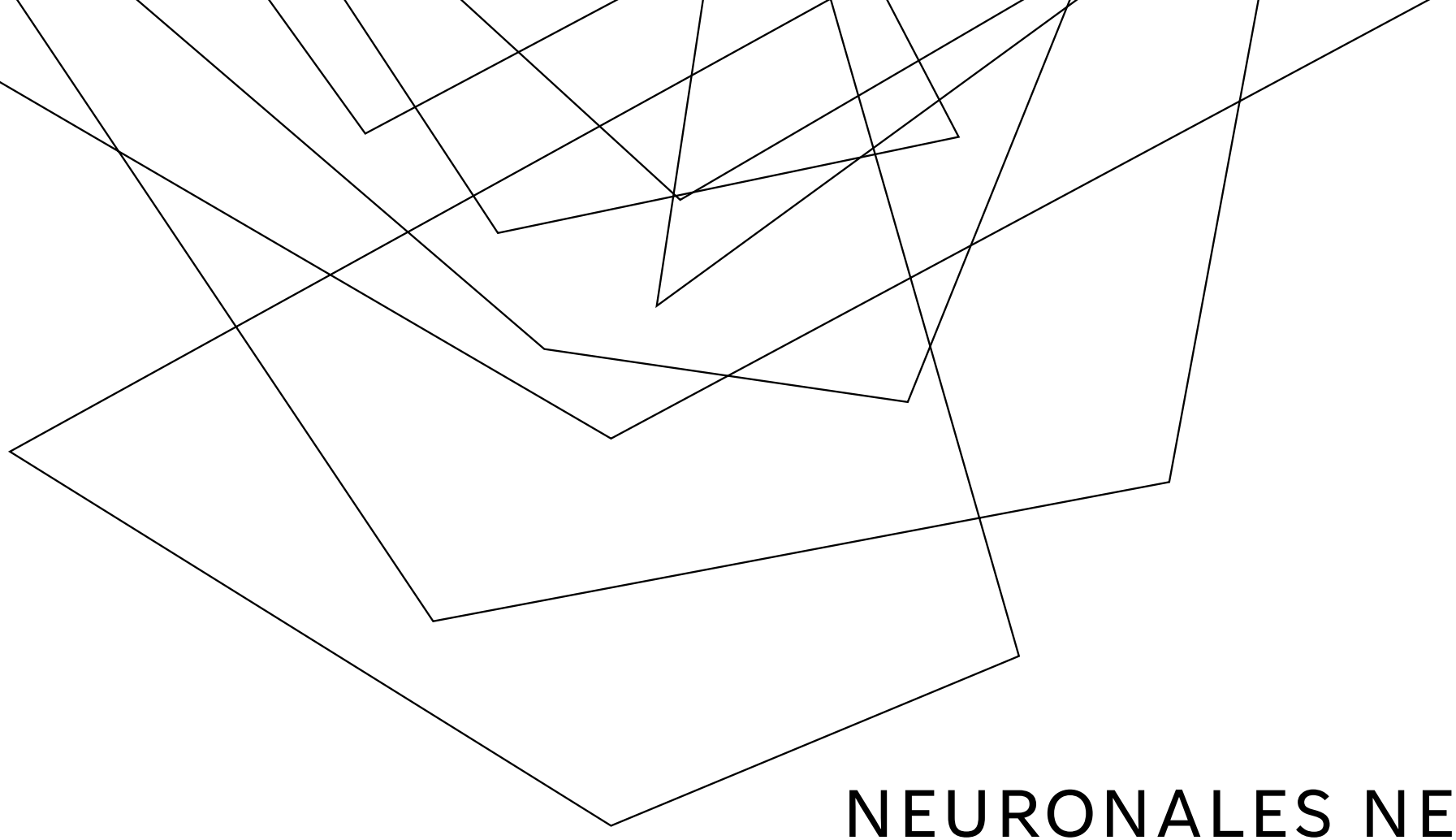
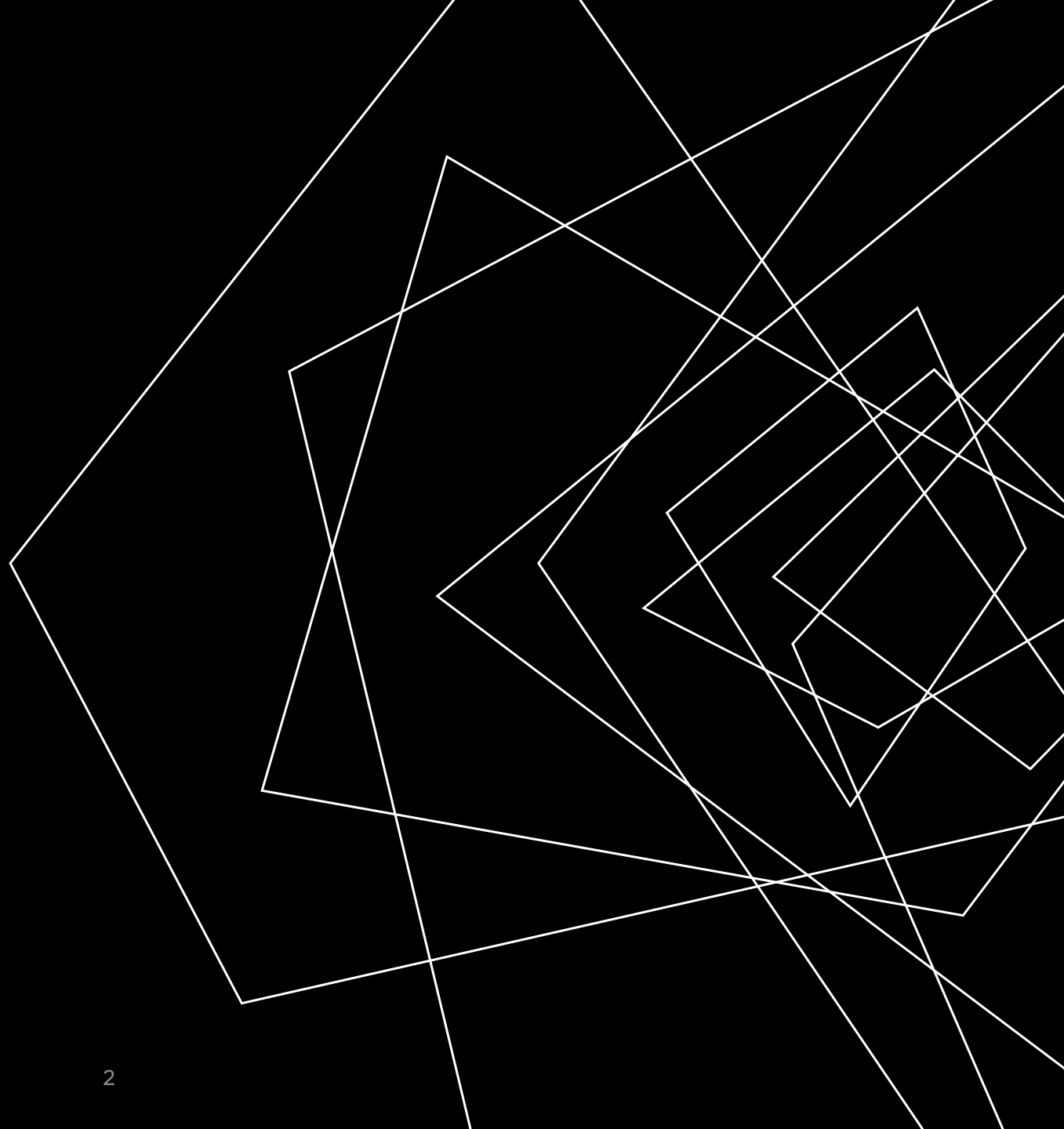
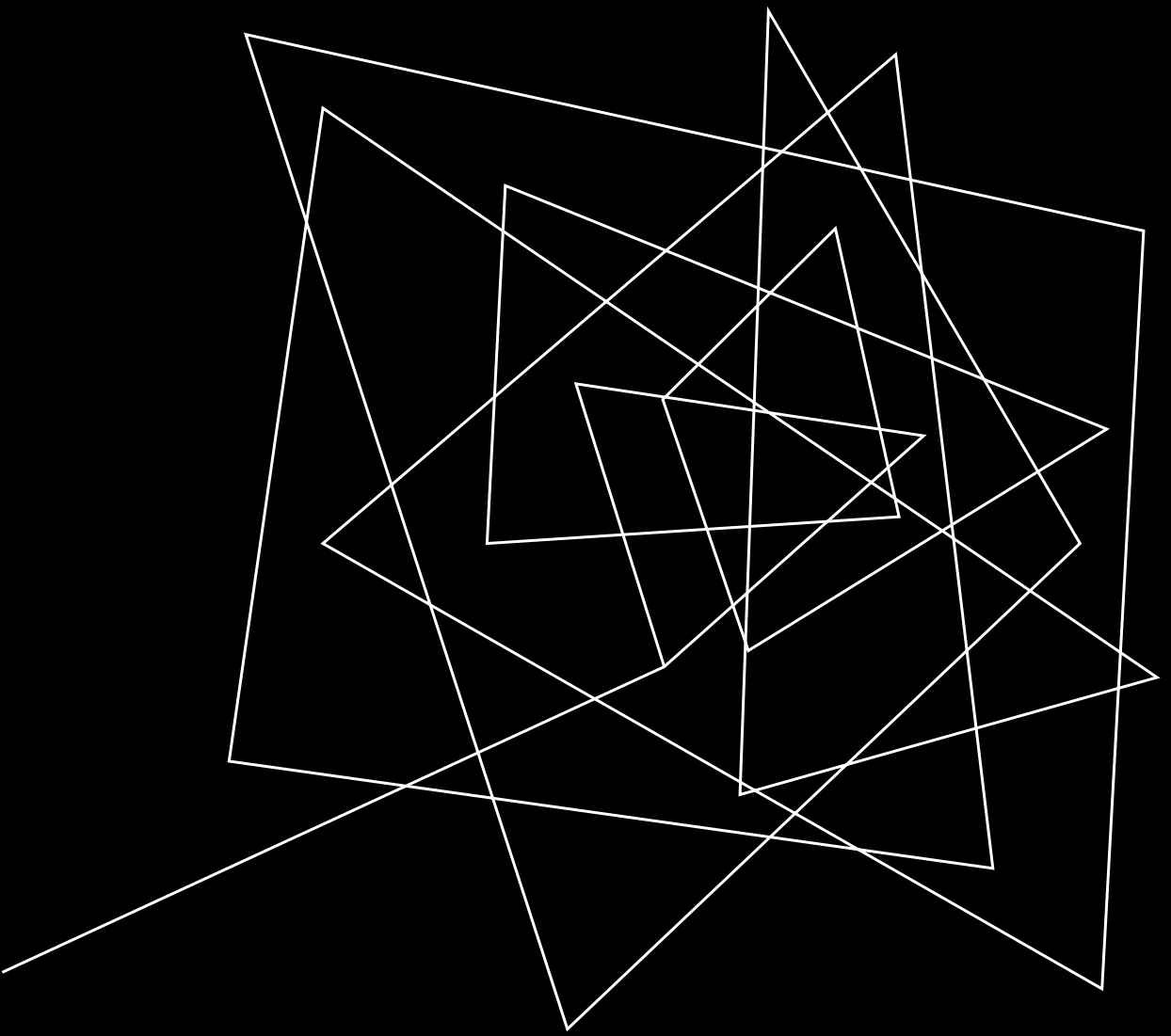


NEURONALES NETZ PORTFOLIO OPTIMIERUNG

Patrick Spohr

AGENDA

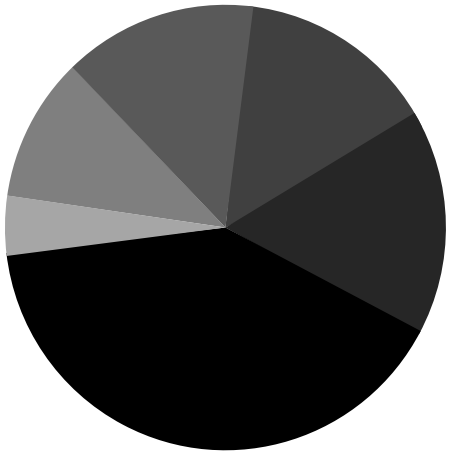




MOTIVATION

MOTIVATION

TOLLE AKTIEN GEFUNDEN...
ABER WIE PASSEN SIE IM PORTFOLIO?



Es ist einfach, wenn man nur fünf Aktien hat.

Aber wenn man fünfzig Aktien hat?

MOTIVATION

Kurze Einleitung

In drei Wörtern zusammengefasst...

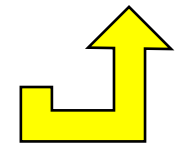
Portfolio-Allokation Optimierung

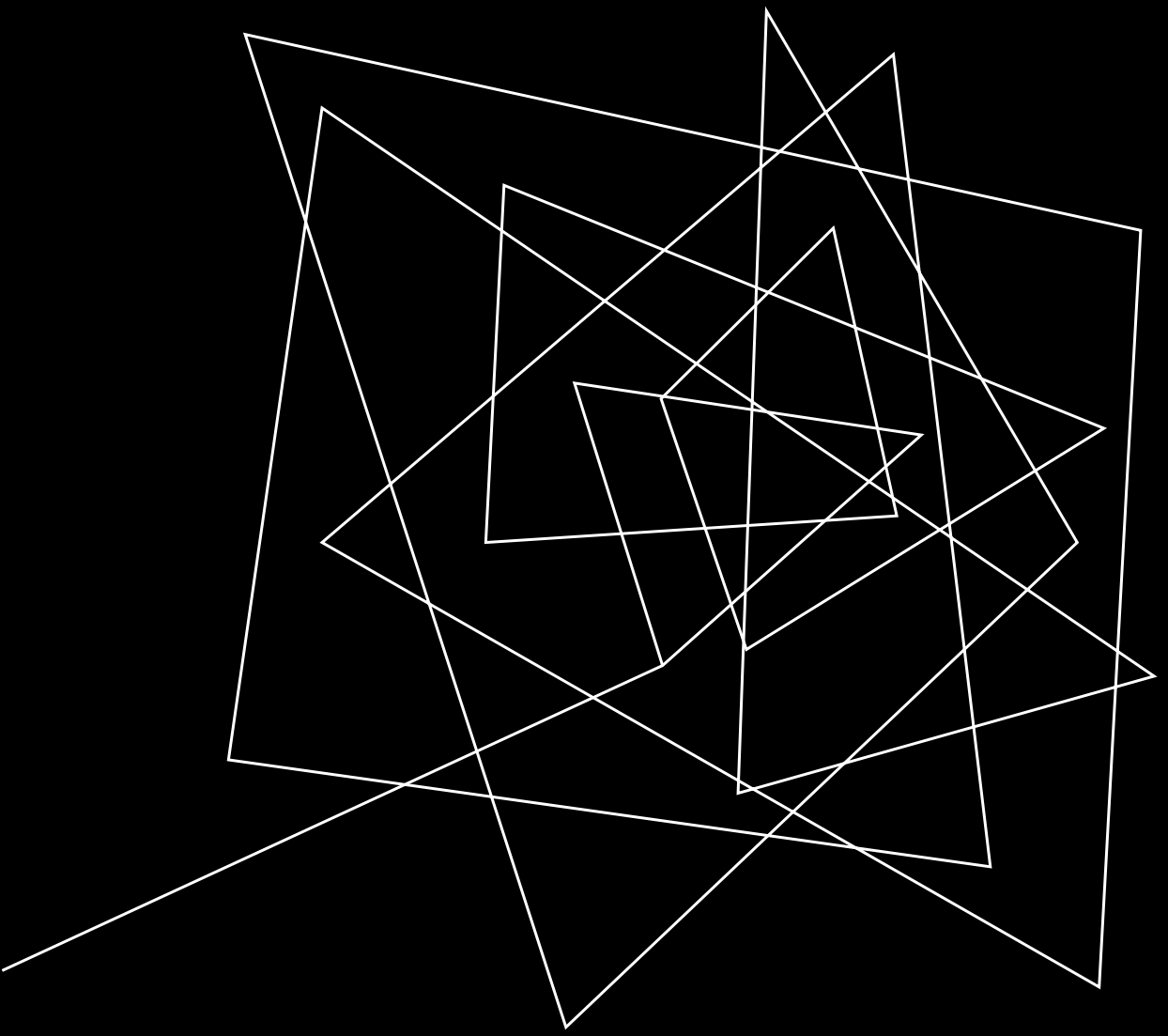
In einem Satz...

Aktien in einem Portfolio zuteilen, indem KNN und LSTM mit Sharpe-Quotient als Verlustfunktion eingesetzt wird.

Auswertung...

Die Portfolios werden mit einem gleichgewichtigen Portfolio und einem Vergleichsindex verglichen.





LITERATUR

LITERATUR

Literaturüberblick

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., . . . Google Brain. (2016). TensorFlow: A system for large-scale machine learning. 12th USENIX Symposium on Operating Systems Design and Implementation (pp. 265-283). Savannah, GA, USA: USENIX Association.

Bishop, C. M. (1995). Neural Networks for Pattern Recognition. Oxford: Clarendon Press. Retrieved from https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation 9, 8, 1735-1780.

Zhang, Z., Zohren, S., & Roberts, S. (2021). Deep Learning for Portfolio Optimization. Oxford-Man Institute of Quantitative Finance. Oxford: University of Oxford. Retrieved from https://arxiv.org/pdf/2005.13665v3.pdf

HAUPTLITERATUR

Deep Learning for Portfolio Optimization

Zihao Zhang, Stefan Zohren, Stephen Roberts
Oxford-Man Institute of Quantitative Finance,
University of Oxford

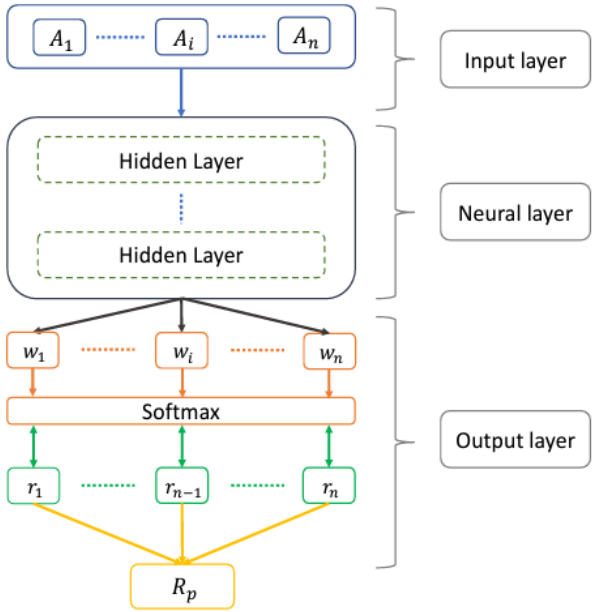


Figure 2: Model architecture schematic. Overall, our model contains three main building blocks: input layer, neural layer and output layer.

Abstract

We adopt deep learning models to directly optimise the portfolio Sharpe ratio. The framework we present circumvents the requirements for forecasting expected returns and allows us to directly optimise portfolio weights by updating model parameters. Instead of selecting individual assets, we trade Exchange-Traded Funds (ETFs) of market indices to form a portfolio. Indices of different asset classes show robust correlations and trading them substantially reduces the spectrum of available assets to choose from. We compare our method with a wide range of algorithms with results showing that our model obtains the best performance over the testing period, from 2011 to the end of April 2020, including the financial instabilities of the first quarter of 2020. A sensitivity analysis is included to understand the relevance of input features and we further study the performance of our approach under different cost rates and different risk levels via volatility scaling.

[Zhang et al.]

DEFINITIONEN

Künstliches Neuronales Netz

KNN sind Algorithmen, die den Neuronen im Gehirn nachempfunden sind. Die Gestalt aller KNN haben ein Input Layer, (mehrere) Hidden Layers und ein Output Layer.

Sharpe-Quotient

Die **SQ** ist eine Kennzahl, die die Rendite zur Volatilität eines Portfolios zeigt.

Risikofreier Zinssatz

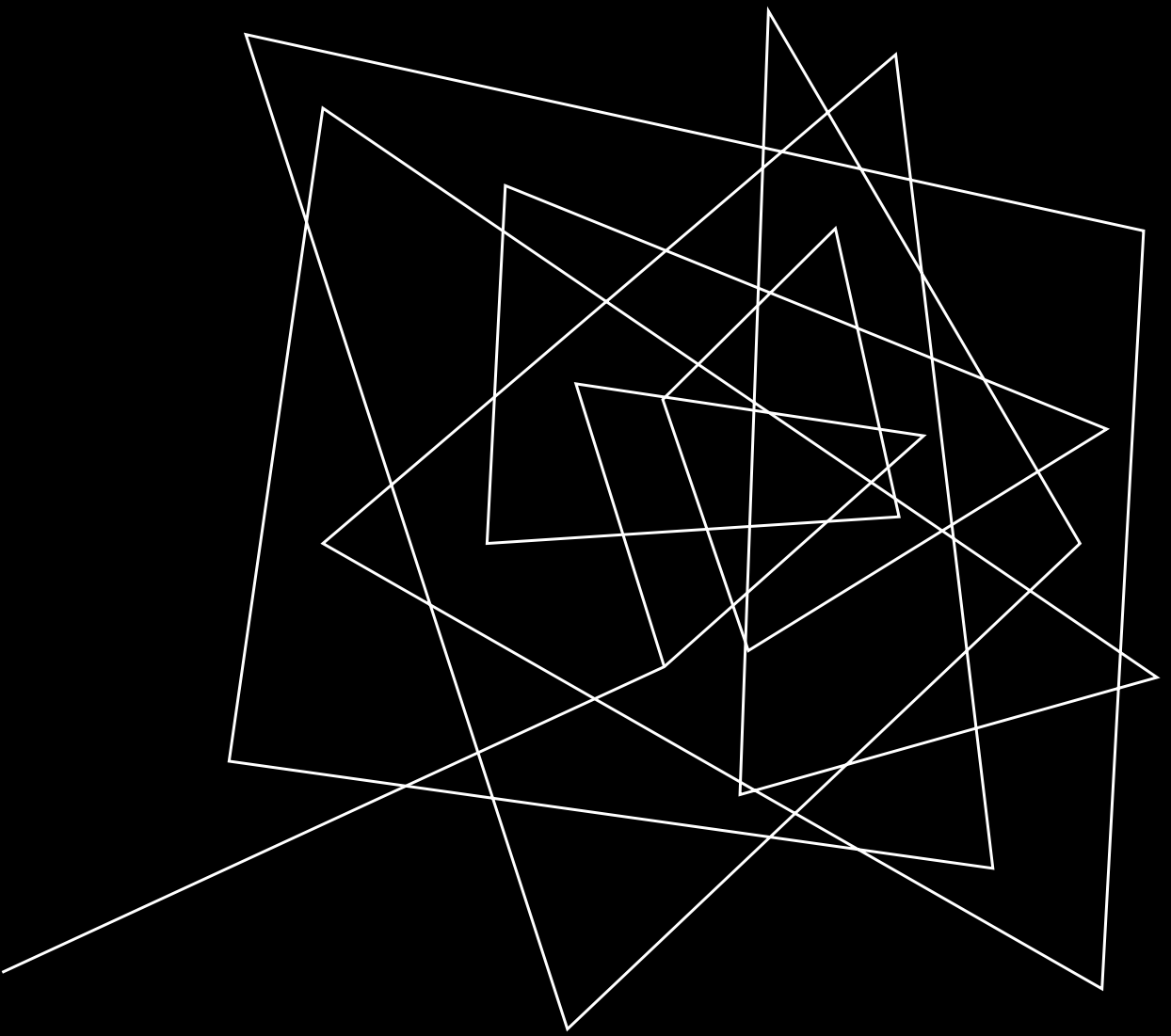
Der **RFZ/RFR** ist ein Referenzzinssatz, der theoretisch kein Ausfallrisiko enthält.

Long Short-Term Memory

LSTM bezeichnet sich als eine Technik zur Verbesserung eines rekurrenten neuronalen Netzes, indem es um eine Art Erinnerung an frühere Erfahrungen verfügt, um das Problem des vanishing Gradient zu lindern.

Verlustfunktion

Die **Verlustfunktion** misst den Fehler zwischen eine Beobachtung und eine Schätzung von einer Regression oder einer Klassifikation.



ZIELE

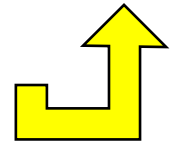
ZIELE

Was sind die Grenzen and Vermutungen meiner Projekte?

Nur Aktien sind im Portfolio.
Die Aktien sind beliebig ausgewählt (von Analysten schon geprüft).
Die Aktien sind passend für ein langfristiges Portfolio (gleiche Shape für das Modell).
Die Gewichte jede Aktie werden durch ein KNN bestimmt.

Was sind die Hauptziele?

Portfolios werden von beliebigen Aktien durch SQ mit und ohne RFR optimiert.
Leistungen werden von OPTs mit gleichgewichtigem Portfolio verglichen.
Leistungen werden von OPTs mit einem Vergleichsindex verglichen.



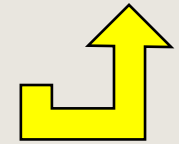
SCHRITTE

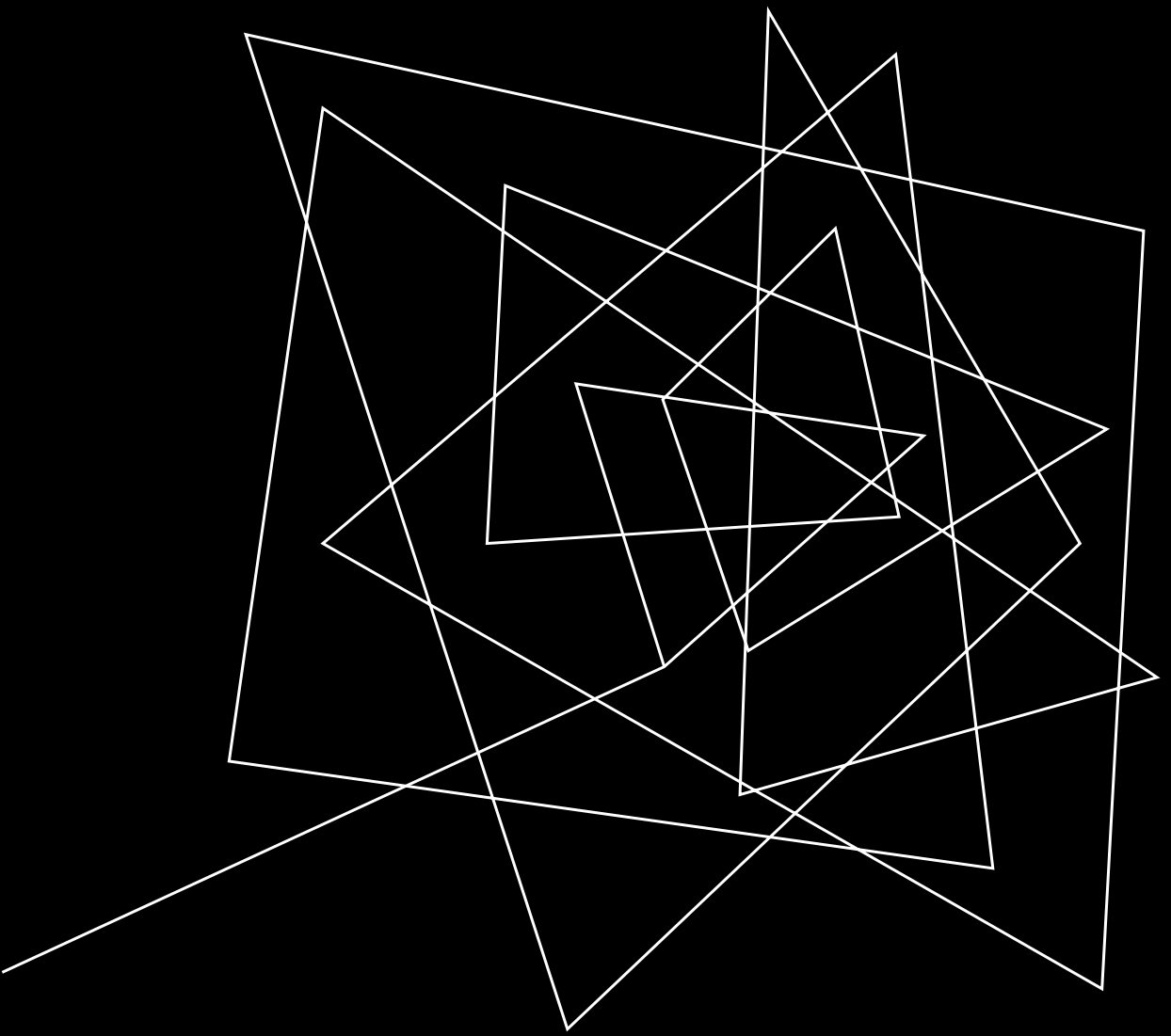
1 — Erstes KNN-Modell ohne RFR bauen

2 — Portfolio mit optimierten Gewichten für jeden Monat zuteilen

3 — KNN-Modell weiterentwickeln mit RFR

4 — Portfolio Auswertung mit einem Vergleichsindex und gleichgewichtiges Portfolio





DATEN

DATEN

Stock Market Dataset
NASDAQ

Data Structure

- **Date** - specifies trading date
- **Close** - close price

[Onyshchak]

S&P 500 (^GSPC)
Yahoo Finance

Data Structure

- **Date** - specifies trading date
- **Close** - close price

[Yahoo! Finance]

10-Year US Treasury Yield
FRED

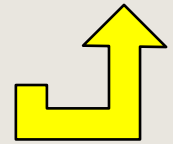
Data Structure

- **Date** - specifies trading date
- **Yield** - returns in percentages

[FRED]

ZEHN BELIEBIGE AKTIEN

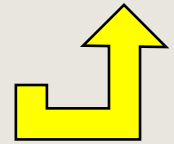
Ticker	Name	Geschäft
PEP	PepsiCo, Inc.	Getränke
CTB	Cooper Tire & Rubber Company	Reifen
SKYW	SkyWest, Inc.	Fluglinie
EQT	EQT Corporation	Energie
COHR	Coherent, Inc.	Laser
SNFCA	Security National Financial Corporation	Lebensversicherung
NSEC	National Security Group, Inc.	Versicherung
OTTR	Otter Tail Corporation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



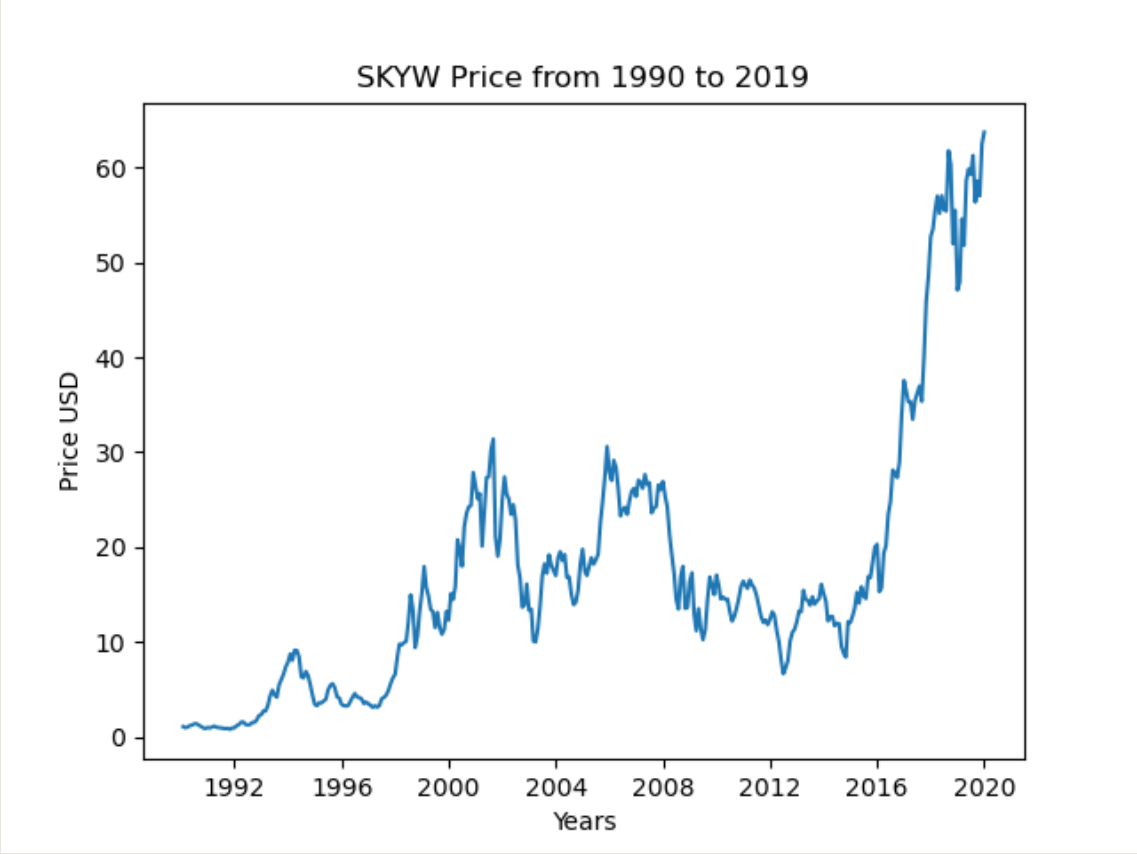
AKTIEN ÜBERBLICK

Aktien Information/Metrics zu Verfügung stehen
1990.01.02 – 2019.12.31

Höchster Preis und Datum
Niedrigste Preis und Datum
Summe monatlicher Renditen
Durchschnittliche monatlich Rendite
Durchschnittliche monatliche SQ

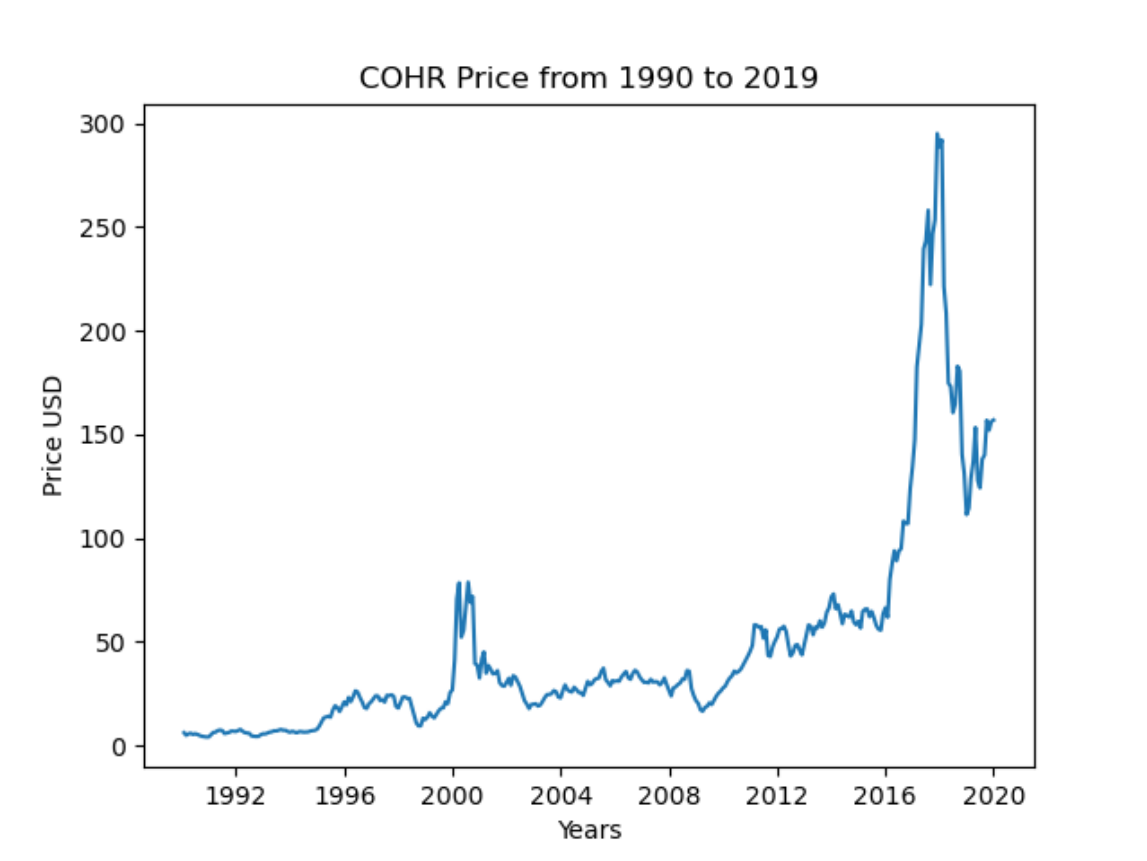


SKYW



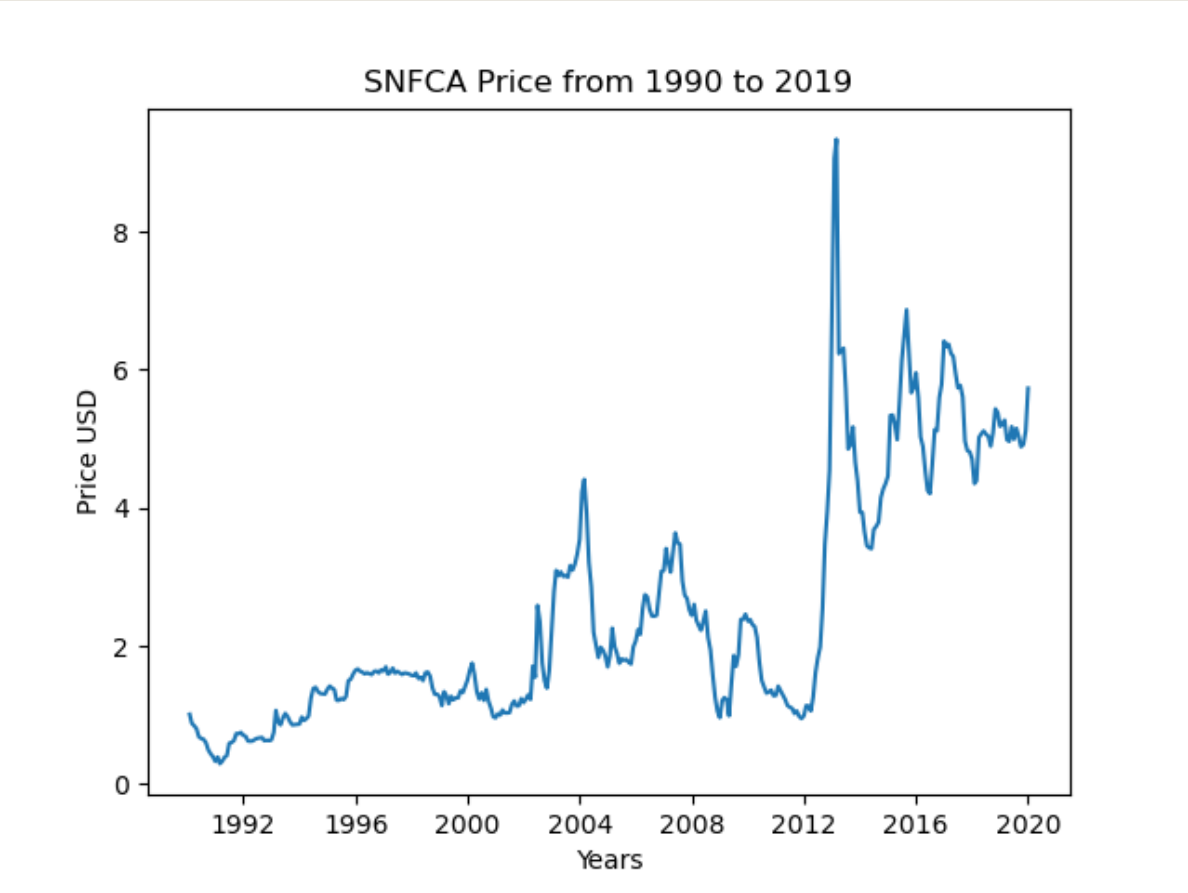
SKYW	metrics
highest_price	65.31999969
highest_price_date	12/23/2019 0:00
lowest_price	0.708333313
lowest_price_date	10/11/1991 0:00
total_monthly_returns	6.357481034
mean_monthly_returns	0.017708861
mean_monthly_sharpe	0.825336015

COHR



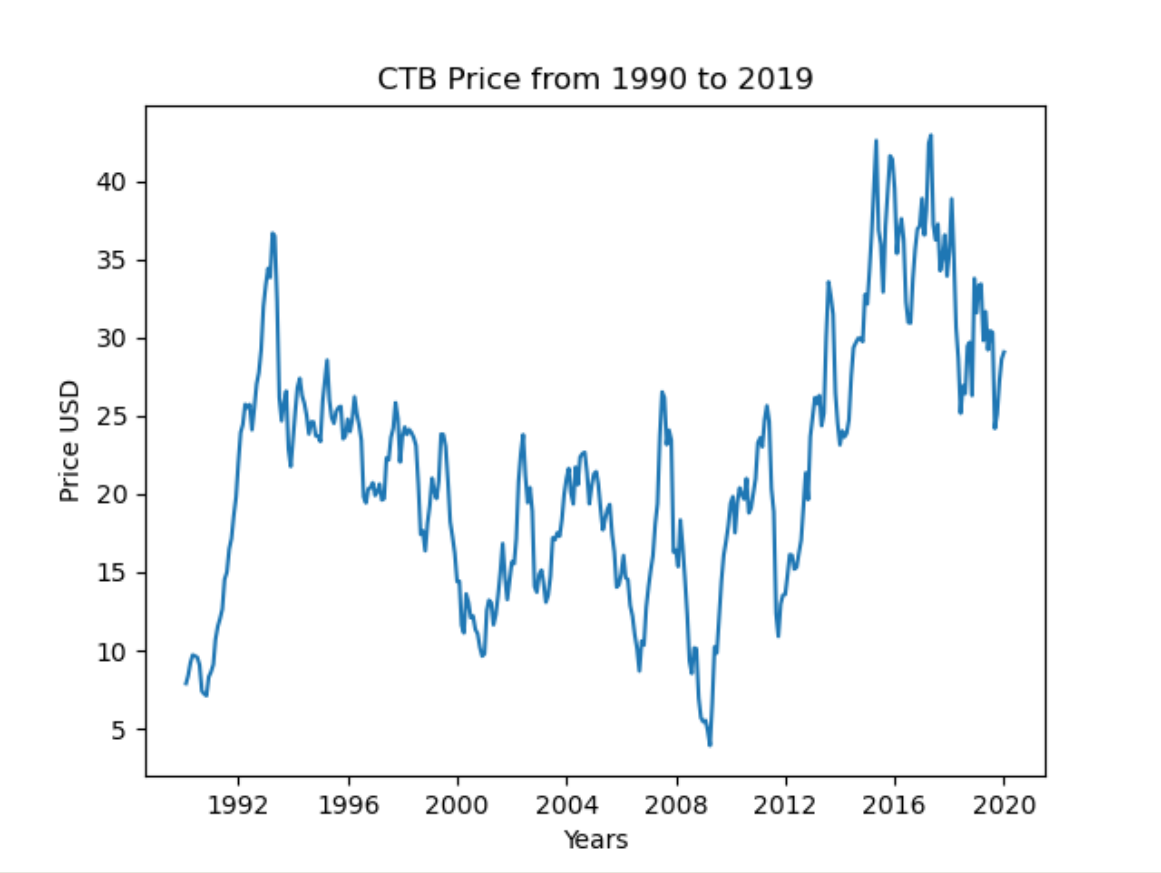
COHR	metrics
highest_price	321.9100037
highest_price_date	1/17/2018 0:00
lowest_price	3.8125
lowest_price_date	10/23/1990 0:00
total_monthly_returns	5.390640202
mean_monthly_returns	0.015015711
mean_monthly_sharpe	0.717664021

SNFCA



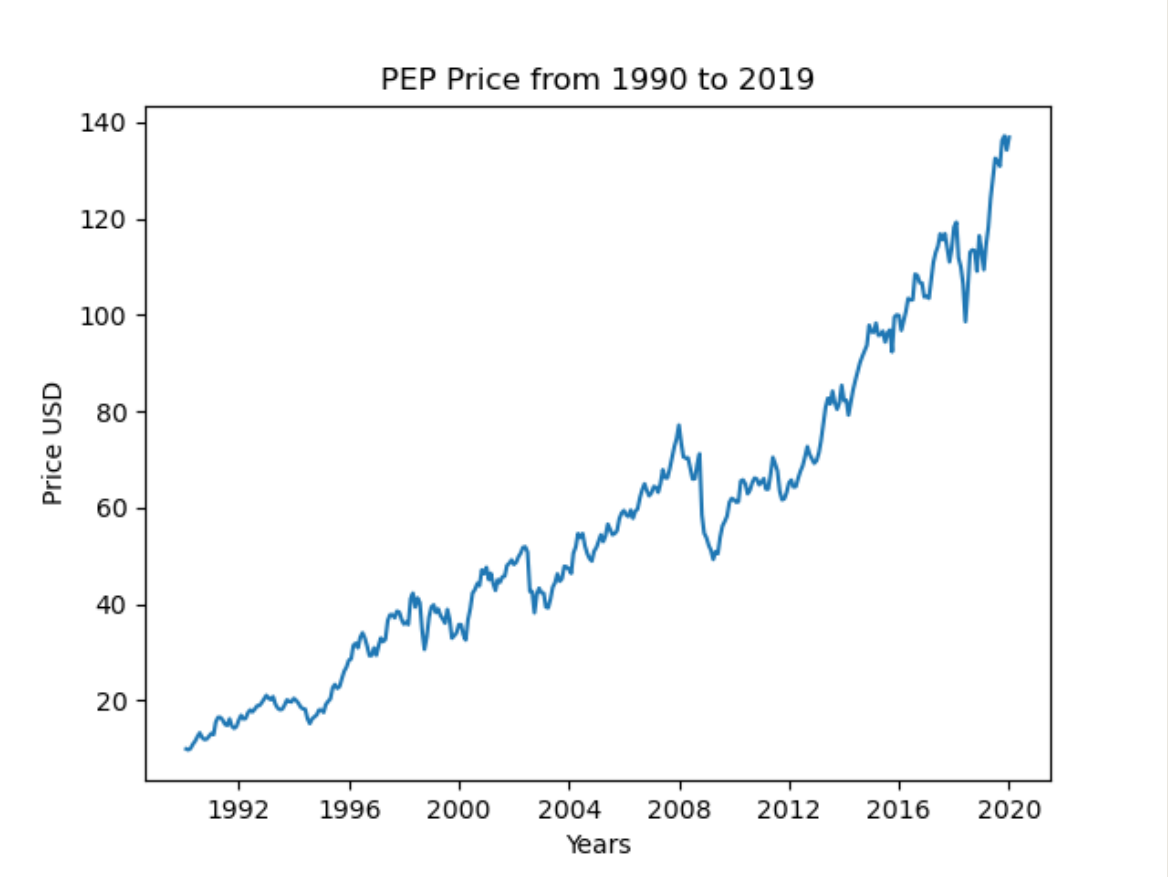
SNFCA	metrics
highest_price	11.87159634
highest_price_date	2/7/2013 0:00
lowest_price	0.281240731
lowest_price_date	12/5/1990 0:00
total_monthly_returns	3.811749281
mean_monthly_returns	0.010617686
mean_monthly_sharpe	0.112407897

CTB



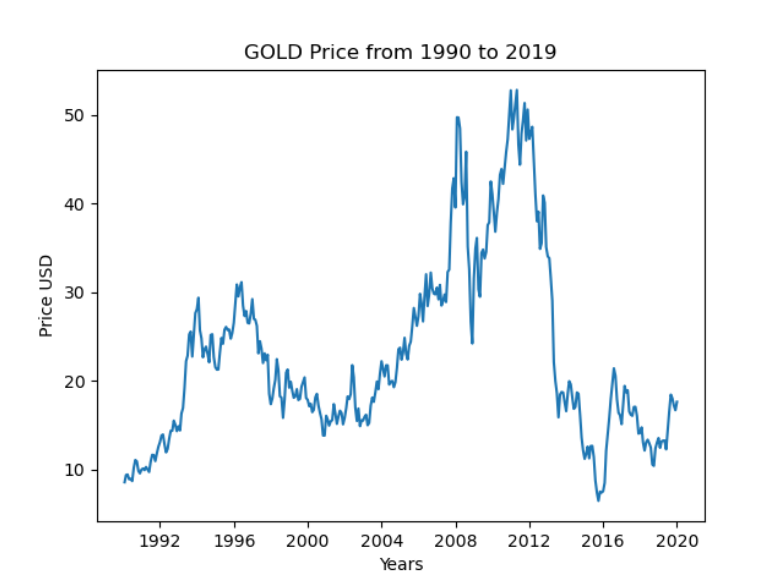
CTB	metrics
highest_price	44.45000076
highest_price_date	4/11/2017 0:00
lowest_price	3
lowest_price_date	3/9/2009 0:00
total_monthly_returns	3.120864174
mean_monthly_returns	0.008693215
mean_monthly_sharpe	0.560364359

PEP

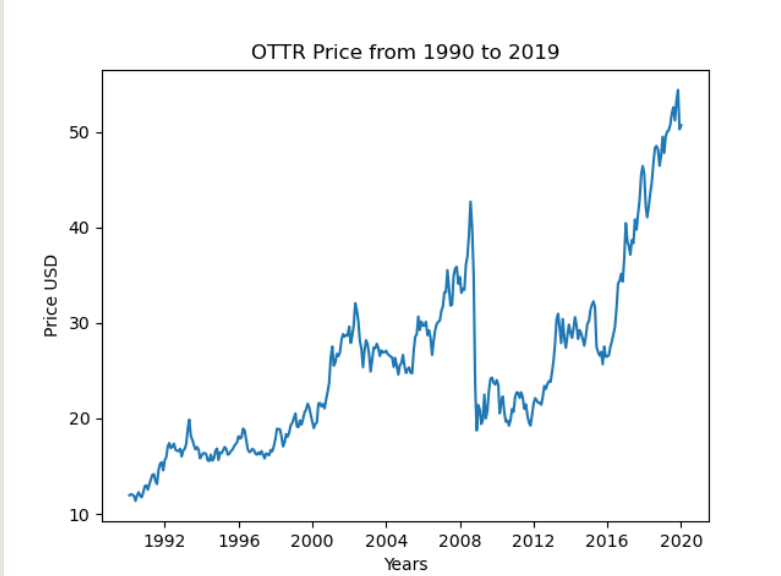


PEP	metrics
highest_price	140.2799988
highest_price_date	10/4/2019 0:00
lowest_price	9.333333015
lowest_price_date	1/26/1990 0:00
total_monthly_returns	2.97400425
mean_monthly_returns	0.008284134
mean_monthly_sharpe	0.840806316

GOLD UND OTTR

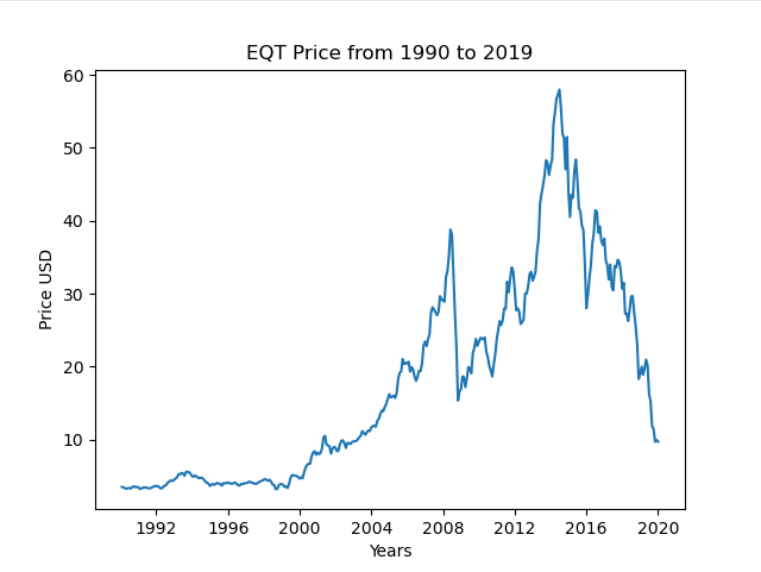


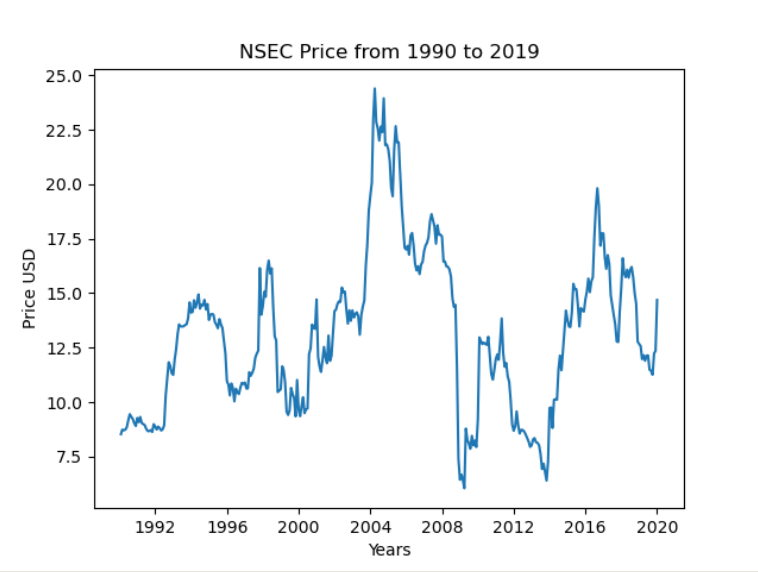
GOLD	metrics
highest_price	55.63000107
highest_price_date	4/21/2011 0:00
lowest_price	5.940000057
lowest_price_date	9/23/2015 0:00
total_monthly_returns	1.982367311
mean_monthly_returns	0.005521915
mean_monthly_sharpe	0.277853674

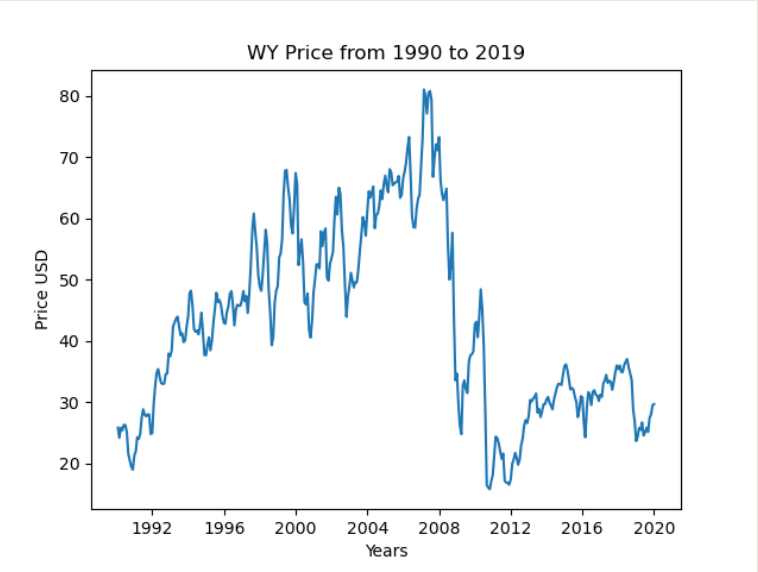


OTTR	metrics
highest_price	57.43000031
highest_price_date	11/1/2019 0:00
lowest_price	11.0625
lowest_price_date	5/21/1990 0:00
total_monthly_returns	1.855106724
mean_monthly_returns	0.005167428
mean_monthly_sharpe	0.579808703

EQT, NSEC UND WY



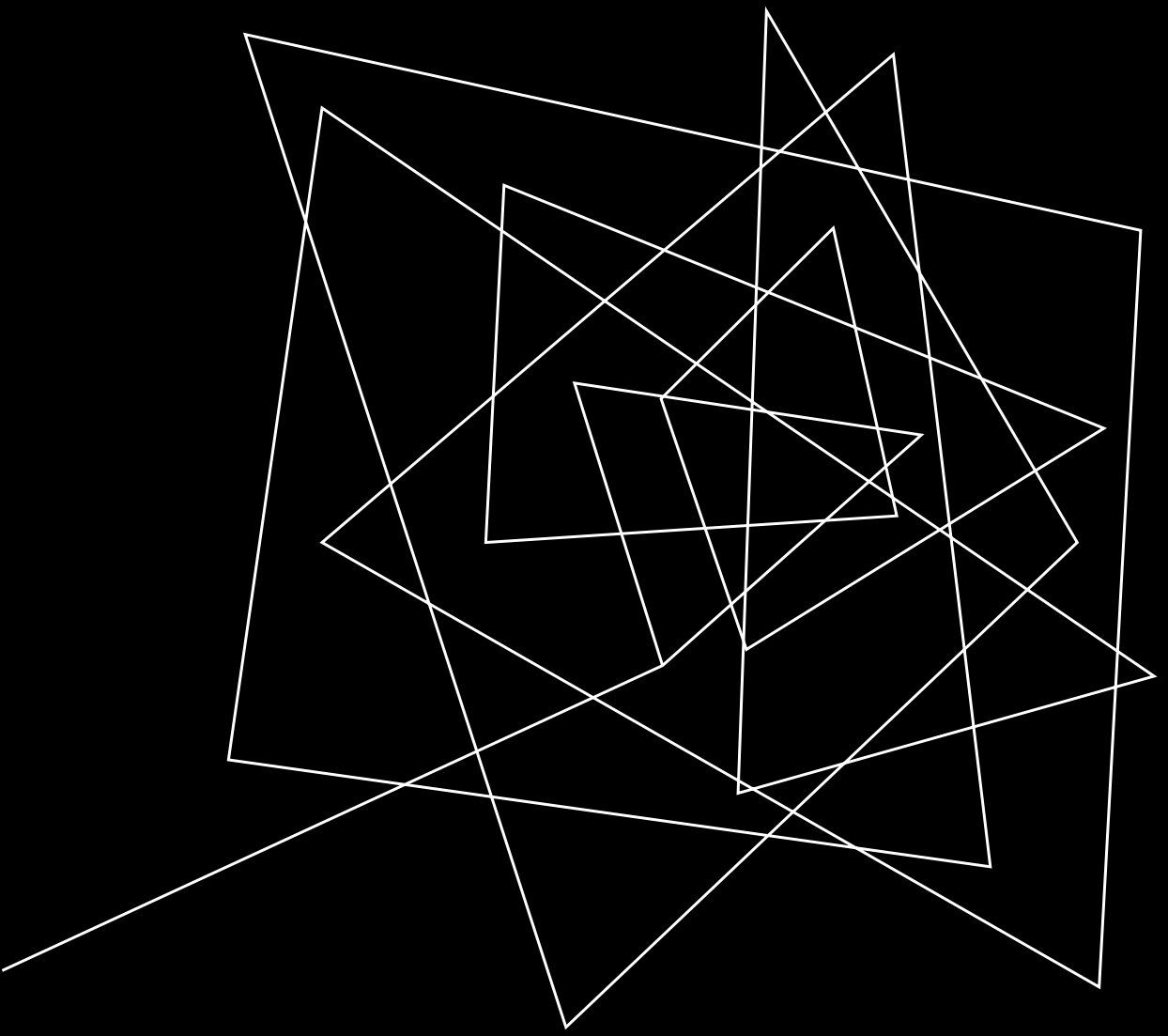




EQT	metrics
highest_price	59.79314041
highest_price_date	5/2/2014 0:00
lowest_price	2.849414825
lowest_price_date	9/3/1998 0:00
total_monthly_returns	1.763426252
mean_monthly_returns	0.004912051
mean_monthly_sharpe	0.568656196

NSEC	metrics
highest_price	26
highest_price_date	3/12/2004 0:00
lowest_price	5.159999847
lowest_price_date	11/28/2008 0:00
total_monthly_returns	1.402949382
mean_monthly_returns	0.003907937
mean_monthly_sharpe	0.150942013

WY	metrics
highest_price	86.19999695
highest_price_date	3/6/2007 0:00
lowest_price	15.22999954
lowest_price_date	10/19/2010 0:00
total_monthly_returns	1.098804923
mean_monthly_returns	0.003060738
mean_monthly_sharpe	0.485444353



METHODEN

METHODEN

Feed-Forward Künstliches Neurales Netz (KNN)

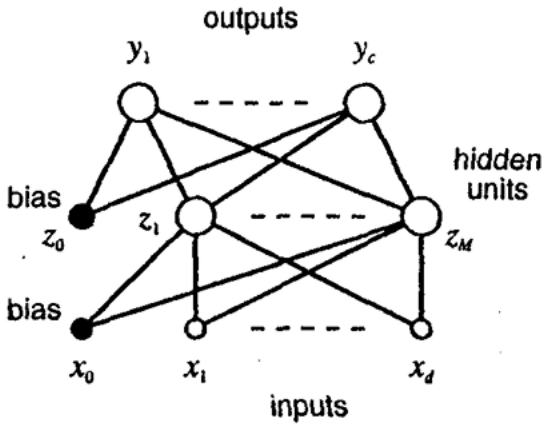


Figure 4.1. An example of a feed-forward network having two layers of adaptive weights. The bias parameters in the first layer are shown as weights from an extra input having a fixed value of $x_0 = 1$. Similarly, the bias parameters in the second layer are shown as weights from an extra hidden unit,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

„We shall view feed-forward neural networks as providing a general framework for representing non-linear functional mappings between a set of input variables and a set of output variables."

[Bishop, 117]

Die drei Teile von einem neuronalen Netz sind:

die Eingangsschicht $x_{1,\dots}x_d$

Die verborgenen Schichten $z_{1,\dots}z_M$

die Ausgangsschicht $y_{1,\dots}y_c$

METHODEN

Künstliches Neurales Netz KNN [Bishop, 116-119]

d: Inputs M: Hidden-Units c: Output-Units

Output von der Hidden-Unit

$$a_j = \sum_{i=1}^{d} w_{ji}^{(1)} x_i + w_{j0}^{(1)}$$

$w_{ji}^{(1)}$ Gewicht in ersten Layer
$w_{j0}^{(1)}$ Bias für Hidden-Unit j

$g(\cdot)$ Aktivationsfunktion

$$z_j = g(a_j)$$

Output vom Netz

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_i + w_{k0}^{(2)}$$

Komplett Funktion für Figure 4.1

$\tilde{g}(\cdot)$ Aktivationsfunktion für Output-Units

$$a_k = \tilde{g}\left(\sum_{j=1}^{M} w_{kj}^{(2)} g\left(\sum_{i=1}^{d} w_{ji}^{(1)} x_i\right)\right)$$

kth Output-Unit $y_k = \tilde{g}(a_k)$

METHODEN

SHARPE

Sharpe-Quotient

$$\text{Sharpe-Ratio} = \frac{(\text{Rendite} - \text{risikoloser Zins})}{\text{Volatilität}}$$

Sharpe-Ratio > 1: Der Fonds erwirtschaftet eine Rendite, die über dem risikolosen Zins liegt und obendrein seine Volatilität übersteigt. Der Anleger wird für sein eingegangenes Risiko also hervorragend entschädigt.

Sharpe-Ratio = 1: Der Fonds erwirtschaftet eine Rendite, die nach Abzug des risikolosen Zinses genauso hoch ist wie die Volatilität. Chancen und Risiken stehen in einem ausgewogenen Verhältnis.

Sharpe-Ratio < 1: Der Fonds erwirtschaftet eine Rendite, die nach Abzug des risikolosen Zinses niedriger liegt als die Volatilität. Der Anleger wird für sein Risiko unterdurchschnittlich entschädigt.

[Fidelity]

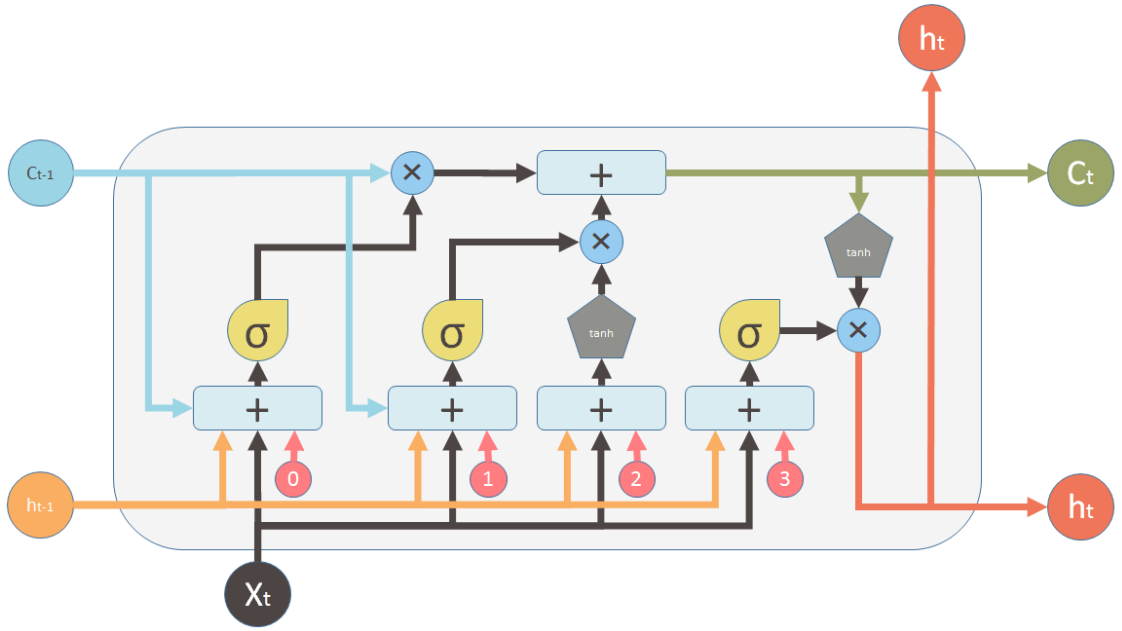
METHODEN

LSTM

LSTM wurde gebildet, um Back-Flow-Problemen von Rekurrent Netz „Short-Term Memory“ zu lindern.

[Hochreiter]

Long Short-Term Memory



Inputs:

- X_t Input vector
- C_{t-1} Memory from previous block
- h_{t-1} Output of previous block

outputs:

- C_t Memory from current block
- h_t Output of current block

Nonlinearities:

- σ Sigmoid
- tanh Hyperbolic tangent

Vector operations:

- × Element-wise multiplication
- \+ Element-wise Summation / Concatenation

Bias: 0

[Yan]

METHODEN

Softmax

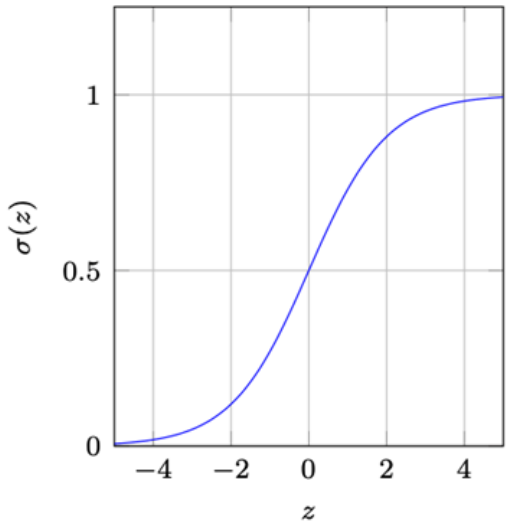
Für Klassifikation mit mehr als zwei Klassen. Die Funktion versichert, dass die Wahrscheinlichkeit von aller Klassen 1 ergibt.

Wichtig für die Gewichten für das Portfolio!

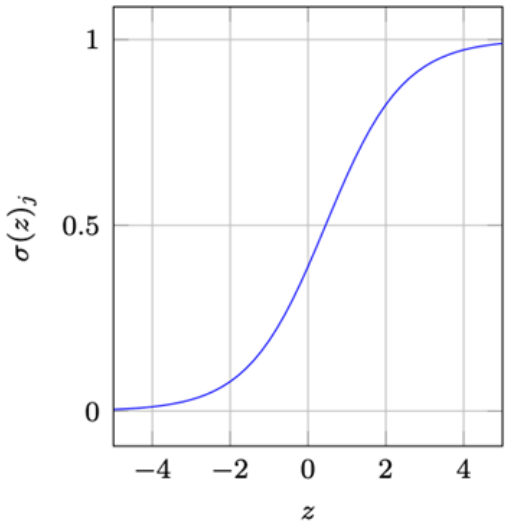
Softmax-Funktion

$$\sigma(x)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

Für j = 1, … , K
x = Input Vektor



(a) Sigmoid activation function.



(b) Softmax activation function.

[Data Base Camp]

METHODEN

PYTHON + LIBRARIES

NumPy – math + ndarray + random

https://numpy.org/doc/stable/reference/index.html

pandas – data handling

https://pandas.pydata.org/docs/reference/index.html

Matplotlib – graphing

https://matplotlib.org/stable/index.html

Other Programs Used

Visual Studio Code (code editor)
Anaconda (Python environments)
Jupyter Notebook (interactive code)

Library Versions

python	3.10.13
tensorflow	2.10.0
numpy	1.26.0
pandas	2.1.1
matplotlib	3.8.0

METHODEN

Sharpe Ratio
OHNE Risk-Free Rate

X Features

A Aktien mit x Close Preis

Y Target

y Gewicht für jede N Aktien

```python
def sharpe_loss(_, y_pred):

    # make all time-series start at 1
    data = tf.divide(self.data, self.data[0])

    # value of the portfolio after allocations applied
    portfolio_values = tf.reduce_sum(tf.multiply(data, y_pred), axis=1)

    # [:-1] stops iteration just before last index
    portfolio_returns = (portfolio_values[1:] - portfolio_values[:-1]) / portfolio_values[:-1]  # % change formula

    # (sharpe = return - risk free rate) / standard deviation
    # did not take risk free rate into account
    sharpe = tf.math.reduce_mean(portfolio_returns) / tf.math.reduce_std(portfolio_returns)

    # since we want to maximize Sharpe, while gradient descent minimizes the loss,
    # we can negate Sharpe (the min of a negated function is its max)
    return -sharpe
```

https://github.com/p-spohr/NN-Portfolio-Optimizer/blob/main/portfolio_allocations_1M.py

METHODEN

Sharpe Ratio
MIT Risk-Free Rate

X Features

A Aktien mit x Close Preis

Y Target

y Gewicht für jede N Aktien

```python
def sharpe_loss(_, y_pred):

    # make all time-series start at 1 (Normalization?)
    data = tf.divide(self.data, self.data[0])
    rfr = tf.divide(self.rfr, self.rfr[0])

    # value of the portfolio after allocations applied
    portfolio_values = tf.reduce_sum(tf.multiply(data, y_pred), axis=1)

    # [:-1] stops iteration just before last index
    portfolio_returns = (portfolio_values[1:] - portfolio_values[:-1]) / portfolio_values[:-1]  # % change formula

    # (sharpe = return - risk free rate) / standard deviation
    # did not take risk free rate into account
    sharpe = tf.math.reduce_mean(portfolio_returns - rfr) / tf.math.reduce_std(portfolio_returns)

    # since we want to maximize Sharpe, while gradient descent minimizes the loss,
    # we can negate Sharpe (the min of a negated function is its max)
    return -sharpe
```

https://github.com/p-spohr/NN-Portfolio-Optimizer/blob/main/portfolio_rfr_allocations_1M.py

MONATLICHE PORTFOLIOWERTE VEKTOR

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

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

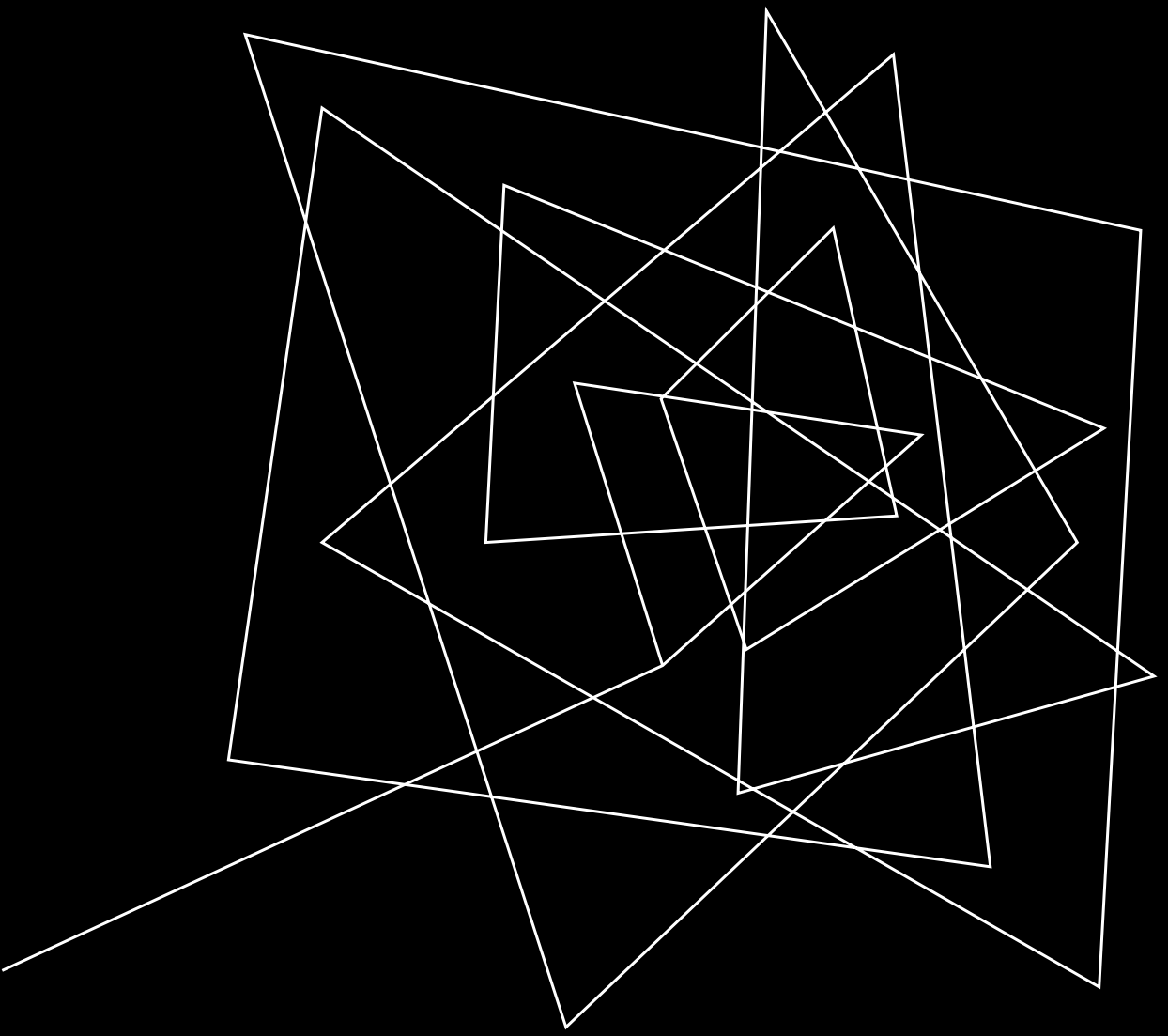
$w_1 = Gewicht\ im\ Januar\ 1990\ von\ Aktie\ A$

$M = 360\ Monate \qquad i = 1, \dots, M$

$A = 10\ Aktien \qquad j = 1, \dots, A$

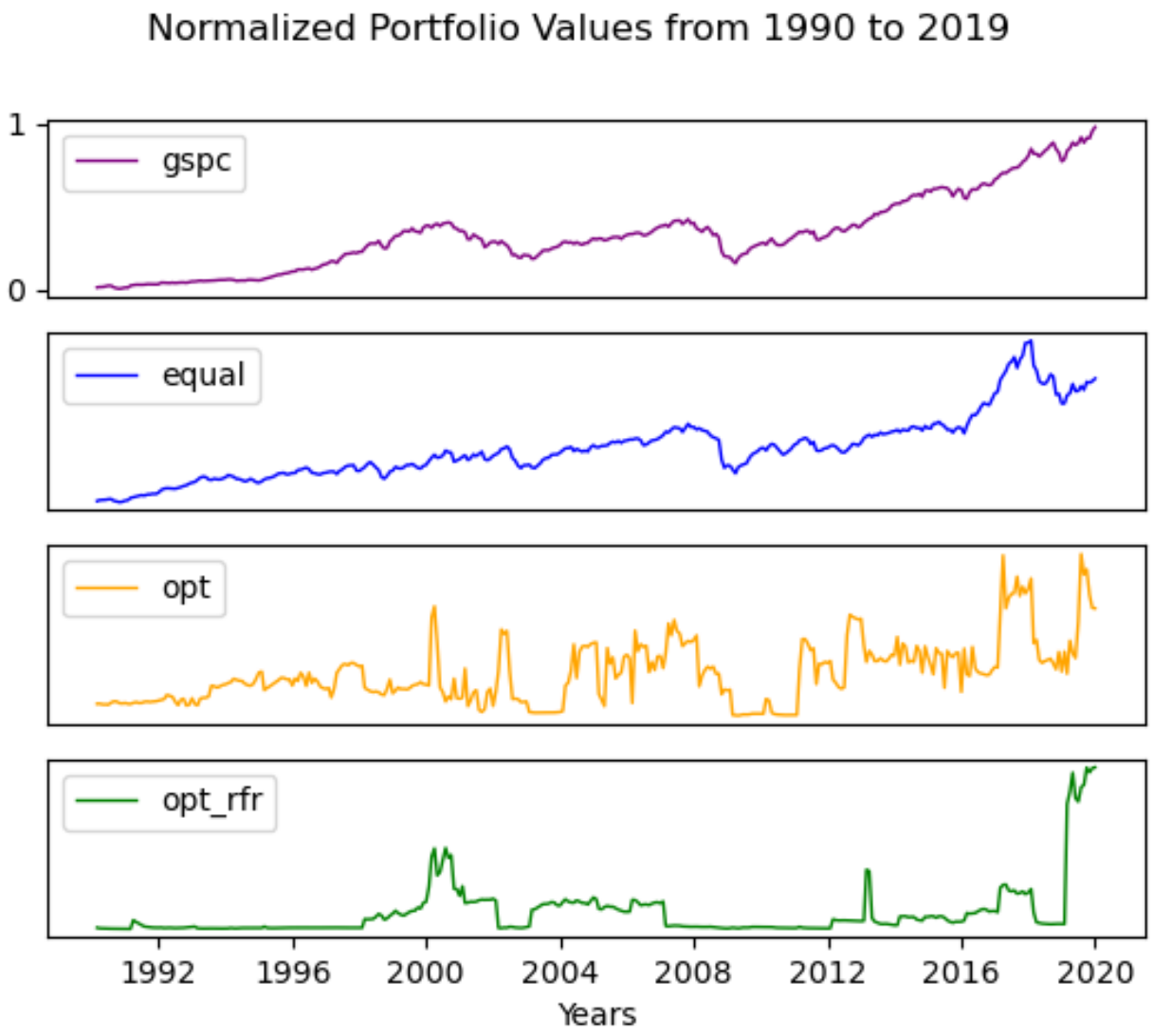
$P = tägliche\ Preise\ Vektor\ von\ Akite\ A\ im\ Monat\ M$

$w = monatliche\ Gewicht\ von\ Akite\ A\ im\ Monat\ M$



ERGEBNISSE

NORMALISIERTE PORTFOLIOWERTE



S&P 500 Vergleichsindex

Portfolio Gleichgewichtig

Optimiertes Portfolio ohne RFR

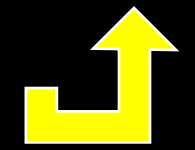
Optimiertes Portfolio mit RFR

Schwierig, die Leistung von jedem Portfolio zu verstehen und es sieht als ob GSPC und EQUAL besser sind...

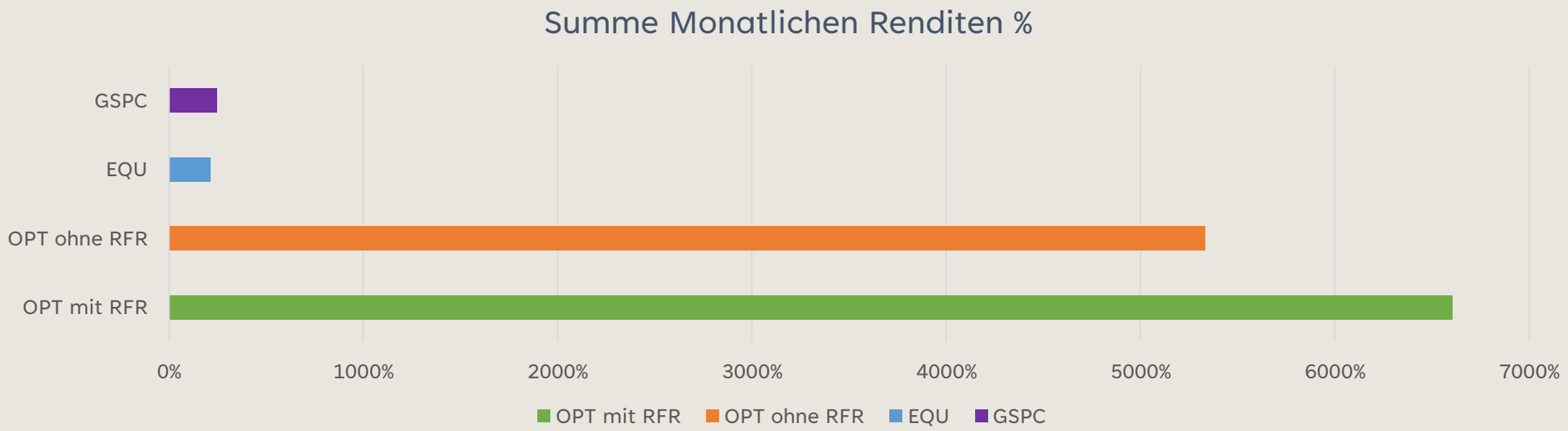
WICHTIGE FRAGE

Was sind die Summe aller monatlichen Rendite?
OPT-Portfolios: optimiert für jeden Monat, verkauft und dann neu zuteilt

Was sind die Unterschiede zwischen die Portfolios mit und ohne RFR?
Wie sieht die Sharpe-Quotienten von die vier Portfolios aus?



PORTFOLIO RENDITEN



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

Methode	Renditen
Erst	**6724,74**
Letzte	6669,50

OPT mit RFR wenn erster und letzter, statt durchschnittlicher Preis

Summe prozentuale Veränderung zwischen durchschnittlichem Preis jeden Monat

SHARPE-QUOTIENTEN OHNE RFR

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

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

$$\frac{\sum_{i=1}^{M} Rendite_i}{\sigma_M} \qquad i = 1,..,M \qquad M = 359\ months$$

$$\frac{1}{M}\sum_{i=1}^{M}\frac{Rendite_i}{\sigma_i} \qquad i = 1,..,M \qquad M = 359\ months$$

$\sigma_M = Standardabweichung\ von\ allen\ monatlichen\ Rendite$ $\sigma_i = Standardabweichung\ von\ monatliche\ Rendite$

SHARPE-QUOTIENTEN MIT RFR (RICHTIG)

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

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

$$\frac{\sum_{i=1}^{M} Rendite_i - RFR_i}{\sigma_M} \qquad i = 1,..,M \qquad M = 359\ months$$

$$\frac{1}{M}\sum_{i=1}^{M} \frac{Rendite_i - RFR_i}{\sigma_i} \qquad i = 1,..,M \qquad M = 359\ months$$

$\sigma_M = Standardabweichung\ von\ allen\ monatlichen\ Rendite$ $\sigma_i = Standardabweichung\ von\ monatliche\ Rendite$

FAZIT

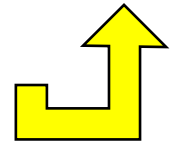
- ***Das zeigt die Wichtigkeit von Aktien Gewichten im Portfolio, weil selbst beliebig ausgewählte Aktien gute Rendite erwirtschaften kann, wenn sie im Portfolio passend zugeteilt werden.***

- *Das Sharpe Ratio als Verlustfunktion hat gut funktioniert.*

- *Das RFR im Modell ergibt besser Rendite.*

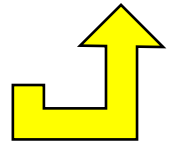
- *Das optimierte Portfolio mit RFR hat über 30 Jahre jährliche Rendite von 220,16 % eingebracht.*

- *Dieses Verfahren lässt sich skalieren und könnte mit mehr als 10 Aktien benutzt.*



WEITERENTWICKLUNG

- Werte OPTs mit Nicht-KNN aus (gurobi-optimods)
  - https://github.com/Gurobi/gurobi-optimods/blob/main/src/gurobi_optimods/sharpe_ratio.py
- Bilde ein größeres Portfolio (mehr als 10 Aktien)
- Baue neue Modell Struktur auf
  - Mehr LSTM Layers oder Nodes
  - Weniger Epochs (Over-Fitting)



LITERATURVERZEICHNIS (1/3)

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., . . . Google Brain. (2016). TensorFlow: A system for large-scale machine learning. 12th USENIX Symposium on Operating Systems Design and Implementation (pp. 265-283). Savannah, GA, USA: USENIX Association.

Bishop, C. M. (1995). Neural Networks for Pattern Recognition. Oxford: Clarendon Press. Retrieved from https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf

Data Base Camp. (2020, May 20). Retrieved from What is the Softmax-Function?: https://databasecamp.de/en/ml/softmax-function

Fidelity. (n.d.). Risiken effektiv abschätzen mit Risikomaßen. Retrieved from Fidelity: https://www.fidelity.de/wissen/tipps-and-strategien/risiko-kennziffern/sharpe-ratio/

LITERATURVERZEICHNIS (2/3)

FRED. (n.d.). Market Yield on U.S. Treasury Securities at 10-Year Constant Maturity, Quoted on an Investment Basis. Retrieved from FRED: https://fred.stlouisfed.org/series/DGS10

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation 9, 8, 1735-1780.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95.

McKinney, W., & others. (2010). Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56).

LITERATURVERZEICHNIS (3/3)

Onyshchak, O. (2020, April 01). Stock Market Dataset. Retrieved from Kaggle: https://www.kaggle.com/datasets/jacksoncrow/stock-market-dataset

Sharpe, W. F. (1994). The Sharpe Ratio. The Journal of Portfolio Management. Retrieved from https://web.stanford.edu/~wfsharpe/art/sr/sr.htm

Yahoo! Finance. (n.d.). S&P 500 (^GSPC). Retrieved from Yahoo! Finance: https://finance.yahoo.com/quote/%5EGSPC/

Yan, S. (2016, March 13). Understanding LSTM and its diagrams. Retrieved from Medium: https://blog.mlreview.com/understanding-lstm-and-its-diagrams-37e2f46f1714

Zhang, Z., Zohren, S., & Roberts, S. (2021). Deep Learning for Portfolio Optimization. Oxford-Man Institute of Quantitative Finance. Oxford: University of Oxford. Retrieved from https://arxiv.org/pdf/2005.13665v3.pdf

VIELEN DANK!

Patrick Spohr

HTW Berlin – FAR Masterstudiengang

Statistical Learning in Finance and Insurance

Prof Dr Christina Erlwein-Sayer