

Quantum Deep Hedging: Potential von quantengestützten Machine Learning-Verfahren als Hedging-Strategie

Hendrik Heine | Daniel Ohl de Mello | Daniel Herr | Ferdinand Graf

Die Schlüsseltechnologie des maschinellen Lernens bzw. der künstlichen Intelligenz (KI) konnte in den letzten Jahren eine spürbare Revolution erfahren und hat sich zu einem wichtigen strategischen, wirtschaftlichen und industriellen Erfolgsfaktor entwickelt. Dabei nimmt die Anwendung von KI-Methodiken zur Klassifizierung, der Anomalieerkennung oder auch der Regressionsanalyse beispielsweise in Hinblick auf Strategien zur Portfoliooptimierung im Finanzsektor eine zunehmend größere Bedeutung ein. In diesem Kontext stellen wir unsere Analysen und deren Ergebnisse zum Ansatz des sogenannten „Quantum Deep Hedging“ vor, welcher das Hedging von Finanzinstrumenten basierend auf tiefen neuronalen Netzen (Deep Neural Networks – DNN) mit „Reinforcement Learning“-Techniken und Anteilen von Quanten-Neuronalen-Netzen beschreibt.

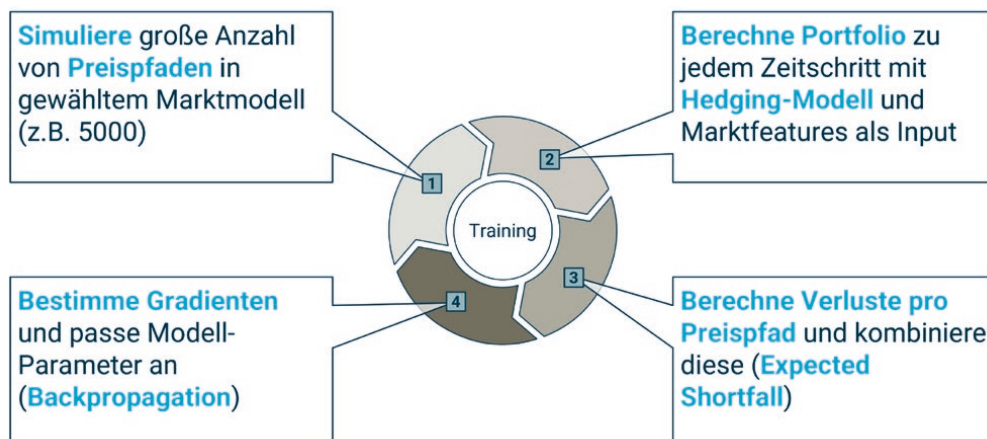
Hedging

Hedging bezeichnet allgemein die Strategie des Absicherns oder der Reduzierung bzw. Minimierung von Risiken, welche durch nachteilige Preisveränderungen im Rahmen des Handels von Finanzinstrumenten entstehen können. Gängige standardisierte Hedging-Verfahren umfassen unter anderem Delta- bzw. Gamma-Hedging sowie Monte-Carlo-Simulations-basierte Ansätze. Die Entwicklung einer erfolgreichen Hedging-Strategie ist höchst anspruchsvoll. Neben der natürlichen Dynamik des Markts mit vielfältigen Korrelationen zwischen verschiedenen Instrumenten tragen Einflüsse durch Nebenbedingungen wie Liquidität, Limitierung von Kapital sowie Transaktionskosten wesentlich zur Komplexität des Problems bei.

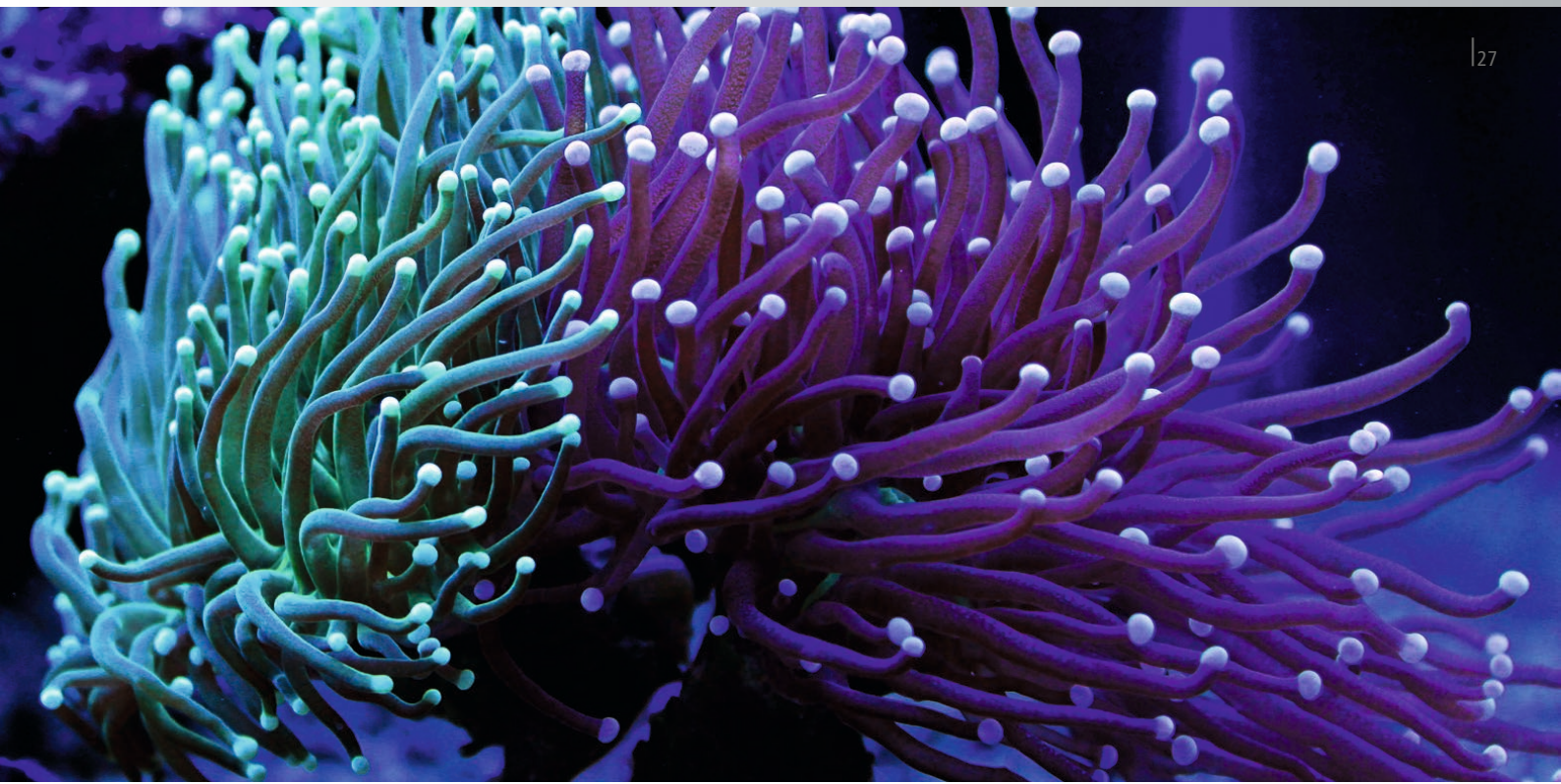
Reinforcement Learning

Systeme des maschinellen Lernens können grundsätzlich nach bestimmten Kriterien wie dem Typ der Supervision klassifiziert werden. Neben dem überwachten beziehungsweise unüberwachten Lernen (supervised/unsupervised learning) stellt das sogenannte „Reinforcement Learning“ eine eigene Methodik dar, bei der das lernende System (auch „Agent“ genannt) mit seiner Umgebung interagiert und gemäß bestimmter Regelwerke Handlungen vornimmt, für die es entweder eine positive oder negative Belohnung (oder Strafe) erhält. Das Regelwerk wird dabei entsprechend der Lernschritte aktualisiert, bis eine optimale Strategie gefunden wurde. Das Ziel für den Agenten besteht darin ein Verhalten zu erlernen, welches seine erwarteten Belohnungen maximiert.

Abb. 01: Schematische Darstellung des Trainingsablaufs eines Deep Hedging Modells.



Quelle: d-fine GmbH



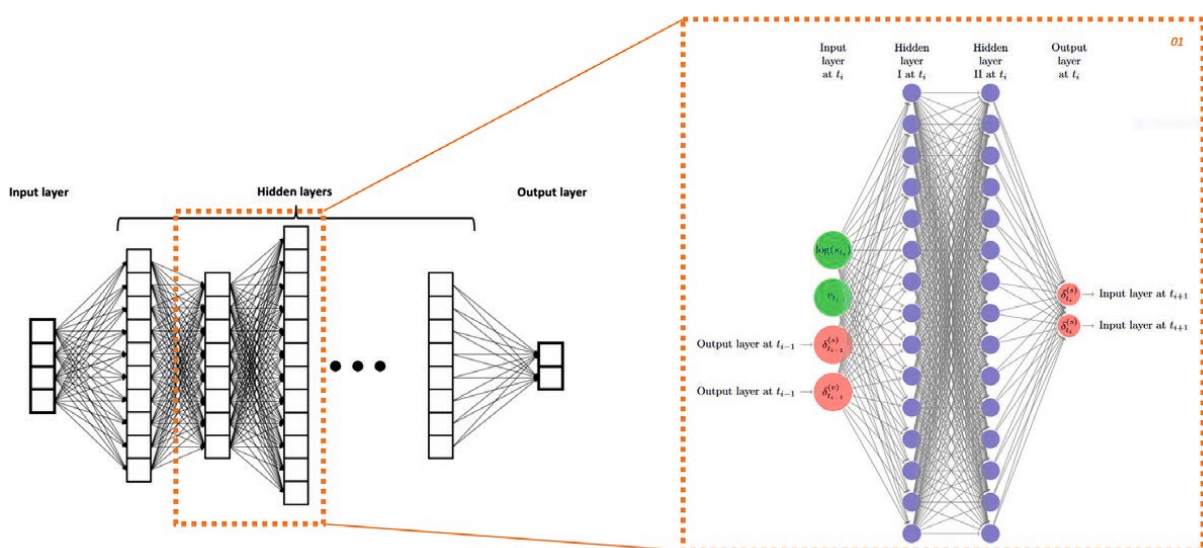
Deep Hedging

Beim „Deep Hedging“ werden Handelsentscheidungen innerhalb der Hedging-Strategie durch ein tiefes neuronales Netzwerk modelliert, wobei unterschiedliche zusätzliche Faktoren wie vergangene Handelsentscheidungen, Liquiditätseinschränkungen oder Nachrichtenauswertungen als „Features“ in das Gesamtmodell integriert werden können. Im Zuge der Kalibrierung eines stochastischen Preismodells, das als Input für das neuronale Netz fungiert, kann entweder die Methode der impliziten Volatilitäten oder auch ein klassischer Zeitreihenansatz unter Verwendung von historischen Werteentwicklungen verwendet werden. Das Ziel des Lernprozesses des neuronalen Netzwerkes besteht darin, eine definierte mathematische Zielfunktion innerhalb der Formulierung

eines mathematischen Optimierungsproblems zu minimieren. Im hier betrachteten Fall verwendeten wir hierfür den 10%-Expected Shortfall als Risikomaß.

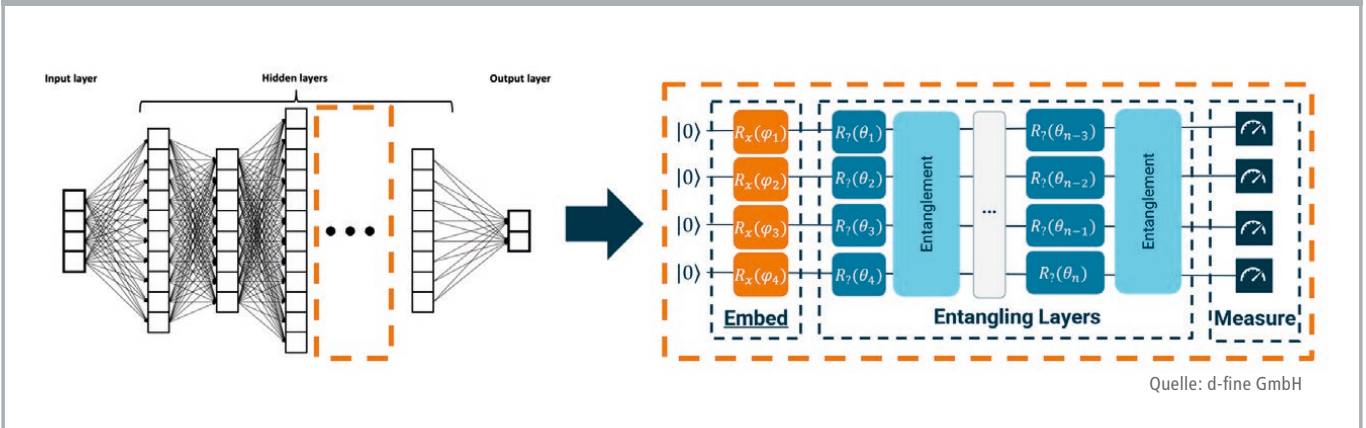
Die grundlegenden relevanten Schritte zum Training des neuronalen Netzwerkes sind in ► Abb. 01 gezeigt. In der technischen Umsetzung wurden das „pfhedge“-Framework [1], welches auf der Pytorch-Bibliothek [2] basiert, verwendet und weiterentwickelt [4]. Das Framework dient insbesondere der Bereitstellung eines umfassenden Modells für Marktprozesse (z.B. Black-Scholes, Heston, Merton-Jump) und Instrumente (z.B. Europäische, Binäre oder Lookback Optionen).

Abb. 02: Darstellung des tiefen neuronalen Netzwerkes (DNN) inklusive der Input/Output Layer sowie verschiedener verborgener (hidden) Layer (links). Detailliertere Auflösung im Bereich der verborgenen Layer mit Abbildung der Neuronen (hier „Perceptron“) als Komponenten der verschiedenen Layer. Grafik adaptiert aus [3].



Quelle: d-fine GmbH

Abb. 03: Hybride Architektur des tiefen neuronalen Netzes mit Quantum-Komponenten. Hierbei entsteht ein hybrides neuronales Netz indem Teile von klassischen verborgenen Layers durch trainierbare Quantum-Komponenten ersetzt werden.

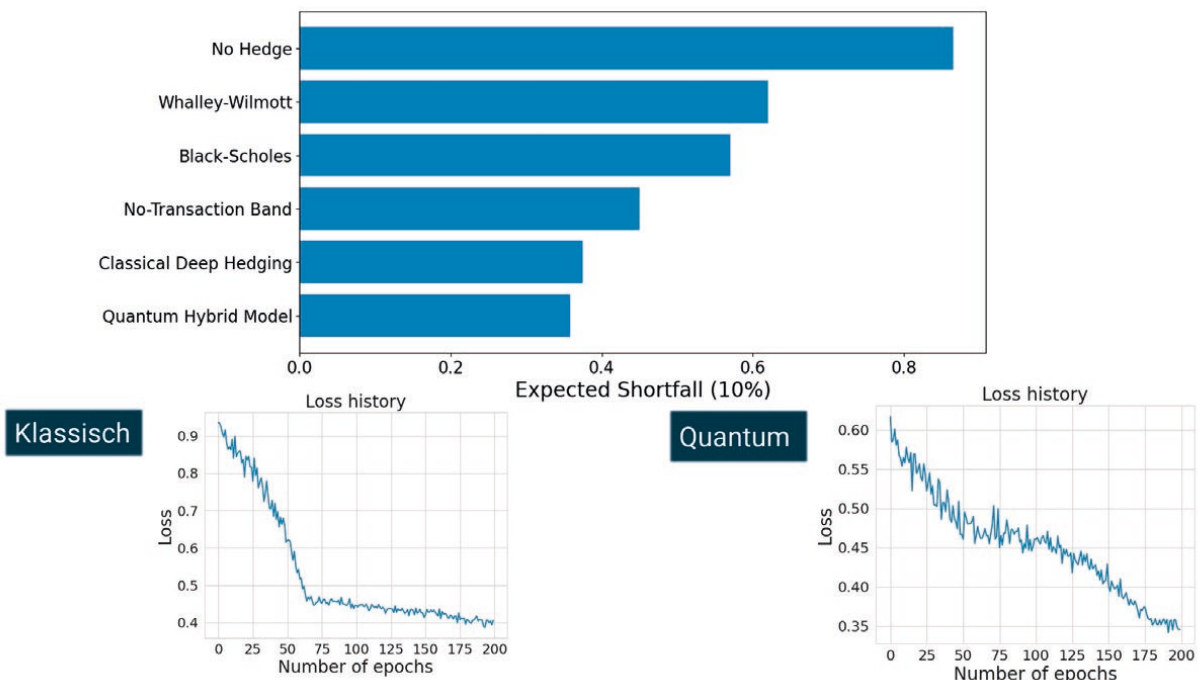


Quantum Computing

Die ersten grundsätzlichen Überlegungen zur Konzeption eines Quantencomputers existieren bereits seit einigen Jahrzehnten. Die technischen Realisierungen unterschiedlicher Architekturen haben jedoch erst in den vergangenen Jahren insbesondere durch die Beteiligung der Entwicklung durch die großen Technologie-Konzerne einen wesentlichen Schub erfahren. Der erhoffte Quantenvorteil gegenüber dem klassischen Computer basiert dabei auf drei grundlegenden Eigenschaften von Quantensystemen bzw. Quantenzuständen. Hierbei handelt es sich um die Superposition, die Verschränkung (engl. Entanglement) von Quantenzuständen, sowie die Möglichkeit der Interferenz der Zustände.

Beim klassischen Computer stellt das klassische Bit mit den binären Zuständen $b=\{0,1\}$ die kleinste Einheit des Informationsgehalts dar. Das Analogon beim Quantencomputer ist das Qubit und wird durch einen quantenmechanischen Zustand repräsentiert, welcher erst zum Zeitpunkt der physikalischen Messung in einen der beiden Basiszustände 0 oder 1 projiziert wird. Ohne den Vorgang der Messung befindet sich der Quantenzustand im Allgemeinen in einer Superposition der beiden Basiszustände, darf sich also gleichzeitig im Zustand 0 und 1 befinden. Beim Vorhandensein von n Qubits ergibt sich demnach eine exponentielle Skalierung der möglichen Basiszustände zu 2^n . Die Manipulation von Qubits erfolgt durch (unitäre) Operatoren (ähnlich zu klassischen logischen Gattern),

Abb. 04: Vergleich von klassischen und quanten-hybriden Modellen in einem restriktiven Setting (beschränkte Parameterzahl und Trainings-Epochen). Oben: Das Quantum-hybride Modell performt vergleichbar mit dem klassischen Deep Hedging und wesentlich besser als traditionelle Ansätze. Unten: Verlauf der jeweiligen Verlustfunktionen über die Trainingsepochen.



welche durch iterative Ausführung und Schaltung zu Quantengattern kombiniert werden können. Die Parameter dieser Operatoren (beispielsweise die Drehung um einen bestimmten Winkel) können in gewisser Analogie zu den Gewichten und dem Bias der Neuronen eines künstlichen neuronalen Netzes betrachtet werden, was den Zusammenhang zum klassischen maschinellen Lernen herstellt. Auch wenn viele der erhofften Quantenvorteile sich wahrscheinlich erst mit noch ausgereifterer, fehlerkorrigierter Hardware der Zukunft manifestieren werden, eignet sich dieser Ansatz auch schon für die aktuell verfügbare, fehlerbehaftete Hardware.

Quantum Deep Hedging

Während klassisches „Deep Hedging“ deutliche Verbesserungen gegenüber expliziten Hedging-Modellen erzielen konnte und daher bereits vereinzelt in der Praxis angewandt wird, ist die Untersuchung möglicher Vorteile bei der Erweiterung auf sogenannte Quanten-neuronale Netze (QNN) ein Gegenstand hochaktueller Forschung. Dabei stehen insbesondere die Möglichkeiten der dynamischen Anpassungen im Umfeld von Marktveränderungen, die nicht notwendigen expliziten Annahmen der zugrundeliegenden Asset-Typen sowie die Option der Berücksichtigung von Extremereignissen im Vordergrund. In ► Abb. 03 wird die Einführung einer Quanten-Komponente im neuronalen Netzwerk gezeigt, indem ein klassischer Layer durch einen Quanten-Schaltkreis ersetzt wird. Ein Vorteil des gewählten Ansatzes ist die vergleichsweise geringe Anfälligkeit für Quantenrauschen.

Das Training von klassischen Layers und Quantenlayers erfolgt simultan wie in ► Abb. 01 dargestellt.

Literatur

- [1] *pfhedge Package*, Code-Referenz auf Github: <https://github.com/pfnet-research/pfhedge>
 [2] *Pytorch*, Referenz: <https://pytorch.org/>
 [3] Buehler, H. et al. [2018]: *Deep Hedging*, arXiv:1802.03042v1
 [4] *pfhedge-Erweiterung auf QNNs*, Referenz: <https://github.com/Progsprach/pfhedge>
 [5] Cherrat et al. [2023]: *Quantum Deep Hedging*, arXiv:2303.16585

Ergebnisse

Im Verlauf der Entwicklung wurden viele verschiedene Marktmodelle und Parametrisierungen verprobt. Die in ► Abb. 04 dargestellten Ergebnisse wurden mit einem Merton-Jump Modell mit großen Sprüngen und einer Lookback-Option als Derivat simuliert, weil dieser Fall besonders gut vom Deep-Hedging-Ansatz profitiert.

In unserer Analyse, bei der hybride Modelle (wie in ► Abb. 03 gezeigt) verwendet wurden, konnte eine gegenüber dem rein klassischen Fall vergleichbare Leistung erzielt werden, auch wenn sich mit den aktuell simulierbaren Größenordnungen noch kein Vorteil durch die Integration von Quantenschaltkreiskomponenten zeigte. Es gab jedoch Indizien dafür, dass sich die Vorteile der Quanten-Anteile bei Problemen größerer Dimension manifestieren könnten, vielversprechende Hinweise darauf liefern die Daten aus ► Abb. 04 für kleine Modellgrößen. Unabhängig von unserem Projekt wurden in [5] mit unterschiedlichen Methoden ähnliche Resultate erzielt, was für die Robustheit dieser Vorteile spricht.

Ein Hindernis für die Skalierung stellt in dem untersuchten Ansatz noch die sehr große Zahl an benötigten Schaltkreis-Ausführungen dar. Etwaige Effizienzsteigerungen, beispielsweise durch besseren Batching-Support, werden hier zukünftig auch die Ausführung und den Vergleich von Problemen größerer Dimension erlauben.



Autoren

Dr. Hendrik Heine
 Consultant,
 d-fine GmbH,
 Frankfurt am Main



Dr. Daniel Herr
 Senior Consultant,
 d-fine AG,
 Zürich



Dr. Daniel Ohl de Mello
 Senior Consultant,
 d-fine GmbH,
 Frankfurt am Main



Dr. Ferdinand Graf
 Partner,
 d-fine GmbH,
 Frankfurt am Main