The input of this model is the stock prices and news headlines pre-processed and indexed. The model is composed of three layers embedding, convolutional, and finally, an LSTM layer, trained on 95947 news headlines of 3300 companies and validated on 31581 samples, resulting in an accuracy of 96.88% which proves stock value change can be predicted to be positive or negative.

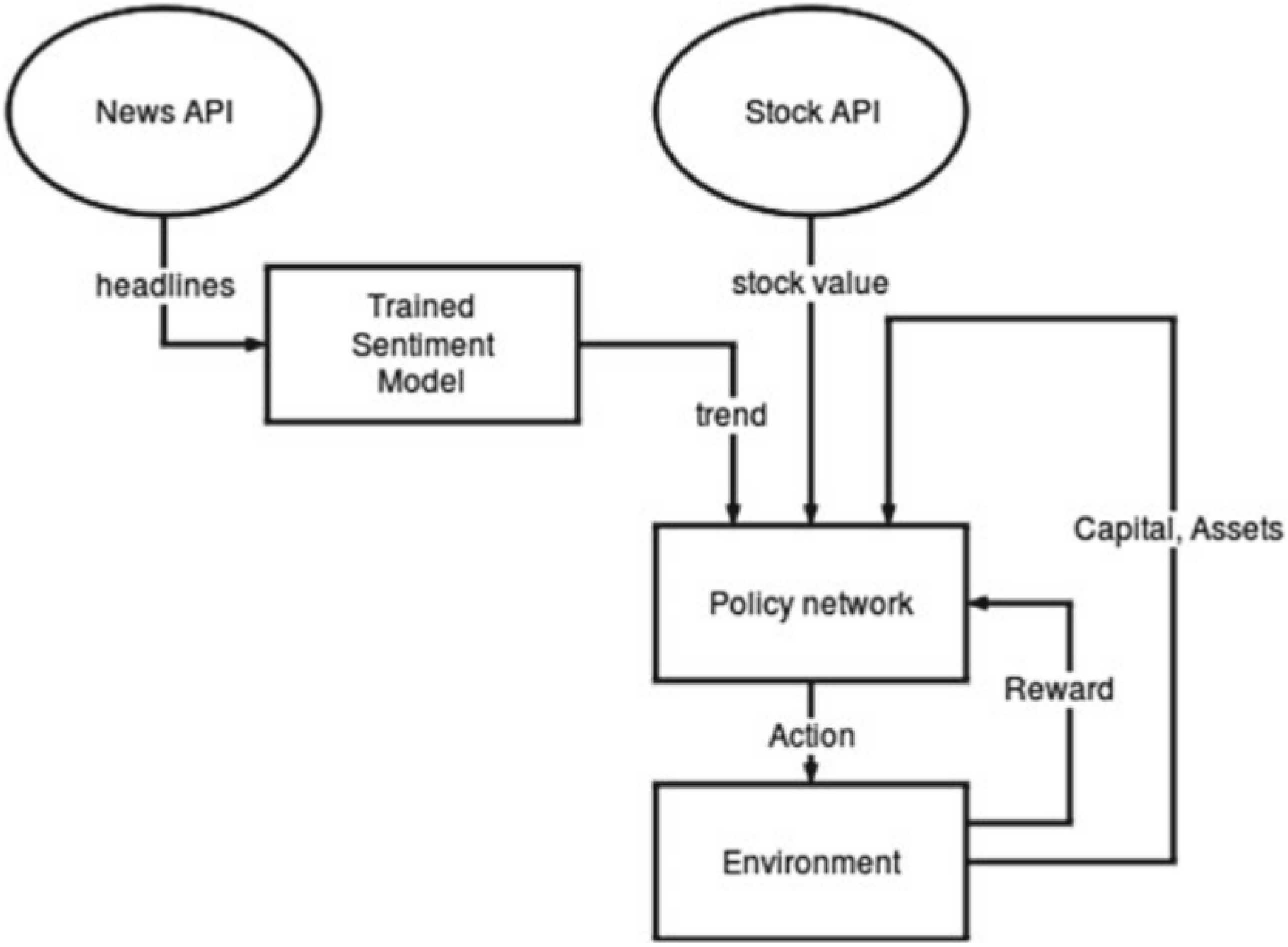

Figure 3.7: Overview of stock trading bot [5]

For the second part, the DDPG agent was trained on historical data (the author didn't state any information about the dataset) and experimented with 3 reward function Difference between RL agent asset value and stagnant asset value, Difference between the cost at which the stocks are sold and the cost at which the stocks were bought and binary reward representing if the action was profitable or not, the first two reward function failed to train the model (network gets stuck in the local minimal) were the binary reward performed the best.

The goal of this study was clear from the start: to demonstrate that sentiment analysis may be used to anticipate price direction and improve trading strategies; the goal was not to construct a fully working, ready-to-use system. This experiment lays the groundwork for further development and more advanced trading methods based on sentiment analysis.

Conclusion

Despite the fact that deep reinforcement learning is a relatively new branch, it has had a lot of success, particularly in the financial and trading worlds, thanks to the rise of deep learning and years of research into the fundamental theories of reinforcement learning. All of the studies discussed in this thesis pave the way for more research and experimentation that could transform the trading sector and lead this industry to become fully automated, from predicting asset prices to fully managing the portfolio to making critical trading decisions such as buying or selling assets. The research also showed that deep reinforcement learning could be applied in all markets (stocks, forex, cryptocurrencies, etc.) and with various strategies and periods (Day trading, Swing trading, etc.) with promising results that have a lot of room for improvement by combining and tweaking different techniques.

Bibliography

[1] Reinforcement learning : Markov-decision process towardsdatascience.com.

[2] Charu C. Aggarwal. *Neural Networks and Deep Learning*. Springer, Cham, 2018.

[3] Roohollah Amiri, Hani Mehrpouyan, Lex Fridman, Ranjan K Mallik, Arumugam Nallanathan, and David Matolak. A machine learning approach for power allocation in hetnets considering qos. In *2018 IEEE international conference on communications (ICC)*, pages 1–7. IEEE, 2018.

[4] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[5] Akhil Raj Azhikodan, Anvitha GK Bhat, and Mamatha V Jadhav. Stock trading bot using deep reinforcement learning. In *Innovations in Computer Science and Engineering*, pages 41–49. Springer, 2019.

[6] Jamil Baz, Nicolas Granger, Campbell R Harvey, Nicolas Le Roux, and Sandy Rattray. Dissecting investment strategies in the cross section and time series. *Available at SSRN 2695101*, 2015.

[7] Brian Beers. Buy and hold definition. https://www.investopedia.com/terms/b/buyandhold.asp.

[8] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[9] biocompare. Molecular details of brain injury revealed.

[10] Yash Bohra. The challenge of vanishing/exploding gradients in deep neural networks.

[11] Thomas J. Brock. Value at risk (var). https://www.investopedia.com/terms/v/var.asp.

[12] Seok-Jun Bu and Sung-Bae Cho. Learning optimal q-function using deep boltzmann machine for reliable trading of cryptocurrency. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 468–480. Springer, 2018.

[13] capitalindex team. Different types of trading strategies. https://www.capitalindex.com/bs/eng/pages/trading-guides/different-types-of-trading-strategies/.

[14] John Carpenter. A guide to company shares. https://www.1stformations.co.uk/blog/a-guide-to-company-shares/.

[15] Michael Carr. Turtle trading: A market legend. https://www.investopedia.com/articles/trading/08/turtle-trading.asp.

[16] Souradeep Chakraborty. Capturing financial markets to apply deep reinforcement learning. *arXiv preprint arXiv:1907.04373*, 2019.

[17] James Chen. Foreign exchange (forex). https://www.investopedia.com/terms/f/foreign-exchange.asp.

[18] James Chen. Forex market. https://www.investopedia.com/terms/forex/f/forex-market.asp.

[19] Lin Chen and Qiang Gao. Application of deep reinforcement learning on automated stock trading. In *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, pages 29–33. IEEE, 2019.

[20] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[21] R.W. Colby and T.A. Meyers. *The Encyclopedia of Technical Market Indicators*. Dow Jones-Irwin, 1988.

[22] B.J. Copeland. artificial intelligence. https://www.britannica.com/technology/artificial-intelligence/.

[23] corporate finance institute team. Financial markets.

[24] Pamela de la Fuente Dayana Yochim, Dayana Yochim. A guide to company shares. https://www.nerdwallet.com/article/investing/stock-trading-how-to-begin.

[25] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. Deep direct reinforcement learning for financial signal representation and trading. *IEEE transactions on neural networks and learning systems*, 28(3):653–664, 2016.

[26] Sebastian Dittert. Reinforcement learning: Value function and policy medium.com.

[27] Corporate finance institute. Profit and loss statement (p&l). https://corporatefinanceinstitute.com/resources/knowledge/accounting/profit-and-loss-statement-pl/.

[28] Yuan Gao, Ziming Gao, Yi Hu, Sifan Song, Zhengyong Jiang, and Jionglong Su. A framework of hierarchical deep q-network for portfolio management. In *ICAART (2)*, pages 132–140, 2021.

[29] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[30] Petter Kowalik Gran, August Jacob Kjellevold Holm, and Stian Gropen Søgård. A deep reinforcement learning approach to stock trading. Master's thesis, NTNU, 2019.

[31] Kevin Gurney. *An Introduction to Neural Networks*. Taylor and Francis, Inc., USA, 1997.

[32] Ikhlaas Gurrib et al. Performance of the average directional index as a market timing tool for the most actively traded usd based currency pairs. *Banks and Bank Systems*, 13(3):58–70, 2018.

[33] Adam Hayes. How does the stock market work? https://www.investopedia.com/articles/investing/082614/how-stock-market-works.asp.

[34] Adam Hayes. Physical asset. https://www.investopedia.com/terms/p/physicalasset.asp.

[35] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[36] Zexin Hu, Yiqi Zhao, and Matloob Khushi. A survey of forex and stock price prediction using deep learning. *Applied System Innovation*, 4(1):9, 2021.

[37] Chien Yi Huang. Financial trading as a game: A deep reinforcement learning approach. *arXiv preprint arXiv:1807.02787*, 2018.

[38] Zhengyao Jiang, Dixing Xu, and Jinjun Liang. A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059*, 2017.

[39] Debasish Kalita. A brief overview of deep reinforcement learning.

[40] Jae Won Lee. Stock price prediction using reinforcement learning. In *ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No. 01TH8570)*, volume 1, pages 690–695. IEEE, 2001.

[41] JoonBum Leem and Ha Young Kim. Action-specialized expert ensemble trading system with extended discrete action space using deep reinforcement learning. *PloS one*, 15(7):e0236178, 2020.

[42] Yuxi Li. Deep reinforcement learning: An overview. *CoRR*, abs/1701.07274, 2017.

[43] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[44] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017.

[45] Giorgio Lucarelli and Matteo Borrotti. A deep reinforcement learning approach for automated cryptocurrency trading. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 247–258. Springer, 2019.

[46] Giorgio Lucarelli and Matteo Borrotti. A deep q-learning portfolio management framework for the cryptocurrency market. *Neural Computing and Applications*, 32(23):17229–17244, 2020.

[47] Wei Luo, Qirong Tang, Changhong Fu, and Peter Eberhard. Deep-sarsa based multi-uav path planning and obstacle avoidance in a dynamic environment. In *International Conference on Swarm Intelligence*, pages 102–111. Springer, 2018.

[48] Mansoor Maitah, Petr Procházka, Michal Cermak, and Karel Šrédl. Commodity channel index: Evaluation of trading rule of agricultural commodities. *International Journal of Economics and Financial Issues*, 6(1):176–178, 2016.

[49] Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search provides a competitive approach to reinforcement learning. *arXiv preprint arXiv:1803.07055*, 2018.

[50] J.B. Maverick. The difference between the sharpe ratio and the sortino ratio. https://www.investopedia.com/ask/answers/010815/what-difference-between-sharpe-ratio-and-sortino-ratio.asp.

[51] Adrian Millea. Deep reinforcement learning for trading; a critical survey. *Data*, 6(11), 2021.

[52] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[53] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[54] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260, 2008.

[55] nirmalbang team. Trading vs. investing. https://www.nirmalbang.com/knowledge-center/trading-and-investing.html.

[56] Christopher Olah. Understanding lstm networks.

[57] Kayur Patel, James Fogarty, James A Landay, and Beverly Harrison. Investigating statistical machine learning as a tool for software development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 667–676, 2008.

[58] David Petersson. supervised learning. https://www.techtarget.com/searchenterpriseai/definition/supervised-learning/.

[59] Mary K. Pratt. unsupervised learning. https://www.techtarget.com/searchenterpriseai/definition/unsupervised-learning/.

[60] Tidor-Vlad Pricope. Deep reinforcement learning in quantitative algorithmic trading: A review. *CoRR*, abs/2106.00123, 2021.

[61] Jonathan Sadighian. Extending deep reinforcement learning frameworks in cryptocurrency market making. *arXiv preprint arXiv:2004.06985*, 2020.

[62] Yoshiharu Sato. Model-free reinforcement learning for financial portfolios: a brief survey. *arXiv preprint arXiv:1904.04973*, 2019.

[63] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[64] Sreelekshmy Selvin, R Vinayakumar, EA Gopalakrishnan, Vijay Krishna Menon, and KP Soman. Stock price prediction using lstm, rnn and cnn-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)*, pages 1643–1647. IEEE, 2017.

[65] Semyon Sergunin. How rpa and machine learning address business use cases. https://www.automationanywhere.com/company/blog/rpa-thought-leadership/how-rpa-and-machine-learning-address-business-use-cases/.

[66] William F Sharpe. Mutual fund performance. *The Journal of business*, 39(1):119–138, 1966.

[67] Hong-Gi Shin, Ilkyeun Ra, and Yong-Hoon Choi. A deep multimodal reinforcement learning system combined with cnn and lstm for stock trading. In *2019 International conference on information and communication technology convergence (ICTC)*, pages 7–11. IEEE, 2019.

[68] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. PMLR, 2014.

[69] smartera3s. Customer micro segmentation. https://www.smartera3s.com/products/customer-segmentation/.

[70] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, zweite edition, 2018.

[71] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[72] Mehran Taghian, Ahmad Asadi, and Reza Safabakhsh. Learning financial asset-specific trading rules via deep reinforcement learning. *Expert Systems with Applications*, page 116523, 2022.

[73] THE INVESTOPEDIA TEAM. 7 technical indicators to build a trading toolkit. https://www.investopedia.com/top-7-technical-analysis-tools-4773275/.

[74] The Editors of Encyclopedia Britannica. *stock exchange*. September 2021.

[75] Mike Thomas. How ai trading technology is making stock market investors smarter. https://builtin.com/artificial-intelligence/ai-trading-stock-market-tech/.

[76] William Wai Him Tsang, Terence Tai Leung Chong, et al. Profitability of the on-balance volume indicator. *Economics Bulletin*, 29(3):2424–2431, 2009.

[77] Rundong Wang, Hongxin Wei, Bo An, Zhouyan Feng, and Jun Yao. Commission fee is not enough: A hierarchical reinforced framework for portfolio management. *arXiv preprint arXiv:2012.12620*, 2020.

[78] J Welles Wilder. *New concepts in technical trading systems*. Trend Research, 1978.

[79] Xing Wu, Haolei Chen, Jianjia Wang, Luigi Troiano, Vincenzo Loia, and Hamido Fujita. Adaptive stock trading strategies with deep reinforcement learning methods. *Information Sciences*, 538:142–158, 2020.

[80] Xing Wu, Haolei Chen, Jianjia Wang, Luigi Troiano, Vincenzo Loia, and Hamido Fujita. Adaptive stock trading strategies with deep reinforcement learning methods. *Information Sciences*, 538:142–158, 2020.

[81] Pengqian Yu, Joon Sern Lee, Ilya Kulyatin, Zekun Shi, and Sakyasingha Dasgupta. Model-based deep reinforcement learning for dynamic portfolio optimization. *arXiv preprint arXiv:1901.08740*, 2019.

[82] Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deep reinforcement learning for trading. *The Journal of Financial Data Science*, 2(2):25–40, 2020.

All links in the bibliography were accessed between 2022-02 and 2022-06; if any of the links are dead, you can access them using the web archive at https://web.archive.org