

Neuronales Netz Portfolio Optimierung

Durch die Sharpe-Quotient

by

Patrick Spohr

09. Februar 2024

Code

<https://github.com/p-spohr/NN-Portfolio-Optimizer>

Masterstudiengang

Finanzmathematik, Aktuarwissenschaften und Risikomanagement

Fachbereich

Informatik, Kommunikation und Wirtschaft

Lehrveranstaltung

Statistical Learning in Finance and Insurance

Professorin

Dr. Christina Erlwein-Sayer

Inhalt

1	Motivation	2
1.1	Portfolio Optimierung	2
2	Literatur	3
2.1	Definitionen	3
2.2	Literaturüberblick	3
2.3	Hauptliteratur	4
3	Ziele	5
3.1	Hauptziele	5
3.2	Grenzen und Vermutungen	5
4	Daten	6
4.1	Daten und Quellen	6
4.2	Überblick der zehn beliebigen Aktien	6
5	Methoden	8
5.1	Künstliches Neuronales Netz	8
5.2	Long Short-Term Memory	9
5.3	Softmax-Funktion	10
5.4	Sharpe-Quotient	11
5.5	Struktur der zwei Modelle	11
5.6	Python und Keras	12
5.7	Monatliche Portfoliowerte	12
6	Ergebnisse	14
6.1	Portfoliowerte und Renditen von 1990 bis 2020	14
6.2	Sharpe-Quotienten	17
6.3	Fazit und Weiterentwicklung	18
	Literaturverzeichnis	19

1 Motivation

1.1 Portfolio Optimierung

Stellen Sie sich eine Situation vor, wobei man vielversprechende Wertpapiere fand, aber er weiß nicht, wie sie in Portfolio zugeteilt werden dürfen. Markowitz [11] zufolge habe das Verfahren, um Portfolio aufzubauen, zwei Schritte. Am Anfang würden die einzelnen Wertpapiere geprüft, indem man Erfahrungen und Beobachtungen erwäge. Danach baue man mit dieser Information ein Portfolio auf. Die einfachste Lösung ist die Wertpapiere mit einem Gleichgewicht versehen und alles unter Fach und Dach zu bringen und warum nicht, die Wertpapiere sind gut, aber das ändert das Portfolio um ein Vielfaches. Die Volatilität jeder neuen Wertpapiere trägt zu dem Portfolio bei und anschließend würde das Risiko des Portfolios erhöhen oder verringern und das könnte die Wünsche der Anleger widersprechen. In Modern Portfolio Theory (MPT) sollte die steigende Volatilität eine entsprechende Rendite erwirtschaften. Studien zeigten, dass 90 Prozent der Renditen eines Portfolios dem 'Allocation Policy' zuschreiben würde [14].

Nun komme ich zu dem Hauptteil meiner Untersuchung. Mit dem zweiten Schritt von Markowitz fange ich an, nämlich, geprüfte Wertpapiere in einem Portfolio zuzuteilen. Zuerst schränke ich die Auswahl der Wertpapiere um US-Aktien ein und dann sollten sie zugeteilt werden, indem ein künstliches neuronales Netz und ein Long Short-Term Memory Layer mit der Sharpe-Quotient als Verlustfunktion eingesetzt wird. Lassen Sie mich diese Begriffe und das Modell, mit dem ich die Gewichte der Aktien gerechnet habe, in der Überschrift 'Methoden' erklären.

Im Fazit, um die Leistungen die Portfolios zu verstehen, werden die optimierten Portfolios mit einem gleichgewichteten Portfolio und einem Vergleichsindex verglichen, weil die Leistung eines Portfolios ohne einen Maßstab keine konkrete Folgerung ergibt.

2 Literatur

2.1 Definitionen

Als Einleitung definiere ich die Hauptbegriffe und deren Kurzwörter. Künstliches Neuronales Netz oder KNN sind nicht-lineare Algorithmen, die den Neuronen im Gehirn nachempfunden sind. Die Gestalt aller KNN hat eine Eingangsschicht, eine oder mehrere verborgenen Schichten und eine Ausgangsschicht [3]. Die verborgene Schicht in meinem KNN heißt Long Short-Term Memory oder LSTM. Man bezeichnet LSTM als eine Technik zur Verbesserung eines rekurrenten neuronalen Netzes, indem es um eine Art Erinnerung an frühere Erfahrungen verfügt, um das Problem des verschwindenden Gradienten zu lindern [9]. Die Verlustfunktion im KNN misst den Fehler zwischen einer Beobachtung und einer Schätzung einer Regression oder einer Klassifikation. Der Sharpe-Quotient, den ich als Verlustfunktion benutzte, ist eine Kennzahl, die die Rendite zur Volatilität eines Portfolios zeigt [15]. Zuletzt ist der risikofreie Zinssatz (risk-free rate, RFR) oder RFZ ein Referenzzinssatz, der in den Sharpe-Quotient eingesetzt wird und theoretisch kein Ausfallrisiko enthält. Es gibt einige RFZ, die zur Verfügung stehen. Ich erläutere später die, die ich benutzte und warum.

2.2 Literaturüberblick

Ich bespreche kurz die Literatur, von der mein Projekt abgeleitet wurde. Die Hauptliteratur, von der ich die Idee und das Modell weiterentwickelte, heißt "Deep Learning for Portfolio Optimization" [18] von Zhang, Zohren und Roberts. Ich erkläre diese Studie ausführlicher im nächsten Teil, denn es gibt Unterschiede, die ich mit meinem Projekt gegenüberstellen will. TensorFlow [1] und dessen Keras API [5] ermöglichten das künstliche neuronale Netz und gab mir die Gelegenheit alles schnell zu erschaffen. Andere Literatur war hilfreich alles zu verstehen, besonders mit den Begriffen künstliches neuronales Netz [3] und Long Short-Term Memory [9] von Hochreiter und Schmidhuber, die das Verfahren initiierten.

2.3 Hauptliteratur

Im Anschluss daran weise ich auf die wichtigen Unterschiede zwischen meinem Projekt und der Studie von Zhang et al. [18] hin. Der größte Unterschied war die Auswahl von Wertpapieren, weil sie Indexfonds oder Exchange-Traded Funds (ETFs) statt Aktien auswählten. Ein Portfolio mit ETFs von Aktien und Anleihen zeigt meines Erachtens die Kraft von diesem KNN-Modell nicht, weil ETFs schon relativ diversifiziert und nicht volatil sind. Ich wollte sehen, wie ein langfristiges Portfolio von einzelnen Aktien, rücksichtslos ihre erwarteten Renditen, optimiert wird. Populärer Indexfonds erwirtschaftet in der Regel positive Rendite, denn sie den Markt folgen und wirken als eine sehr konservative Probe in diesem Experiment.

Als Nächstes stehe ich die Sharpe-Quotient im Fokus, die sie als Verlustfunktion anwendeten. Sie hatten keinen risikofreien Zinssatz in ihrem Modell benutzt, während ich ein Modell mit und ohne einen aufbaute. Der RFZ ist ein wichtiger Teil der Sharpe-Quotient und kann verschiedene Ergebnisse bewirken, wie in der Überschrift 'Ergebnisse' dargestellt wird. In Hinsicht auf die Ergebnisse gibt es einen anderen Unterschied, den ich erwähnen muss, weil ich keine Handelskosten betrachtete. Das lässt sich als eine wichtige Auslassung zeigen, denn die Aktien im ganzen Portfolio mussten regelmäßig komplett zugeteilt. Wegen dieser Berücksichtigung wurde zumindest ein Monat abgewartet, bis die Gewichte des Portfolios wieder optimiert werden durften, weil tägliches oder wöchentliches Trading zu viel gekostet hätte.

3 Ziele

3.1 Hauptziele

Im Folgenden gehe ich auf die Hauptziele meines Projektes. Was hatte ich am Ende vor, zu erzielen? Am wichtigsten werden Portfolios von beliebigen Aktien durch die Sharpe-Quotient mit und ohne einen RFZ optimiert. Das heißt genau, dass ich die Gewichte jeder Aktie für einen Zeitraum erzeuge. Dadurch, dass ich die Portfoliowerte bekomme, multipliziere ich die Gewichte und Aktienkurs. Danach werden die Leistung von den optimierten Portfolios mit einem gleichgewichtigen Portfolio und einem Vergleichsindex verglichen. 30 Jahre über kann ich die monatlichen Renditen jedes Portfolio rechnen, um die Sharpe-Quotient jedes Portfolio auszurechnen. Ich hatte am Anfang keine Annahme, weil mein Projekt weit weg genug von der Hauptliteratur [18] war, dass ich nicht wusste, ob die optimierten Portfolios besser als ein gleichgewichtiges Portfolio oder ein Vergleichsindex würden sein. Im besten Fall würden zumindest die optimierten Portfolios dem gleichgewichtigen Portfolio überlegen sein.

3.2 Grenzen und Vermutungen

Wie ich gerade erwähnt habe, wird das Portfolio nur mit Aktien aufgebaut. Angesichts dieser Bedingung schränke ich den Umfang des Portfolios weiter, um zu verständlichen Schlüssen zu führen. Zuerst kommen alle Aktien aus derselben Börse, NASDAQ, also nur US-Aktien. Zweitens werden die Aktien beliebig ausgewählt, die im Zeitraum von 1990 bis 2019 stetig gehandelt wurden. Von diesen Grenzen folgen die Vermutungen, dass alle Aktien schon geprüft wurden und zumindest zahlungsfähig waren, weil sie für 30 Jahre auf dem Markt gehandelt wurden. Konkret bedeutet, dass die Analysten in der Lage waren, Unternehmen zu vermeiden, die bankrottgegangen waren. Das ist wichtig, weil das Modell nur funktioniert, wenn die Form der Daten gleich sind oder anders gesagt, die Anzahl der Tage auf dem Markt gleich sind. Außerdem wählte ich US-Treasury Staatsanleihen als einen RFZ, weil ich nur US-Aktien haben.

4 Daten

4.1 Daten und Quellen

Ich gehe nun über zu den Daten und beschreibe, woher ich die Daten bekomme und was ich aus den Daten auszog. Zusammenfassend hatte ich drei Datenquellen: Kaggle, Yahoo! Finance, und FRED, der für Federal Reserve Economic Data steht. Ich erhielt den täglichen Aktienpreis und Datum aus Stock Market Dataset auf Kaggle [13]. Der Vergleichsindex ist der S&P 500 und kommt aus Yahoo! Finance [6]. Es ist zum Bemerken, dass der Ticker GSPC heißt. Aus dem GSPC erhielt ich auch den täglichen Indexpreis und Datum. Zuletzt bekam ich von FRED [7] den risikofreien Zinssatz – ‘Market Yield on U.S. Treasury Securities at 1-Month Constant Maturity.’ Hier könnten andere RFZ angewendet werden, und vielleicht hätte ein anderer besser Ergebnisse ergeben, aber ich bespreche diesen Punkt im Fazit. Der RFZ hat ein Datum und ein tägliches Prozent, aber seltsamerweise gab es paar Lücken im Datensatz von FRED und ich musste deswegen die füllen, indem ich den Durchschnitt von dem frühen und nächsten Prozent rechnete. Ich denke, das beeinflusst die Ergebnisse nicht viel, weil es sich über 30 Jahren erstrecken, aber ich mitteile es trotzdem.

4.2 Überblick der zehn beliebigen Aktien

Bei den Aktien steht die Frage im Mittelpunkt, ob sie wirklich ein typisches oder realistisches Portfolio ähneln. Obwohl sie beliebig ausgewählt wurden, fand ich es wichtig, die relevante Information über sie auszurechnen, weil es passieren könnte, dass das Portfolio, egal von den Gewichten, den S&P 500 komplett übertrifft, wenn es Ausreißer wie Monster Beverage Corp (MNST) enthält, der über 162.590 Prozent Gesamtrendite von 1990 bis 2019 einbrachte. Außerdem ist die Summe aller monatlichen Renditen 1.144 Prozent.

Aus aller beliebig ausgewählten Aktien ist SkyWest, Inc. am besten, die die Summe aller monatlichen Renditen 635,7 Prozent erwirtschaftete und eine durchschnittliche monatliche Sharpe-Quotient 0,825 hatte. Alle Aktien im Portfolio hatten eine durchschnittliche monatliche Sharpe-Quotient zwischen null und eins, also es gibt keine außergewöhnliche Aktie. Die Aktien gehörten zu verschiedenen Industrien und deswegen ist das Portfolio diversifiziert in

dieser Hinsicht. Die Aktien in Table 1 bauten das Portfolio auf, das ich im Modell eingesetzt hatte.

Ticker	Name	Geschäft
COHR	Coherent, Inc.	Laser
CTB	Cooper Tire & Rubber Company	Reifen
EQT	EQT Corporation	Energie
GOLD	Barrick Gold Corporation	Bergarbeit
NSEC	National Security Group, Inc.	Versicherung
OTTR	Otter Tail Corporation	Energie
PEP	PepsiCo, Inc.	Getränke
SKYW	SkyWest, Inc.	Fluglinie
SNFCA	Security National Financial Corporation	Lebensversicherung
WY	Weyerhaeuser Company	Abholzung

Table 1: Aktien des Portfolios

5 Methoden

5.1 Künstliches Neuronales Netz

Dieser Abschnitt behandelt die wichtigen Teile eines künstlichen neuronalen Netzes, besonders ein Feed-Forward KNN. Das KNN hat drei Teile: eine Eingangsschicht x_1, \dots, x_d mit d Inputs, eine oder mehrere verborgenen Schichten, z_1, \dots, z_M mit M Hidden-Units, und eine Ausgangsschicht y_1, \dots, y_c mit c Output-Units.

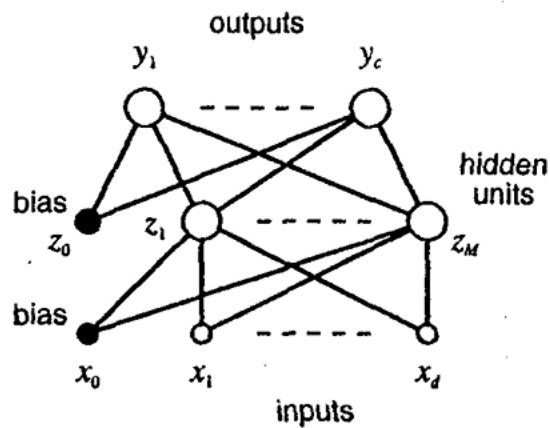


Figure 4.1. An example of a feed-forward network having two layers of adaptive weights. The bias parameters in the first layer are shown as weights from an extra input having a fixed value of $x_0 = 1$. Similarly, the bias parameters in the second layer are shown as weights from an extra hidden unit, with activation again fixed at $z_0 = 1$.

Figure 1: Feed-Forward Neuronales Netz [3]

Daten fließen in der Eingangsschicht, wo die Eingabe und die Gewichte jeder Neuronenverbindung multipliziert werden, um eine Ausgabewert zu ergeben. Für jedes Neuron wird der Aktivierungszustand durch eine Aktivierungsfunktion mit dessen Eingabewert bestimmt, sodass ein Neuron feuert oder ruht. Die Ausgabe dieser Aktivierungsfunktion geht weiter in den nächsten Schichten und wie vor, dieselben Verfahren gebrauchen, bis die Ausgangsschicht, wobei der Fehler der Schätzung durch eine Verlustfunktion gerechnet wird. Dann fängt Back-Propagation an, wo, von hinten nach vorne, die Gewichte jeder Neuronenverbindung mit

Hilfe des Fehlers durch ein Gradientenverfahren aktualisiert werden. Diese Feed-Forward und Back-Propagation Verfahren wiederholen sich, bis der Fehler minimisiert ist. Die Training-Genauigkeit sollte in der Regel nicht viel größer als die Validation-Genauigkeit, weil das Over-Fitting bedeuten könnte und weist darauf, dass das Modell dem Geräusch oder der Varianz folgt.

5.2 Long Short-Term Memory

Die verborgene Schicht meines Modelles ist LSTM und ich erkläre die Wichtigkeit dieser Weiterentwicklung des KNN-Verfahrens. Lassen Sie mich zuerst kurz die Schwäche vom traditionellen Feed-Forward Verfahren hervorheben. Feed-Forward KNN hat sogenannte kurze Erinnerung oder 'Short-Term Memory,' weil die Neuronen keine Informationen über frühere Neuronen verfügen. Normalerweise ist das kein Problem, weil die frühere Information keinen großen Einfluss auf die nächste Information haben. Das ändert sich, wenn Zeitreihen als Eingangsdaten einfließen, weil die früheren Daten wichtige Information enthalten, die die Ergebnisse verbessern können. Zum Beispiel sind Aktienkurse Zeitreihen und die früheren Preise beeinflussen die nächsten.

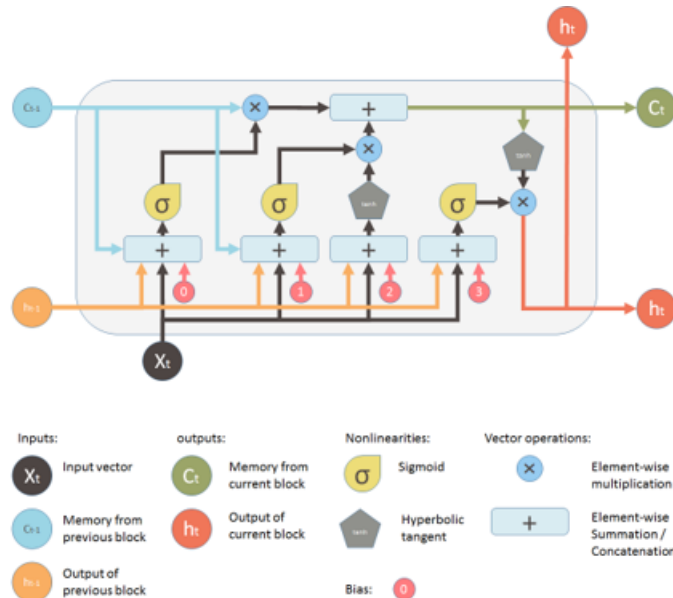


Figure 2: Long Short-Term Memory Unit [17]

Kurz gesagt, LSTM wurde gebildet, um Back-Flow-Problemen von Feed-Forward KNN 'Short-Term Memory' zu lindern. Dadurch, dass ich das beste Verfahren für meine Daten benutzen will, wendete ich, wie meine Hauptliteratur [18], LSTM an. Ich bemerke auch, dass LSTM eine Art rekurrentes neuronales Netz oder RNN ist, weil die Ausgaben der früheren Neuronen erinnert werden. Im RNN heißt Back-Propagation unter einen anderen Namen: 'Back-Propagation Through Time' oder BPTT. Der Grundidee bleibt wie vor, aber das Verfahren nutzt eine Reihe von Zeitschritten mit Eingaben und Ausgaben Paare und rechnet die kumulierten Fehler über jeden Zeitschritt [4].

5.3 Softmax-Funktion

Wie früher erwähnt hat KNN eine Aktivierungsfunktion. Mein Modell benutzte die Softmax-Funktion, die eine normalisierte Exponentialfunktion ist.

$$\sigma(x)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad \text{für } j = 1, \dots, K \text{ und } x = \text{Input Vektor}$$

Die Softmax-Funktion wird eingesetzt, um eine Wahrscheinlichkeitsverteilung über alle unterschiedlichen möglichen Ereignisse zu rechnen. Die Funktion versichert für Klassifikation mit mehr als zwei Klassen, dass die Wahrscheinlichkeit aller Klassen eins ergibt und das ist wichtig für die Gewichte der Aktien im Portfolio, weil die Summe aller Gewichte auch eins ist. Es ist auch sinnvoll zu vermerken, dass die Softmax-Funktion der Sigmoid-Funktion ähnelt aber die Summe aller Werte in der Sigmoid-Funktion nicht eins ist.

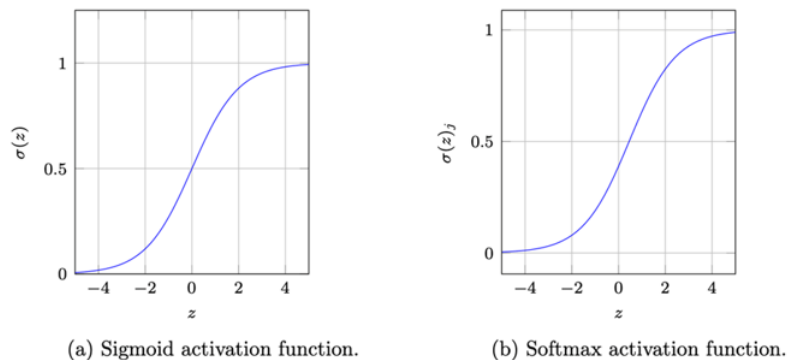


Figure 3: Long Short-Term Memory Unit [2]

5.4 Sharpe-Quotient

Anschließend stelle ich die Sharpe-Quotient vor, die als Verlustfunktion verwendet wird. Die Sharpe-Quotient enthält drei Teile: die Rendite des Portfolios, ein risikofrier Zinssatz und die Volatilität oder die Standardabweichung der Renditen. Was sagt die Sharpe-Quotient genau? Eine Sharpe-Quotient über eins bedeutet, dass das Portfolio eine Rendite mehr als der risikofreie Zinssatz erwirtschaftet und deren Ergebnis größer als die Volatilität der Rendite ist. Einfacherweise wird der Anleger für das Risiko hervorragend entschädigt. Eine Sharpe-Quotient gleich eins zeigt, dass Renditen und Risiken in einem ausgewogenen Verhältnis stehen. Unter eins zeigt eine unterdurchschnittliche Entschädigung für das entsprechende Risiko [15].

$$Sharpe = \frac{R_p - R_f}{\sigma_p}$$

R_p = Portfolio Renditen
 R_f = risikofreier Zinssatz
 σ_p = Standardabweichung der Renditen

In meinem Model ist A Aktien mit x Close Preis die Features und das Y Target ist das y Gewicht für jede N Aktie. Typischerweise würde die Verlustfunktion minimiert aber die Sharpe-Quotient in meinem Modell wird negiert, weil das Minimum von einer negierten Funktion das Maximum ist. Das Maximum von der Sharpe-Quotient bedeutet, die größte Rendite für die entsprechende Volatilität des Portfolios.

5.5 Struktur der zwei Modelle

Ich bespreche kurz die überall Struktur der zwei Modelle, die ich benutzte, um das monatliche Gewicht für jede Aktie zu bestimmen. Das Struktur ist in Figure 4 dargestellt. Das erste Modell bleibt ungeändert von dem Modell in der Hauptliteratur [18]. Das bedeutet, dass das Modell ein RNN mit einer LSTM-Schicht mit 64 Einheiten ist. Dieses erste Modell setzt keinen risikofreien Zinssatz in die Sharpe-Quotient-Verlustfunktion ein. Das zweite Modell beachtet den risikofreien Zinssatz, um der optimalen Gewichte zu generieren. Die Eingangsschicht enthält die täglichen Preise und tägliche prozentuelle Änderung jede Aktie. Am Ende ergibt das Modell das monatliche Gewicht für jede Aktie, die die Sharpe-Quotient

optimiert.

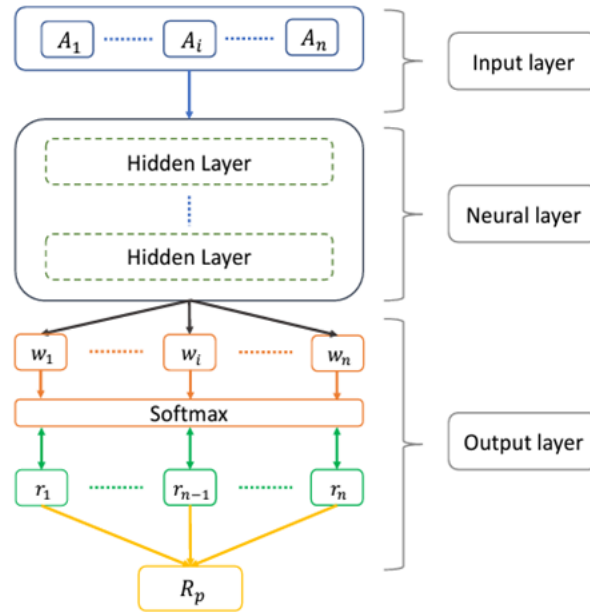


Figure 2: Model architecture schematic. Overall, our model contains three main building blocks: input layer, neural layer and output layer.

Figure 4: Struktur des Modelles [18]

5.6 Python und Keras

Um die Reproduzierbarkeit des Experiments zu dürfen, liste ich alle notwendige Anwendungen und Versionen in Table 2 auf. NumPy [8] ist wichtig für schnelle Arbeit mit Arrays und ist mit pandas [16] [12] verbunden, um DataFrames zu erzeugen und bearbeiten. Matplotlib [10] lässt man einfach Grafik schaffen. Auf TensorFlow [1] baut sich die API Keras [5] auf, damit man Python anwenden kann.

5.7 Monatliche Portfoliowerte

Um die Leistung dieses Modell zu schätzen, wurde eine Simulation durchgeführt, wo man diese erzeugten Gewichte benutzen, um die Sharpe-Quotient des Portfolios zu optimieren und hohe Renditen zu erwirtschaften. Die monatliche Portfoliowerte wurde gerechnet, in-

Library	Versionen
python	3.10.13
tensorflow	2.10.0
numpy	1.26.0
pandas	2.1.1
matplotlib	3.8.0

Table 2: Python und Library Versionen

dem die optimalen Gewichte für jede Aktie im Portfolio für Zeit t , wurde mit den Aktienpreisen für Zeit $t+1$ multipliziert. Das simulierte den Aktienhandel mit Hilfe von den erzeugten Gewichten des Modelles. Mit den monatlichen Portfoliowerten zur Hand konnten die monatlichen Renditen gerechnet werden. Das simulierte monatliches Trading und damit konnten die monatlichen Renditen summiert, um die Leistung des Portfolios auszuwerten.

$$V_p = \sum_{j=1}^A \sum_{i=1}^M P_{i+1}^T \cdot w_i$$

$M = 360$ Monate

w = monatliche Gewicht von Aktie A im Monat M

$A = 10$ Aktien

P_{1+1} = Preise im Februar 1990 von Aktie A

P = tägliche Preise von Aktie A im Monat M

w_1 = Gewicht im Januar 1990 von Aktie A

Insgesamt wurden vier Portfolios miteinander verglichen, um die Leistungsfähigkeit des Modelles zu schätzen: S&P 500 Vergleichsindex, das gleichgewichtige Portfolio, das optimierte Portfolio ohne den risikofreien Zinssatz und das optimierte Portfolio mit dem risikofreien Zinssatz. Nachdem die Renditen der Portfolios berechnet wurden, waren die Sharpe-Quotienten jedes Portfolio gerechnet worden, um zu sehen, ob die Sharpe-Quotient jedes Portfolio maximiert waren.

6 Ergebnisse

6.1 Portfoliowerte und Renditen von 1990 bis 2020

Dieser Abschnitt beschäftigt sich mit den Portfoliowerten und Portfoliorenditen, die man in diesem Experiment, besonders wenn man auf die normalisierten Portfoliowerte schaut, in die Irre führen. Bei erstem Blick auf Figure 5 sieht man die typische Gestalt des S&P 500s (gspc), der und EQUAL eine ähnliche Linie folgen. Das Portfolio unten, OPT ohne RFZ (opt), hat große Schwankungen, während OPT mit RFZ (opt_rfr) ziemlich flach aussieht. Die optimierten Portfolios, besonders OPT ohne RFZ, vielleicht keine gute Leistung gebracht hatte – dachte ich, weil normalerweise ein steigender Portfoliowert und eine höhere Rendite verbunden sind.

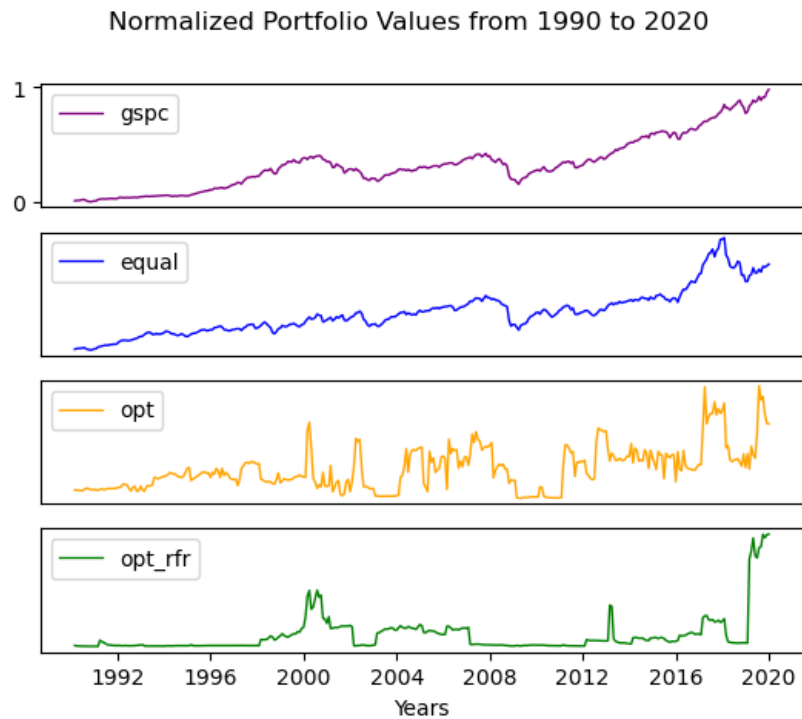


Figure 5: Normalisierte Portfoliowerte von 1990 bis 2020

Hier musste ich nachhaken und wichtige Fragen stellen. Was hat das Modell genau gerechnet und wie sollte ich die Ergebnisse rechnen und vergleichen? Das Modell ergab die durch Sharpe-Quotient optimierten monatlichen Gewichte und deswegen sollte ich die monatliche

Rendite des Portfolios betrachten, weil die Vermutung steht, dass man die Portfolios jeden Monat wieder zuteilen, oder einfacherweise, kaufen und verkaufen.

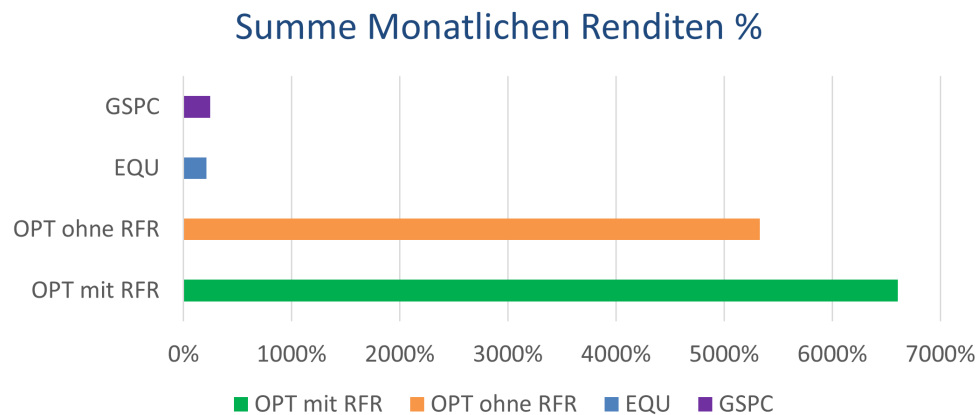


Figure 6: Summe monatlichen Renditen von 1990 bis 2020

In Figure 6 und Table 3 werden sich die Unterschiede zwischen den Portfolios abgehoben. Das GSPC ist EQUAL überlegen, aber hinkt den optimierten Portfolios stark hinterher. Das Portfolio mit den besten Renditen ist OPT mit RFZ und ist größer als GSPC um 6.356,17 Prozent. Die Einbeziehung vom RFZ hatte die Renditen in Höhe von 1.272,90 Prozent verbessert. Hier muss zugegeben werden, dass die Unterschiede der Portfolios in diesem Experiment wichtiger als die Rendite selbst, weil keine Handelskosten verrechnet werden und die Preise jede Aktie pro Monat gemittelt werden, deshalb verliert die Ergebnisse Genauigkeit. Die GSPC und EQUAL haben fast gar keine Handelskosten im Verhältnis zu den optimierten Portfolios, aber man kann mit einigermaßen Sicherheit sagen, dass es sich lohnt, das Portfolio zu optimieren.

Laut Rückmeldung nach meiner Präsentation sei es besser, mehr Information über die Gewichte darzustellen. Am Anfang fand ich schwierig diese Gewichte zu erklären, weil sie sich so jäh und schnell änderten. Meine Lösung zu diesem Problem zeigt sich in Figure 7. Diese Grafik lässt man die Entwicklung über 30 Jahre blicken, selbst wenn die einen Herzmonitor ähneln. Der klare Gewinner im OPT mit RFZ war Security National Financial Corporation oder SNFCA.

In Hinsicht auf den Aktienkurs hatte ich später nicht nur den Mittelwert des Preises,

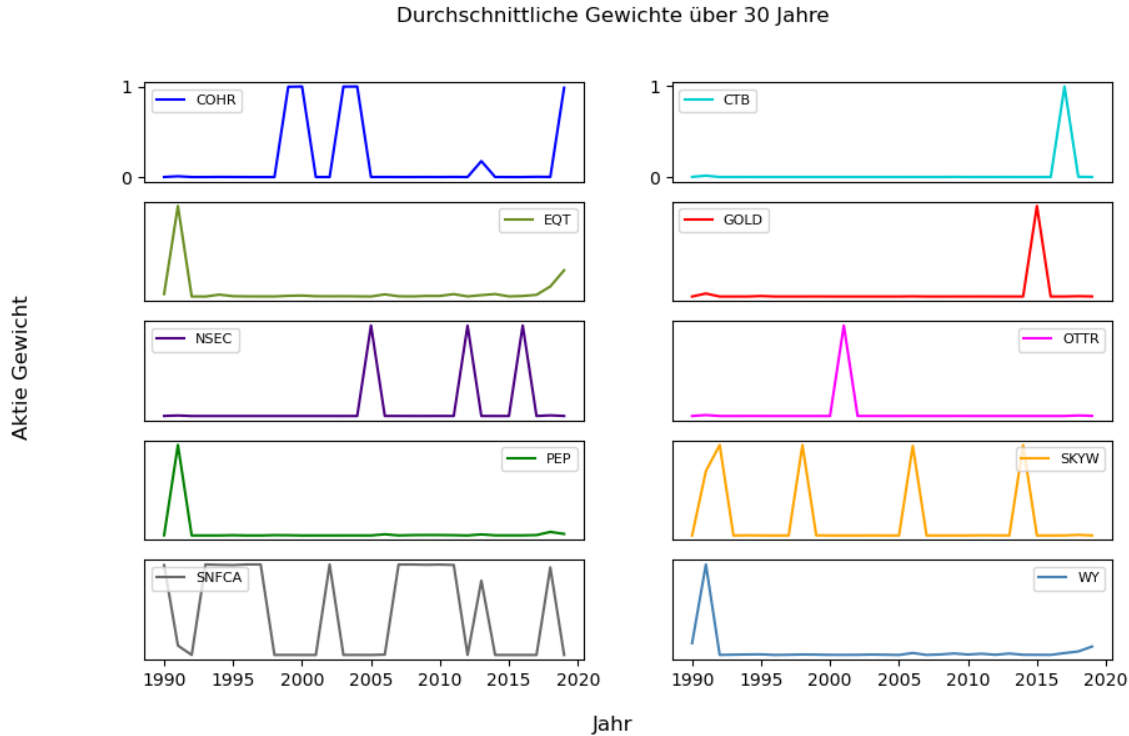


Figure 7: Gemittelte Jährliche Gewichte jede Aktie im optimierten Portfolio mit dem Risikofreien Zinssatz von 1990 bis 2020

sondern auch den ersten und letzten Preis jeden Monat eingesetzt, um die monatliche Rendite zu bekommen. Hier war die Ergebnisse des OPT mit RFR Portfolios sogar besser, beziehungsweise 6.724,74 Prozent und 6.669,50 Prozent. Diese leichte Verbesserung gilt für das GSPC auch, aber ich beachte diese Ergebnisse als unverbindlich, weil ich die nicht genau untersucht und verglichen hatte. Zusammenfassend sind die Summe prozentualer Veränderung zwischen den durchschnittlichen Preise jeden Monat für die optimierten Portfolios viel besser als die nicht optimierten Portfolios.

Ich möchte auch andere Betrachtungen über den Close-Preis, den RFZ und GSPC erörtern. Der Close-Preis beinhaltet keine Dividenden oder Aktienaufteilung und man sollte deswegen den Adjusted-Close-Preis wählen, da er den kompletten Zustand des Unternehmens berücksichtigt. In diesem Fall würde den GSPC nicht passend sein, weil der auch keine Dividenden oder Aktienaufteilung einrechnet. Außerdem würde es besser einen kurzfristigeren RFZ – Market Yield on U.S. Treasury Securities at 1-Month Constant Maturity – von FRED

einsetzen, weil die monatlichen Renditen gerechnet worden waren. In Table 7 kann man sehen, dass GSPC eine negative monatliche Sharpe-Quotient von -1,1035 hatte und das wegen eines höheren RFZs sein könnte.

Portfolio	Summe Monatlichen Renditen %
OPT mit RFR	6604,73
OPT ohne RFR	5331,83
EQU	213,32
GSPC	248,56

Table 3: Summe Monatlichen Renditen

6.2 Sharpe-Quotienten

Als nächster Teil meiner Untersuchung wurden die Sharpe-Quotienten für jedes Portfolio ausgerechnet. Da ich zwei Modelle mit und ohne den RFZ aufgebaut hatte, fand ich es in Ordnung, die Sharpe-Quotienten für jedes Portfolio mit und ohne den RFZ zu rechnen (Table 4 und Table 5). Mit diesen Parametern teilte ich die Rechnungen wieder, indem ich zuerst die Summe aller Renditen durch die Standardabweichung von allen monatlichen Renditen rechnete und dann alle monatliche Sharpe-Quotienten mittelte (Table 6 und Table 7). Der Gewinner mit der höchsten Sharpe-Quotient ohne RFZ ist GSPC in beiden Fällen. Die gesamte Sharpe-Quotient des GSPC-Portfolios ist gigantisch bei 72,302 Punkten und ist nur, wenn es mit den anderen Portfolios verglichen wurden, hilfreich. Interessanterweise hat OPT mit RFZ die schlechtesten Ergebnisse, -0,0026 Punkte, selbst wenn die Gewichte des Portfolios durch die Sharpe-Quotient optimiert wurden.

Aber, als früher erwähnt, die Sharpe-Quotient ohne den RFZ ist falsch angewendet. Es war mir eine Überraschung, wie viel die Ergebnisse sich umgekehrt hatten. Hier hatte GSPC die schlechtesten Ergebnisse in beiden Fällen. Die Sharpe-Quotient des GSPC-Portfolios ohne und mit den RFZ sank von 72,302 Punkten um 467,3736 Punkten auf -395,0716 Punkten, obwohl das OPT mit RFZ Portfolio nur um 9,7243 Punkten fiel. Das OPT mit RFZ Portfolio hat die beste Sharpe-Quotient, aber bei 0,08449 Punkten würde die Quotient sagen, dass die

Portfolio	Monatlicher Durchschnitt
OPT mit RFR	39,9735
OPT ohne RFR	41,1696
EQU	52,8962
GSPC	72,302

Table 4: Gesamtsumme der Sharpe-Quotienten ohne den Risikofreien Zinssatz

Portfolio	Monatlicher Durchschnitt
OPT mit RFR	-0,0026
OPT ohne RFR	0,0225
EQU	0,0934
GSPC	1,7403

Table 5: Monatlicher Durchschnitt der Sharpe-Quotienten ohne den Risikofreien Zinssatz

Rendite nicht genug für das Risiko wäre.

Portfolio	Monatlicher Durchschnitt
OPT mit RFR	30,2492
OPT ohne RFR	28,7633
EQU	-345,5259
GSPC	-395,0716

Table 6: Gesamtsumme der Sharpe-Quotienten mit dem Risikofreien Zinssatz

6.3 Fazit und Weiterentwicklung

Die Ergebnisse der Portfolios lässt mich zu dem Schluss kommen, dass die Wichtigkeit von passenden Gewichten im Aktienportfolio nicht unterschätzt werden sollte. Selbst beliebig ausgewählte Aktien gute Renditen erwirtschaften könnten, wenn sie im Portfolio passend zugeteilt würden. Insgesamt hatte die Sharpe-Quotient als Verlustfunktion gut funktion-

Portfolio	Monatlicher Durchschnitt
OPT mit RFR	0,08449
OPT ohne RFR	0,08034
EQU	-0,9651
GSPC	-1,1035

Table 7: Monatlicher Durchschnitt der Sharpe-Quotienten mit dem Risikofreien Zinssatz

iert, um die monatlichen Renditen anzusteigen, aber die Sharpe-Quotient der optimierten Portfolios, gegen alle Erwartungen, bleibt niedrig. Das RFZ im Modell ergibt viel besser monatlichen Renditen als ohne. Abschließend brachte das optimierte Portfolio mit RFR über 30 Jahre jährliche Rendite von 220,16 Prozent ein.

Dieses Projekt ergibt sich einginge Weiterentwicklungen aus. Wie früher erwähnt, kann ein anderer RFZ benutzt werden, damit man genauere Sharpe-Quotienten bekommt. Vielleicht verbessert diese Änderung auch die Renditen der Portfolios. Als nächster Schritt kann man ein nicht KNN-Modell benutzen, um die Unterschiede der Ergebnisse zu untersuchen. Außerdem lässt dieses Verfahren sich skalieren und könnte mit mehr als 10 Aktien benutzt werden. Ein Modell mit nur zehn Aktien wird oft bei einer Aktie übernommen, aber bei erster Probe hatte sich die Allokationen eines Portfolios mehr verteilt, als die Anzahl der Aktien wuchs. Vielleicht hätte das Modell unter Over-Fitting gelitten, weil fünfzig Epochen zu viel wären, deswegen kann man die Hyperparameter des Modelles verfeinern. Das gilt auch für die Struktur des Modelles und hier kann man mehr LSTM-Schichten hinzufügen oder die Anzahl die Einheiten in der LSTM-Schicht ändern.

Literaturverzeichnis

- [1] Martin Abadi et al. “TensorFlow: A system for large-scale machine learning”. In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. 2016, pp. 265–283. URL: <https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf>.

- [2] Data Basecamp. May 2023. URL: <https://databasecamp.de/en/ml/softmax-function>.
- [3] Christopher M. Bishop. “Neural Networks for Pattern Recognition”. In: (1995). URL: <https://people.sabanciuniv.edu/berrin/cs512/lectures/Book-Bishop-Neural%20Networks%20for%20Pattern%20Recognition.pdf>.
- [4] Jason Brownlee. *A Gentle Introduction to Backpropagation Through Time*. Aug. 2020. URL: <https://machinelearningmastery.com/gentle-introduction-backpropagation-time/>.
- [5] François Chollet et al. *Keras*. 2015. URL: <https://keras.io>.
- [6] Yahoo! Finance. *S&P 500 (^GSPC)*. URL: <https://finance.yahoo.com/quote/%5EGSPC/>.
- [7] FRED. *Market Yield on U.S. Treasury Securities at 10-Year Constant Maturity, Quoted on an Investment Basis*. URL: <https://fred.stlouisfed.org/series/DGS10>.
- [8] Charles R. Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- [9] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: <https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf>. URL: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [10] John D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.
- [11] Harry Markowitz. “Portfolio Selection”. In: *The Journal of Finance* 7.1 (1952), pp. 77–91. ISSN: 00221082, 15406261. URL: <http://www.jstor.org/stable/2975974>.
- [12] Wes McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

- [13] Oleh Onyshchak. *Stock Market Dataset*. URL: <https://www.kaggle.com/datasets/jacksoncrow/stock-market-dataset>.
- [14] Frank K. Reilly, Keith C. Brown, and Sanford J. Leeds. *Investment Analysis & Portfolio Management*. 11th ed. Cengage, 2022, p. 44. ISBN: 978-1-305-26299-7.
- [15] William F. Sharpe. “The Sharpe Ratio”. In: *The Journal of Portfolio management* (1994). URL: <https://web.stanford.edu/~wfs Sharpe/art/sr/sr.htm>.
- [16] The pandas development team. *pandas*. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: <https://doi.org/10.5281/zenodo.3509134>.
- [17] Shi Yan. *Understanding LSTM and its diagrams*. Mar. 2016. URL: <https://blog.mlreview.com/understanding-lstm-and-its-diagrams-37e2f46f1714>.
- [18] Zihao Zhang, Stefan Zohren, and Stephen Roberts. “Deep Learning for Portfolio Optimization”. In: *The Journal of Financial Data Science* 2.4 (Aug. 2020), pp. 8–20. ISSN: 2640-3943. URL: <https://arxiv.org/pdf/2005.13665v3.pdf>.