

tbd.

2. Projektarbeit

vorgelegt am 20.11.2023

Fakultät Wirtschaft und Gesundheit

Studiengang Wirtschaftsinformatik

Kurs WWI2021F

von

JONAS MICHEL

Betreuer in der Ausbildungsstätte:

IBM
Dr. Stefan Kister
Quantum Ambassador

Unterschrift des Betreuers

DHBW Stuttgart:

Prof. Dr. Kai Holzweißig
Studiengangsleiter WWI2021F

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
1 Einleitung	1
2 Diskussion des aktuellen Stands der Forschung	3
2.1 Methodik der Literaturanalyse	3
2.2 Long Short Term Memory Modell	3
2.2.1 LSTM Netzwerk Architektur	4
2.2.2 Modell Training und Evaluation	5
2.3 Quantum Long Short Term Memory Model	6
2.3.1 Quantum Neuronale Netzwerke	7
2.3.2 Feature Map	8
2.3.3 Variational Layer (Ansatz)	9
2.3.4 Quantum Measurement	10
2.3.5 QLSTM Netzwerk Architektur	11
2.4 Analyse der Vergleichsobjekte	12
3 Darlegen der Zielsetzung und Forschungsmethodik	13
3.1 Spezifizierung der Zielsetzung	13
3.2 Forschungsmethodik	13
4 Design und Implementation des Experiments	14
4.1 Entwicklungsumgebung	14
4.2 Testdatensätze	14
4.3 Klassische LSTM Architektur Implementation	15
4.4 Quantum LSTM Architektur Implementation	15
4.4.1 Paper 1	15
4.4.2 Paper 2	15
4.5 Test Aufbau	15
4.6 Test Durchführung	15
5 Evaluation des Artefakts	16
5.1 Auswertung der Ergebnisse	16
5.2 Beantworten der Forschungsfrage	16
6 Kritische Reflexion	17
Anhang	18
Literaturverzeichnis	25

Abkürzungsverzeichnis

LSTM	Long Short Term Memory
MAE	Mean Absolute Error
MSE	Mean Squared Error
QML	Quantum Machine Learning
QNN	Quantum Neural Network
QLSTM	Quantum Long Short Term Memory
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
VQC	Variational Quantum Circuit

Abbildungsverzeichnis

1	Long Short Term Memory Architektur	4
2	Variational Quantum Circuit	7
3	Beispiel Feature Map	8
4	Beispiel Variational Layer	9
5	Quantum Long Short Term Memory Architektur	11
6	Testdatensätze	14

Tabellenverzeichnis

1 Einleitung

Für die Verarbeitung von sequentiellen Daten, werden in der Praxis Long Short Term Memory (LSTM)-Modelle eingesetzt.¹ Beispielhaft hierfür sind die Anwendung bei Video zu Text Applikationen² oder Timeseries Forecasting (Zeitreihenvorhersagung)³. LSTM Architekturen sind außerdem die Basis der für GPT-3 benutzten Transformer Modelle.⁴ In der aktuellen Literatur stellen Chen, S. Y.-C./Yoo/Fang 2022 und Yu u. a. 2023 jeweils LSTM-Architekturen vor, welche mit Quantum Computing erweitert wurden. Diese sogenannten Quantum Long Short Term Memory (QLSTM)-Architekturen, überzeugen mit einer höheren Genauigkeit und einer schnelleren Trainingszeit. Währenddessen gibt es immer mehr Unternehmen wie IBM⁵, Google⁶, Microsoft⁷ und D-Wave⁸, welche einsatzfähige Quantumcomputer entwickelt haben. Quantenfehlerverhinderung Algorithmen von IBM⁹ und Google¹⁰ beschleunigen die Entwicklung zusätzlich. Ein Quantumfortschritt lässt sich auch bei Machine Learning beobachten, Abbas u. a. 2021 zeigen, dass gewisse Quantum Neural Network (QNN)'s bessere Ergebnisse liefern und dazu schneller trainiert werden können, als vergleichbare klassische Modelle. In diversen physikalischen Experimenten haben Huang/Kueng/Preskill 2021 einen aktueller Quantenvorteil im Vergleich zu klassischen Machine Learning Modellen nachgewiesen. Bei zyklischen wellenformigen Datensätzen, wie zum Beispiel einer Sinuskurve oder einem gedämpften Oszillator, haben spezielle QLSTM Architekturen einen Vorteil gegenüber klassischen Modellen.¹¹ Quantum Machine Learning Modelle eröffnen also perspektivisch die Aussicht, aktuelle hochkomplexe physikalische und chemische Probleme zu lösen.¹²

Wie gut ein QNN performt hängt stark von der Architektur ab. Die Feature Map und der Variational Layer können beliebig angepasst werden und sind somit maßgeblich für die Genauigkeit des Modells verantwortlich.¹³ Wie sich verschiedene Feature Maps und Variational Layer auf unterschiedliche Datensätze auswirken ist hierbei noch zu untersuchen.¹⁴ Aktuell gibt es einige wissenschaftliche Artikel¹⁵ welche verschiedenen QLSTM Architekturen benutzen um spezifische Datensätze zu bearbeiten. Diese Publikationen benutzten alle unterschiedliche Feature Maps, Variational Layer und eigene Anwendungsfälle. Keine der Publikationen vergleicht die verschiedenen Architekturen miteinander oder wendet das Modell auf unterschiedliche Datensätze an.

¹Vgl. Sutskever/Vinyals/Le 2014

²Vgl. Venugopalan u. a. 2015

³Vgl. Chen, S. Y.-C./Yoo/Fang 2022

⁴Vgl. Vaswani u. a. 2017, S. 1ff.

⁵Vgl. IBM Quantum 2023

⁶Vgl. Google Quantum AI 2023

⁷Vgl. Microsoft Azure 2023

⁸Vgl. D-Wave 2023

⁹Vgl. Kim u. a. 2023

¹⁰Vgl. Acharya u. a. 2023

¹¹Vgl. Chen, S. Y.-C./Yoo/Fang 2022

¹²Vgl. Huang/Kueng/Preskill 2021, S. 1

¹³Vgl. Abbas u. a. 2021, S. 404

¹⁴Vgl. Abbas u. a. 2021, S. 407

¹⁵Vgl. Chen, S. Y.-C./Yoo/Fang 2022; Yu u. a. 2023; Qi/Yang/Chen, P.-Y. 2021

Angesichts dieser beschriebenen Problematik stellt sich die Frage, **welche Art von QLSTM-Architektur für unterschiedliche Datensätze am besten geeignet ist**. Das Ziel dieser Praxisarbeit besteht darin, vier verschiedene QLSTM-Architekturen auf zwei unterschiedliche Datensätze anzuwenden, um sie im Anschluss miteinander zu vergleichen und zu evaluieren. Hierdurch soll ermittelt werden, ob eine spezifische QLSTM-Architektur existiert, die für beide Datensätze als besonders geeignet ist, oder ob bestimmte Architekturen für spezifische Datensätze besser geeignet sind. Die Auswahl der Datensätze und Architekturen erfolgt dabei so, dass möglichst alle relevanten Aspekte dieser Fragestellung abgedeckt werden.

Forschungsmethode (Experiment) und Aufbau tbd.

2 Diskussion des aktuellen Stands der Forschung

2.1 Methodik der Literaturanalyse

Die gesamte Literaturanalyse wurde systematisch nach der Methode Webster/Watson 2002 durchgeführt. Dabei wurden die als relevant identifizierten Konzepte erfasst¹⁶:

LSTM, QLSTM, Quantum Machine Learning (QML), Variational Quantum Circuit (VQC) und Quantum Data Encoding.

Die erstellte Konzeptmatrix in Anhang 1/1 zeigt die Zusammenhänge der Konzepte untereinander. Um passende Literatur zu finden, wurde eine Liste an Suchbegriffen erstellt. Diese ist in Anhang 1/2 zu finden. Die Suchbegriffe wurden in den folgenden Datenbanken gesucht: Google Scholar¹⁷, IEEE Xplore¹⁸, ScienceDirect¹⁹ und arXiv²⁰. Die jeweils ersten 10 Suchergebnisse wurden auf Relevanz geprüft, und die passenden Artikel wurden in der Konzeptmatrix aufgenommen. Von relevanten Artikeln wurden außerdem Referenzen und Zitationen mit Connected-Papers²¹ visualisiert und ebenfalls geprüft.

Im nächsten Teil der Literaturanalyse werden nun die einzelnen Konzepte und ihre Zusammenhänge erläutert. Eine Analyse der Vergleichsobjekte findet in Kapitel 2.4 statt.

2.2 Long Short Term Memory Modell

Die erste LSTM-Netzwerk Architektur wurde von Hochreiter/Schmidhuber, Jürgen 1997 vorgestellt, um das Vanishing Gradient Problem zu lösen.²² Dieses Problem tritt hauptsächlich bei einem Recurrent Neural Network (RNN) auf.²³ LSTM-Netzwerke sind also die Weiterentwicklung der klassischen RNN-Netzwerke.²⁴ Heute bilden sie die Grundlage für sogenannte Transformer, welche für Large Language Modelle benutzt werden.²⁵

¹⁶Vgl. Webster/Watson 2002, S. 15

¹⁷<https://scholar.google.com/>

¹⁸<https://ieeexplore.ieee.org/>

¹⁹<https://www.sciencedirect.com/>

²⁰<https://arxiv.org/>

²¹<https://www.connectedpapers.com/>

²²Vgl. Van Houdt/Mosquera/Nápoles 2020, S. 5931

²³Vgl. Bengio/Frasconi/Schmidhuber, Jürgen 2003

²⁴Vgl. Hochreiter/Schmidhuber, Jürgen 1997

²⁵Vgl. Vaswani u. a. 2017

2.2.1 LSTM Netzwerk Architektur

Der Datenfluss einer LSTM-Zelle ist in Abbildung 1 dargestellt und lässt sich mathematisch durch folgende Gleichungen beschreiben:

$$f_t = \sigma(W_f \cdot v_t + b_f) \quad (2.1)$$

$$i_t = \sigma(W_i \cdot v_t + b_i) \quad (2.2)$$

$$\tilde{C}_t = \tanh(W_C \cdot v_t + b_C) \quad (2.3)$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{C}_t \quad (2.4)$$

$$o_t = \sigma(W_o \cdot v_t + b_o) \quad (2.5)$$

$$h_t = o_t * \tanh(c_t) \quad (2.6)$$

wobei σ die Sigmoid-Funktion bezeichnet und W_n klassische neuronale Netzwerke sind ($n = f, i, C, o$)²⁶

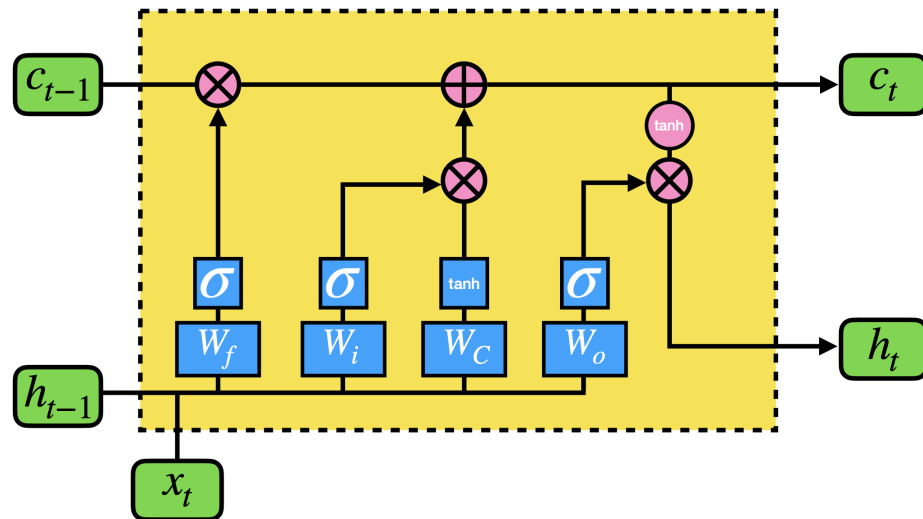


Abb. 1: LSTM Zellen Architektur entnommen aus Chen, S. Y.-C./Yoo/Fang 2022, S. 6

Die obenbeschriebenen Formeln und Zellenstruktur werden zum größten Teil genau so in der aktuell diskutierten Literatur verwendet.²⁷ In einigen Publikationen sind die Formeln leicht abgewandelt, da es auch andere LSTM-Varianten gibt.²⁸ Das ändert die obenbeschriebene Grundstruktur allerdings nicht. Diese Struktur wird auch in der Literaturanalyse von Van Houdt/Mosquera/Nápoles 2020 als klassische LSTM bezeichnet.

Die klassische LSTM-Architektur besteht aus drei Gates, welche die Eingabe x_t verarbeiten (Input Gate), eine Ausgabe h_t erzeugen (Output Gate) und den Zustand c_t der Zelle verwalten

²⁶Vgl. Chen, S. Y.-C./Yoo/Fang 2022, S. 5

²⁷Vgl. Chen, S. Y.-C./Yoo/Fang 2022; Yu u. a. 2023; Qi/Yang/Chen, P.-Y. 2021; Yan/Ouyang 2018; Fischer/Krauss 2018; Liu, Y. 2019; Sak/Senior/Beaufays 2014; Sagheer/Kotb 2019

²⁸Vgl. Van Houdt/Mosquera/Nápoles 2020, S. 5931f.

(Forget Gate).²⁹ Das Forget Gate ist in der ursprünglichen Version von Hochreiter/Schmidhuber, Jürgen 1997 nicht enthalten und wurde erst durch Gers/Schmidhuber, J. A./Cummins 2000 hinzugefügt. Durch diese drei Gates wird der Informationsfluss der LSTM-Zelle geregelt und es ist möglich, dass die Zelle Informationen über beliebige Zeitintervalle hinweg speichern kann.³⁰ Diese Informationen sind in den zwei Zuständen gespeichert, während c_t der Langzeit-Speicher und h_t der Kurzzeit-Speicher ist.³¹

Es ist möglich aus LSTM-Architekturen mehrdimensionale Netzwerke zu konstruieren.³² Sagheer/Kotb 2019 beschreiben ein Deep LSTM-Netzwerk, wobei mehrere LSTM-Zellen hintereinander geschaltet werden.³³ Deep LSTM-Netzwerke werden in der Literatur beispielsweise für Finanzprognosen³⁴ oder Bedarfsprognosen³⁵ verwendet.

Für die in Kapitle 4 aufgesetzt und durchgeführten Experimente, werden nur eindimensional Netzwerke betrachtet, da die in Kapitel 2.4 betrachteten QLSTM-Architekturen alle auf eindimensionalen Netzwerken basieren.³⁶ Auch andere in dieser Arbeit nicht analysierte QLSTM-Modelle benutzten die zuvor genannten Formeln und Zellenstruktur.³⁷

2.2.2 Modell Training und Evaluation

Bei Machine Learning Modellen ist es wichtig, dass nicht nur die Trainingsdaten korrekt vorhergesagt werden, sondern auch neuen, ungesehene Daten. Um diese Generalisierung des Modells zu erreichen, wird das Modell mit ungesehenen Testdaten evaluiert.³⁸ Wie der Datensatz in Trainings und Testdaten gesplittet wird ist Anwendungsfall abhängig. Chen, S. Y.-C./Yoo/Fang 2022 splitten den Datensatz beispielsweise bei $\frac{2}{3}$.

Die Initialisierung der Gewichtsmatrizen erfolgt in der Regel zufällig, wobei die Gewichte oft mit einer Normalverteilung initialisiert werden.³⁹ Sak/Senior/Beaufays 2014 beispielsweise initialisiert die Gewichte mit einer Normalverteilung zwischen $[-0.2, 0.2]$. Um diese zufällig initialisierten Gewichte, welche die verschiedenen LSTM-Komponenten verbinden, zu trainieren, wird ein Optimierungsverfahren angewandt. Hierbei wird eine Verlustfunktion minimiert, die die Differenz zwischen den Vorhersagen und den tatsächlichen Werten berechnet. Diese Verfahren basieren auf dem Konzept der Backpropagation erstmal vorgestellt durch Werbos 1990. Das bedeutet, dass der Zellenzustand $c(t)$ während der Backpropagation Gradienten von $y(t)$ sowie vom

²⁹Vgl. Van Houdt/Mosquera/Nápoles 2020, S. 5932

³⁰Vgl. Van Houdt/Mosquera/Nápoles 2020, S. 5931

³¹Vgl. Van Houdt/Mosquera/Nápoles 2020, S. 5932f.

³²Vgl. Sak/Senior/Beaufays 2014

³³Vgl. Sagheer/Kotb 2019, S. 206

³⁴Vgl. Liu, Y. 2019; Fischer/Krauss 2018; Yan/Ouyang 2018

³⁵Vgl. Rodrigues/Markou/Pereira 2019

³⁶Vgl. Chen, S. Y.-C./Yoo/Fang 2022; Yu u. a. 2023; Qi/Yang/Chen, P.-Y. 2021

³⁷Vgl. Cao u. a. 2023

³⁸Vgl. Panesar 2021, S. 191

³⁹Vgl. Chen, S. Y.-C./Yoo/Fang 2022; Sak/Senior/Beaufays 2014

nächsten Zellenzustand $c(t + 1)$ erhält.⁴⁰ Das führt dazu, dass die Gradienten akkumuliert werden, um alle Gewichtsmatrizen bestmöglich anzupassen. Basierend auf dieser Backpropagation gibt es verschiedene Optimierungsverfahren wie beispielsweise Stochastic Gradient Descent oder Adam⁴¹.

Ist ein Modell trainiert, wird die Qualität der Vorhersagen bzw. die Generalisierung evaluiert. Für die Evaluierung der Testergebnisse, stehen verschiedene Fehlermetriken zur Verfügung. LSTM Publikationen benutzten häufig den Root Mean Squared Error (RMSE)/Mean Squared Error (MSE)⁴² und den Mean Absolute Error (MAE)⁴³. Welche Metriken am besten geeignet sind, hängt stark von der Modellarchitektur und -aufgabe ab.⁴⁴

Formeln der Evaluationsmetriken:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

2.3 Quantum Long Short Term Memory Model

QLSTM-Modelle liefern teilweise einen Vorteil gegenüber klassischen Modellen, indem sie mit weniger Epochs bessere Ergebnisse liefern⁴⁵ oder generell höhere Genauigkeit erzielen⁴⁶. Im Gegensatz zu klassischen Modellen, sind die Gewichtsmatrizen durch sogenannte Quantum Layer ersetzt, diese Layer beinhalten die zu trainierenden Parameter.

Hybride Quantum Machine Learning Modelle werden wie klassische Modelle trainiert und evaluiert (siehe Kapitle 2.2.2).⁴⁷ Dies wird durch verschiedene Schnittstellen ermöglicht. Die Qiskit-Bibliothek bietet einen TorchConnector⁴⁸, welcher es ermöglicht, hybride Modelle mit PyTorch zu trainieren. Tensorflow Quantum⁴⁹ bietet eine ähnliche Funktion in Kooperation mit Google Cirq.

⁴⁰Vgl. Van Houdt/Mosquera/Nápoles 2020, S. 5933

⁴¹Vgl. Kingma/Ba 2014

⁴²Vgl. Yu u. a. 2023; Sagheer/Kotb 2019; Chen, S. Y.-C./Yoo/Fang 2022

⁴³Vgl. Chen, S. Y.-C./Yoo/Fang 2022

⁴⁴Vgl. Panesar 2021, S. 201

⁴⁵Vgl. Chen, S. Y.-C./Yoo/Fang 2022

⁴⁶Vgl. Yu u. a. 2023

⁴⁷Vgl. Kwak u. a. 2021, S. 414

⁴⁸Qiskit Dokumentation: TorchConnector 2023

⁴⁹TensorFlow Quantum Dokumentation 2023

Aus diesem Grund wird im folgenden Kapitel die Funktionsweise der Quantum Komponenten und die QLSTM-Architektur erläutert, nicht aber der Trainings- und Evaluierungsprozess.

2.3.1 Quantum Neuronale Netzwerke

Es gibt verschiedene Ansätze für Quantum Machine Learning Modelle. Echo-State Netzwerke vergleichen zwei Zustände und eignen sich daher für Klassifikationsprobleme.⁵⁰ Während Quantum Neuronale Netzwerke, wie klassische Neuronale Netzwerke verschaltet sind und Gewichtsparameter besitzen.⁵¹

Quantum Neuronale Netzwerke sind spezielle Quantum Circuits, sogenannte VQCs.⁵² Ein Quantum Circuit besteht aus verschalteten Quantum Gates, welche die Zustände der Qubits verändern.⁵³ Welche Art von Quantum Gates angewendet werden können wird von Williams/Clearwater u. a. 1998, S. 51ff. ausführlich diskutiert. Qubits bilden die Grundlage von Quantum Computern und können Zustände zwischen $|0\rangle$ und $|1\rangle$ annehmen.⁵⁴ Jedes Qubit hat seinen eigenen Quantum Zustand, welcher sich messen lässt. Die mathematischen Grundlagen, hinter Qubits und ihren Zuständen, werden von Williams/Clearwater u. a. 1998, S. 8ff. Kwak u. a. 2021, S. 413 und Weigold u. a. 2021, S. 97 erklärt. Die quadrierten Quantumzustände geben die Wahrscheinlichkeit an, mit der ein Qubit in einem bestimmten Zustand (0 oder 1) gemessen wird.⁵⁵ Dies gilt auch für mehrere Qubits, wobei die Wahrscheinlichkeit für jeden Zustand (Bsp. 00, 10, 01, 11) berechnet wird.⁵⁶

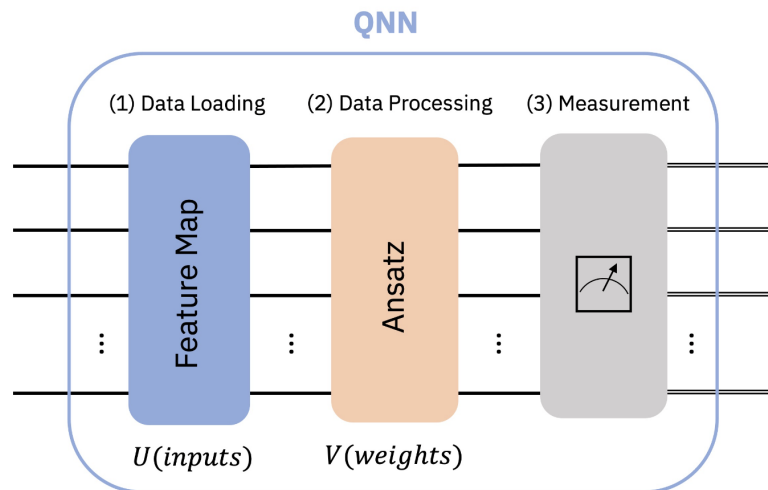


Abb. 2: VQC entnommen aus Qiskit Textbook: QNN 2023

⁵⁰Vgl. Mitarai u. a. 2018, S. 2

⁵¹Vgl. Monaco u. a. 2023, S. 2

⁵²Vgl. Kwak u. a. 2021, S. 413

⁵³Vgl. Williams/Clearwater u. a. 1998, s. 51

⁵⁴Vgl. Kwak u. a. 2021, S. 413

⁵⁵Vgl. Weigold u. a. 2021, S. 96

⁵⁶Vgl. Chen, S. Y.-C./Yoo/Fang 2022, S. 8

Ein VQC für ein Quantum Neuronales Netzwerk besteht, wie in Abbildung 2 zu sehen ist aus 3 Teilen. Die genaue Benennung der Teile variiert je nach Publikation, jedoch sind diese immer gleich aufgebaut.⁵⁷ In dieser Arbeit wird die Bezeichnung gemäß der Darstellung in Abbildung 2 verwendet. Die Feature Map ist für die Codierung der Eingabedaten verantwortlich und wird deshalb auch Data Encoding Layer genannt. Wie die Feature Map aufgebaut ist, hängt von den Eingabedaten ab.⁵⁸ Der Ansatz, auch Variational Layer genannt, ist für die Gewichtung der Eingabedaten zuständig. Die Gewichtung wird durch die Parameter θ bestimmt.⁵⁹ Nach der Gewichtung werden die Daten durch den Measurement Layer gemessen und als Ergebnis ausgegeben. Das errechnete Ergebniss kann im Anschluss auf einem klassischen Computer weiter verarbeitet werden.⁶⁰

Die Feature Map und der Variational Layer werden in den folgenden Kapiteln genauer erläutert (siehe Kapitel 2.3.2 Feature Map; Kapitel 2.3.3 Variational Layer (Ansatz)).

2.3.2 Feature Map

Die Feature Map sorgt dafür, dass die klassischen Eingabedaten in Quantumzustände (beschrieben in Kapitel 2.3.1) übertragen werden.⁶¹ Grundsätzlich haben Feature Maps Quantum Gates, welche als nicht trainierbare Parameter, die Eingabedaten des VQC's erhalten.⁶² Dadurch kommt es zu einer Veränderung der Quantumzustände und somit zu einer Veränderung der Wahrscheinlichkeiten der einzelnen Zustände.⁶³

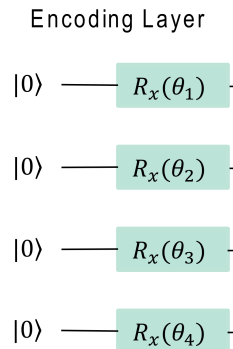


Abb. 3: Beispiel Feature Map entnommen aus Yu u. a. 2023, S. 3

In Abbildung 3 ist eine Feature Map zu sehen, welche die Eingabedaten $x_1, x_2 \dots x_4$, durch eine einzelne Pauli-Rotation in einen Quantum Zustand überträgt. Solch ein Pauli-Rotation kann nur eine einfache Sinuskurve lernen, während wiederholte Pauli-Codierungen das Abbildungs-

⁵⁷Vgl. McClean u. a. 2018; Hu u. a. 2022; Chen, S. Y.-C./Yoo/Fang 2022; Yu u. a. 2023; Cao u. a. 2023

⁵⁸Vgl. Cao u. a. 2023, S. 2

⁵⁹Vgl. Yu u. a. 2023, S. 4

⁶⁰Vgl. Chen, S. Y.-C./Yoo/Fang 2022, S. 9

⁶¹Vgl. Yu u. a. 2023, S. 3f.

⁶²Vgl. Hu u. a. 2022, S. 2

⁶³Vgl. Weigold u. a. 2021, S. 96

spektrum linear erweitern.⁶⁴ Pauli-Rotationen sind die Art von Gates, welche um die x , y oder z -Achse rotieren.⁶⁵

Welche Gates in einer Feature Maps verwendet werden, hängt von den Eingabedaten ab.⁶⁶ Es gibt verschiedene Arten des Encodings: Basis, Amplitude, QuAM und Angle, welche verschiedene Ergebnisse liefern.⁶⁷ Die später betrachteten QLSTM-Architekturen verwenden hauptsächlich das Angle Encoding.⁶⁸ Aufgrund der Einfachheit ist diese Art des Encodings am meisten verbreitet.⁶⁹ Auch andere relevante Publikationen über QNN's benutzen Angle Encoding.⁷⁰

Angle Encoding benötigt n Qubits, um n Eingabedaten zu codieren. Die Qubits werden durch Rotationsgates in einen bestimmten Zustand gebracht, welcher durch die Eingabedaten als Parameter bestimmt wird.⁷¹

2.3.3 Variational Layer (Ansatz)

Sind die klassischen Daten durch die Feature Map in Quantumzustände übertragen, werden diese mit den Ansatz, auch Variational Layer genannt, weiter verarbeitet. Dieser Layer besteht aus mehreren Quantum Gates, welche durch trainierbare Parameter, die Gewichtung der geladenen Eingabedaten bestimmen.⁷² Aufgrund der trainierbaren Parameter,⁷³ funktioniert der Variational Layer ähnlich wie klassischen Netzwerken.⁷⁴ Wie in Abbildung 4 zu sehen, sind die verschiedenen Qubits miteinander verschaltet. Diese Verknüpfung wird mithilfe von Controlled-Not-Gates realisiert und gewährleistet die Verbindung aller Qubits, also sämtlicher Datenpunkte, miteinander.⁷⁵ Außerdem kommen Rotationsgates zum Einsatz, welche mit Parametern die Gewichtung der einzelnen Qubits bestimmen.

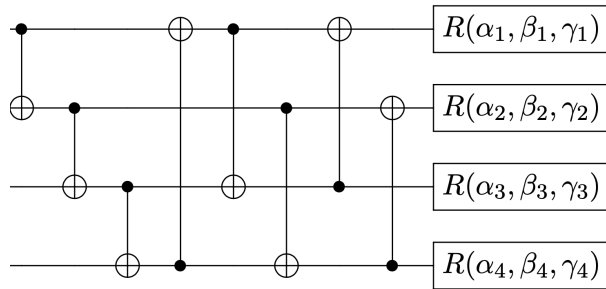


Abb. 4: Beispiel Variational Layer entnommen aus Chen, S. Y.-C./Yoo/Fang 2022, S. 7

⁶⁴Vgl. Schuld/Sweke/Meyer 2021, S. 2

⁶⁵Vgl. Williams/Clearwater u. a. 1998, S. 81

⁶⁶Vgl. Weigold u. a. 2022, S. 9

⁶⁷Vgl. Weigold u. a. 2022, S. 8

⁶⁸Vgl. Chen, S. Y.-C./Yoo/Fang 2022; Yu u. a. 2023; Cao u. a. 2023

⁶⁹Vgl. Ovalle-Magallanes u. a. 2023, S. 4

⁷⁰Vgl. Liu, J. u. a. 2021; Mari u. a. 2020; Henderson u. a. 2020; LaRose/Coyle 2020

⁷¹Vgl. Ovalle-Magallanes u. a. 2023, S. 4

⁷²Vgl. Hu u. a. 2022, S. 2

⁷³Vgl. remote Cao u. a. 2023, S. 3

⁷⁴Vgl. Kwak u. a. 2021, S. 414

⁷⁵Vgl. Yu u. a. 2023, S. 4

Im abgebildeten Beispiel sind dies die trainierbaren Parameter α , β und γ benannt.

Wie die Qubits miteinander verschaltet sind und welche Rotationsgates verwendet werden hängt vom Use-Case ab.⁷⁶ Grundsätzlich ist es möglich Qubits und Model Komplexität hinzufügen, um die Genauigkeit des Modells zu erhöhen.⁷⁷ Es ist möglich den Variational Layer mehrfach zu schalten, um die Tiefe des VQC's zu erhöhen.⁷⁸ Sim/Johnson/Aspuru-Guzik 2019, S. 8 diskutieren die Eignung verschiedener Circuits und untersuchen die Ausdruckswerte, beziehungsweise die Fähigkeit einen möglichst großen Datenraum abzubilden.

Welche verschiedenen Variational Layer in aktuellen Publikationen verwendet werden, wird in Kapitel 2.4 erläutert.

2.3.4 Quantum Measurement

Der Measurement Layer ist für die Ausgabe des VQC's verantwortlich, indem die Qubits gemessen und in klassische Daten umgewandelt werden.⁷⁹ Die Pauli matrix σ_z (siehe Gleichung 2.7) wird verwendet, um die Erwartungswerte der einzelnen Qubits zu berechnen.⁸⁰ Nach Yu u. a. 2023, ist es möglich die Erwartungswerte der Qubits mit Frameworks zu berechnen, welche die Quantenfunktionen simulieren.

$$\sigma_Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (2.7)$$

Busch u. a. 2016 Braginsky/Braginskiĭ/Khalili 1995 und Wiseman/Milburn 2010 setzten sich genauer mit den Thema der Quantenmessung auseinander.

⁷⁶Vgl. Yu u. a. 2023; Chen, S. Y.-C./Yoo/Fang 2022; Cao u. a. 2023; Sim/Johnson/Aspuru-Guzik 2019

⁷⁷Vgl. Cao u. a. 2023, S. 2

⁷⁸Vgl. Chen, S. Y.-C./Yoo/Fang 2022, S. 9

⁷⁹Vgl. Cao u. a. 2023, S. 2

⁸⁰Vgl. Yu u. a. 2023, S. 4

2.3.5 QLSTM Netzwerk Architektur

Der Datenfluss bei einer QLSTM-Zelle ist in Abbildung 5 dargestellt und lässt sich mathematisch durch folgende Gleichungen beschreiben:

$$f_t = \sigma(VQC_1(v_t)) \quad (2.8)$$

$$i_t = \sigma(VQC_2(v_t)) \quad (2.9)$$

$$\tilde{C}_t = \tanh(VQC_3(v_t)) \quad (2.10)$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{C}_t \quad (2.11)$$

$$o_t = \sigma(VQC_4(v_t)) \quad (2.12)$$

$$h_t = VQC_5(o_t * \tanh(c_t)) \quad (2.13)$$

$$y_t = VQC_6(h_t * \tanh(c_t)) \quad (2.14)$$

Wobei Input für VQC_1 bis VQC_4 v_t , die Vektor Verkettung von x_t und h_{t-1} , ist.

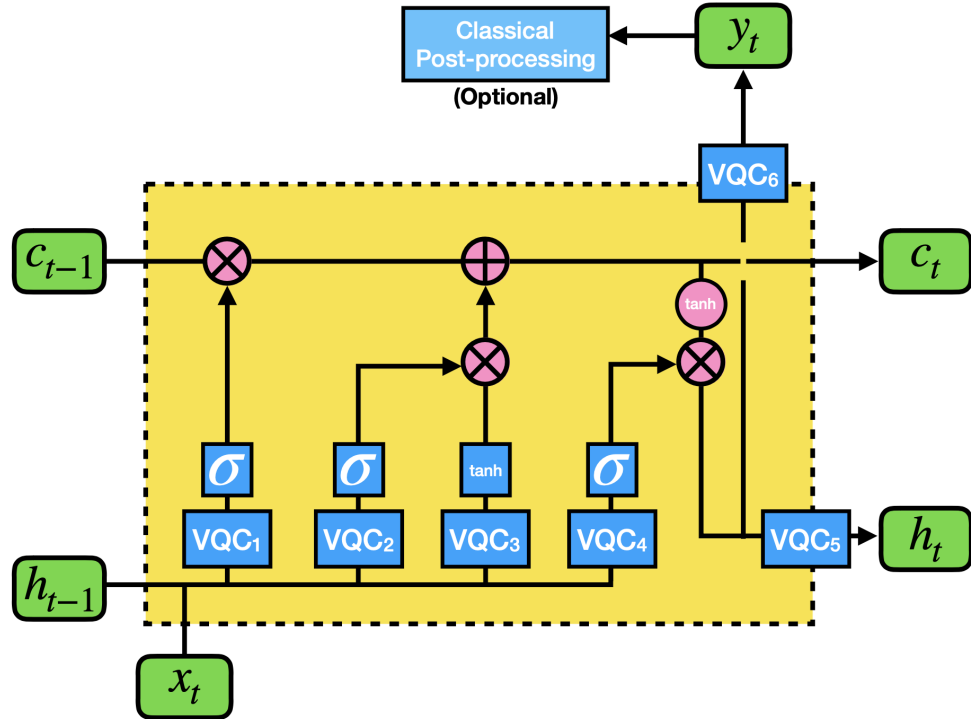


Abb. 5: QLSTM Zellen Architektur entnommen aus Chen, S. Y.-C./Yoo/Fang 2022, S. 10

Im Vergleich zur klassischen LSTM-Zellen Struktur, gibt es keine Gewichtsmatrizen mehr, sondern nur noch VQC's. Diese enthalten die trainierbaren Parameter (siehe Kapitel 2.3.3). In der hier abgebildeten Architektur, gibt es sechs VQC's, welche die Gewichtung der Eingabedaten übernehmen. Aus diesem Grund, ist der Output y_t und der Hidden State h_t nicht gleich wie bei klassischen LSTM-Architektur. Dies hängt aber von der gewählten Anzahl der VQC's ab. Während Chen, S. Y.-C./Yoo/Fang 2022 noch sechs VQC's vorschlägt, schlägt Chen, S. Y.-C. 2023

nur noch fünf vor. Es gibt auch Architekturen mit nur vier VQC's.⁸¹ In der Publikation von Cao u. a. 2023, werden ebenfalls 6 VQC's verwendet, jedoch werden diese mit klassischen Layern kombiniert.⁸² Eine Abbildung dieser Architektur ist in Anhang xy zu sehen.

Werden weniger VQC's verwendet, ist es möglich, dass der Output y_t und der Hidden State h_t gleich sind. Der sechste VQC wird in diesem Fall nicht benötigt, da der Hidden State h_t bereits die gewichteten Eingabedaten enthält.⁸³ Gibt es nur vier VQC's, wird der Hidden State h_t nicht gewichtet und der Output y_t ist gleich dem Hidden State h_t .⁸⁴

2.4 Analyse der Vergleichsobjekte

die anzahl der vqcs in den vergleichsobjekten

depth der variational layer

feature maps im vergleich

variational layer im vergleich anazhl der parameter hier!

zu sehen sind alle am ende! im anhang

später wird noch geschaut wie viele parameter es tatsäch gibt!

⁸¹Vgl. Yu u. a. 2023, S. 4

⁸²Vgl. Cao u. a. 2023, S. 11

⁸³Vgl. Chen, S. Y.-C. 2023, S. 3

⁸⁴Vgl. Yu u. a. 2023, S. 4

3 Darlegen der Zielsetzung und Forschungsmethodik

max. 2 Seiten

3.1 Spezifizierung der Zielsetzung

was wollen wir jetzt genau machen grober ausblick wie

3.2 Forschungsmethodik

wie funktioniert diese und warum wird sie benutzt? (Realtitsbsp)

4 Design und Implementation des Experiments

bla bla bkla kommen wir nun zur implementierung und Entwicklung hier wichtig aller code steht zur verfügung und kann gerunt werden. Alle benötigten schritte sind in der Arbeit vermekrt (umgebung und so weiter...)

4.1 Entwicklungsumgebung

Um die im folgenden Kapitel 4 beschriebene Implementation lokal durchzuführen, ist es erforderlich, eine geeignete Entwicklungsumgebung aufzusetzen. Die Programmierung erfolgte mithilfe von Visual Studio Code als Code-Editor. Die Python-Umgebung wurde unter Einsatz des Anaconda Navigators erstellt. Der gesamte Quellcode basiert auf der Python Version 3.9.16. Die wichtigsten Bibliotheken für eine erfolgreiche Implementation des nachfolgenden Artefaktes sind Qiskit und PyTorch. Die Qiskit Version 0.43.0 ist notwendig, da in anderen Versionen von Qiskit Syntaxänderungen auftreten können.

Im Anhang 3 finden sich eine umfassende Aufstellung aller Abhängigkeiten und Bibliotheken, inklusive ihrer jeweiligen Versionsnummern, die während der Entwicklung und Ausführung des Artefakts zum Einsatz kamen. Zusätzlich sind dort detaillierte Informationen zur Entwicklungsumgebung sowie zur verwendeten Software verfügbar.

4.2 Testdatensätze

Zackige und runde Daten - erzeugt in: dort ausführlich dokuemntiert aber hier nochmal die websites: Doku bei CODE!

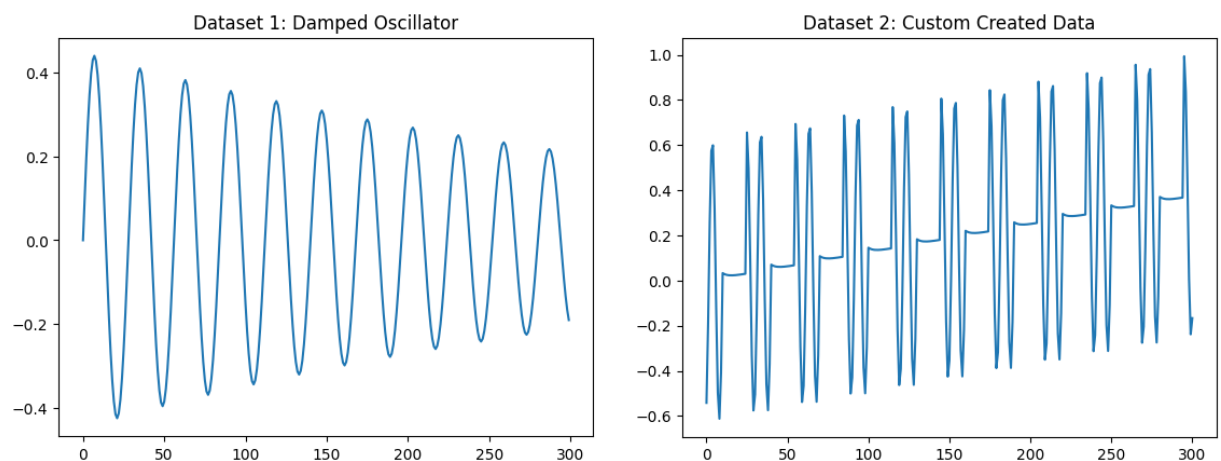


Abb. 6: Testdatensätze

4.3 Klassische LSTM Architektur Implementation

als standard

4.4 Quantum LSTM Architektur Implementation

4.4.1 Paper 1

4.4.2 Paper 2

4.5 Test Aufbau

4.6 Test Durchführung

5 Evaluation des Artefakts

5.1 Auswertung der Ergebnisse

5.2 Beantworten der Forschungsfrage

6 Kritische Reflexion

2 Seiten pa

- Zielsetzung der Arbeit kurz einführen
- Kritische Reflexion der Ergebnisse und Methodik
- Implikationen der Arbeit für Theorie und Praxis
- Ausblick

Anhang

Anhangverzeichnis

Anhang 1	Literaturanalyse	19
Anhang 1/1	Konzeptmatrix	19
Anhang 1/2	Suchbegriffe	21
Anhang 2	Visualisierungen	22
Anhang 2/1	Test Quantum Circuits	22
Anhang 2/2	Testergebnisse: 4 QLSTM im Vergleich	22
Anhang 3	Entwicklungsumgebung	24

Anhang 1: Literaturanalyse

Anhang 1/1: Konzeptmatrix

Um die gesamte Matrix übersichtlicher zu gestalten wurden die gängigen Abkürzungen benutzt. Quantum Data Encoding ist mit QDE abgekürzt, diese Abkürzung ist nicht im Abkürzungsverzeichnis zu finden, da sie nur für diese Tabelle benutzt wird.

Unter Artikel ist die bibtex ID des Artikels zu finden. Diese kann benutzt werden um den Artikel in der Literaturdatenbank zu finden.

Artikel	Konzepte				
	LSTM	QLSTM	QML	VQC	QDE
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					
xy					

[illegible]

Anhang 1/2: Suchbegriffe

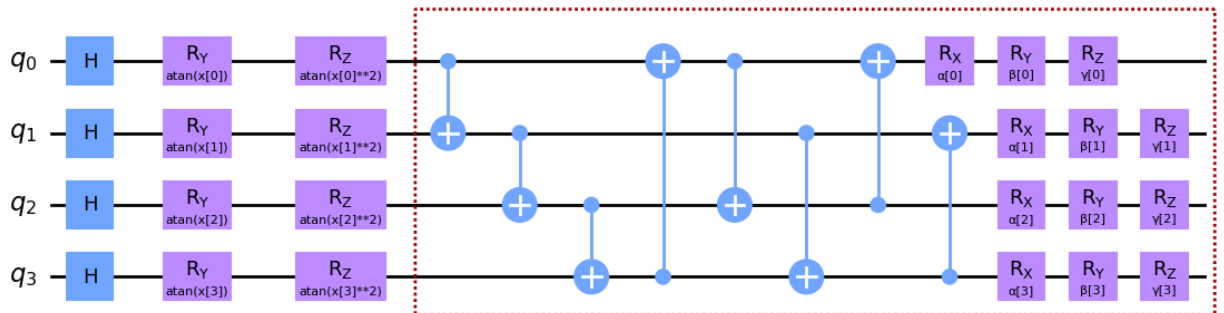
Suchbegriffe	Konzepte					
	LSTM	QLSTM	QML	VQC	QDE	Synonyme
xy						
xy						
xy						
xy						
xy						
xy						
xy						

Anhang 2: Visualisierungen

Anhang 2/1: Test Quantum Circuits

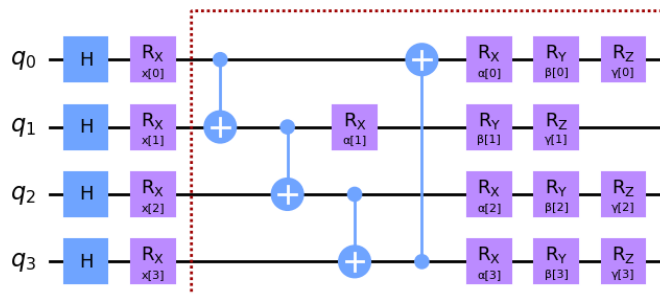
Dieses Anhangskapitel, zeigt die Visualisierung aller getesteten VQCs. Die Quelle der entnahme ist in dem Titel der Abbildung angegeben. Die letzte Abbildung zeigt einen neu modifizierten VQC, basierend auf bereits vorhandenen Publikationen. Die Überschriften der Abbildungen, sind im Code verwendeten Variablennamen. Die Publikationen wurden zufällig durchnummeriert. Aller erstellten Visualisierung wurden im Notebook `pa2_code/visual_q_circuits.ipynb` erstellt.

Paper 1:



Test-VQC 1 entnommen aus Chen, S. Y.-C./Yoo/Fang 2022

Paper 2:



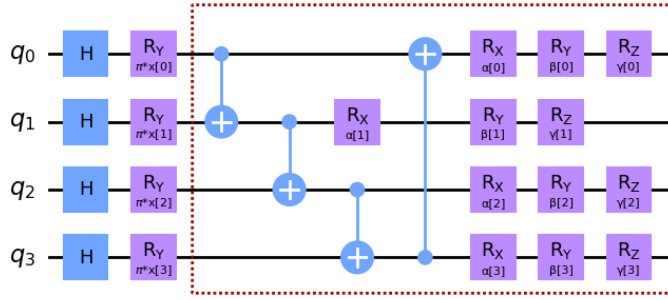
Test-VQC 2 entnommen aus Yu u. a. 2023

Paper 3:

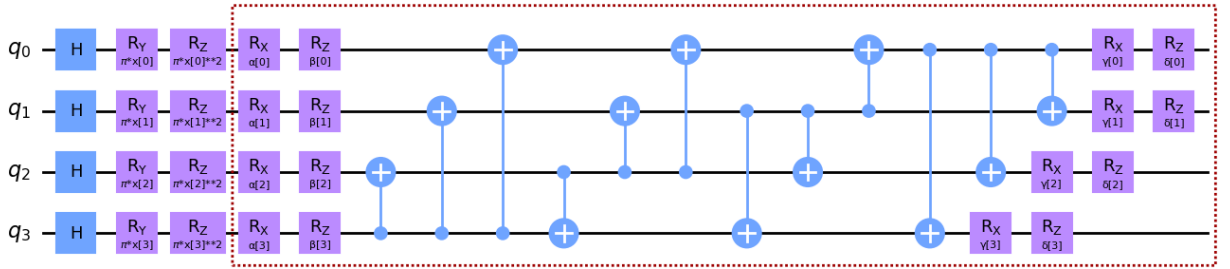
My_Own:

Anhang 2/2: Testergebnisse: 4 QLSTM im Vergleich

Lorem Ipsum ist ein einfacher Platzhaltertext für den Druck- und Satzindustrie. Lorem Ipsum hat im Wesentlichen die Norm des Industriestandards seit dem 1500s, als ein unbekannter Drucker eine Hand voll von Typen nahm und sie durcheinander warf, um ein Musterbuch zu machen.



Test-VQC 3 entnommen aus Qi/Yang/Chen, P.-Y. 2021



Test-VQC 4 entnommen aus Sim/Johnson/Aspuru-Guzik 2019 und Chen, S. Y.-C./Yoo/Fang 2022

Es hat nicht nur fünf Jahrhunderte überlebt, sondern auch den Sprung in die elektronische Schriftbearbeitung, ohne wesentlich zu ändern. Es wurde in den 1960s mit dem Aufkommen von Letraset-Blättern mit Lorem Ipsum-Passagen und mehr kürzlich mit Desktop-Publishing-Software wie Aldus PageMaker einschließlich Versionen von Lorem Ipsum popularisiert.

Anhang 3: Entwicklungsumgebung

Hardware und Software

MacBook 2019; 2,4 GHz 8-Core Intel Core i9, 32 GB RAM

Anaconda Navigator 2.3.2

Visual Studio Code: 1.82.2 (Universal)

Python Umgebung

Bibliothek	Version
ipykernel	6.23.1
ipython	8.13.2
ipywidgets	8.0.6
matplotlib	3.7.1
matplotlib-inline	0.1.6
numpy	1.23.5
pandas	2.0.1
pip	23.1.2
qiskit	0.43.0
qiskit-aer	0.12.0
qiskit-ibmq-provider	0.20.2
qiskit-machine-learning	0.6.1
qiskit-terra	0.24.0
torch	2.0.1
torchmetrics	0.11.4

Literaturverzeichnis

- Abbas, A./Sutter, D./Zoufal, C./Lucchi, A./Figalli, A./Woerner, S. (2021): The power of quantum neural networks. In: *Nature Computational Science* 1.6, S. 403–409. ISSN: 2662-8457. DOI: 10.1038/s43588-021-00084-1. URL: <https://doi.org/10.1038/s43588-021-00084-1>.
- Acharya, R. u. a. (2023): Suppressing quantum errors by scaling a surface code logical qubit. In: *Nature* 614.7949, S. 676–681. DOI: 10.1038/s41586-022-05434-1. URL: <https://doi.org/10.1038/s41586-022-05434-1>.
- Bengio, Y./Frasconi, P./Schmidhuber, Jfirgen (2003): Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies. In: *A Field Guide to Dynamical Recurrent Neural Networks*.
- Braginsky, V. B./Braginskii, V. B./Khalili, F. Y. (1995): Quantum measurement. Cambridge University Press.
- Busch, P./Lahti, P./Pellonpää, J.-P./Ylinen, K. (2016): Quantum measurement. Springer.
- Cao, Y./Zhou, X./Fei, X./Zhao, H./Liu, W./Zhao, J. (2023): Linear-layer-enhanced quantum long short-term memory for carbon price forecasting. In: *Quantum Machine Intelligence* 5. DOI: 10.1007/s42484-023-00115-2.
- Chen, S. Y.-C. (2023): Quantum Deep Recurrent Reinforcement Learning. In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, S. 1–5. DOI: 10.1109/ICASSP49357.2023.10096981.
- Chen, S. Y.-C./Yoo, S./Fang, Y.-L. L. (2022): Quantum Long Short-Term Memory. In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, S. 8622–8626. DOI: 10.1109/ICASSP43922.2022.9747369.
- D-Wave (2023): Unlock the Power of Practical Quantum Computing Today. URL: <https://www.dwavesys.com/> (Abruf: 29.09.2023).
- Fischer, T./Krauss, C. (2018): Deep learning with long short-term memory networks for financial market predictions. In: *European Journal of Operational Research* 270.2, S. 654–669. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2017.11.054>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221717310652>.
- Gers, F. A./Schmidhuber, J. A./Cummins, F. A. (2000): Learning to Forget: Continual Prediction with LSTM. In: *Neural Comput.* 12.10, S. 2451–2471. ISSN: 0899-7667. DOI: 10.1162/089976600300015015. URL: <https://doi.org/10.1162/089976600300015015>.
- Google Quantum AI (2023): Explore the possibilities of quantum. URL: <https://quantumai.google/> (Abruf: 29.09.2023).
- Henderson, M./Shakya, S./Pradhan, S./Cook, T. (2020): Quadvolutional neural networks: powering image recognition with quantum circuits. In: *Quantum Machine Intelligence* 2.1, S. 2. ISSN: 2524-4914. DOI: 10.1007/s42484-020-00012-y. URL: <https://doi.org/10.1007/s42484-020-00012-y>.

- Hochreiter, S./Schmidhuber, Jürgen (1997):** Long Short-Term Memory. In: *Neural Computation* 9.8, S. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
- Hu, Z./Dong, P./Wang, Z./Lin, Y./Wang, Y./Jiang, W. (2022):** Quantum Neural Network Compression. In: *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design. ICCAD '22*. San Diego, California: Association for Computing Machinery. ISBN: 9781450392174. DOI: 10.1145/3508352.3549382. URL: <https://doi.org/10.1145/3508352.3549382>.
- Huang, H.-Y./Kuang, R./Preskill, J. (2021):** Information theoretic bounds on quantum advantage in machine learning. In: *Physical review letters* 126 19, S. 190505. DOI: 10.1103/PhysRevLett.126.190505.
- IBM Quantum (2023):** Now entering the era of quantum utility. URL: <https://www.ibm.com/quantum> (Abruf: 29.09.2023).
- Kim, Y./Eddins, A./Anand, S./Wei, K. X./van den Berg, E./Rosenblatt, S./Nayfeh, H./Wu, Y./Zaletel, M./Temme, K./Kandala, A. (2023):** Evidence for the utility of quantum computing before fault tolerance. In: *Nature* 618.7965, S. 500–505. ISSN: 1476-4687. DOI: 10.1038/s41586-023-06096-3. URL: <https://doi.org/10.1038/s41586-023-06096-3>.
- Kingma, D./Ba, J. (2014):** Adam: A Method for Stochastic Optimization. In: *International Conference on Learning Representations*.
- Kwak, Y./Yun, W. J./Jung, S./Kim, J. (2021):** Quantum Neural Networks: Concepts, Applications, and Challenges. In: *2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN)*, S. 413–416. DOI: 10.1109/ICUFN49451.2021.9528698.
- LaRose, R./Coyle, B. (2020):** Robust data encodings for quantum classifiers. In: *Phys. Rev. A* 102 (3), S. 403–409. DOI: 10.1103/PhysRevA.102.032420. URL: <https://link.aps.org/doi/10.1103/PhysRevA.102.032420>.
- Liu, J./Lim, K. H./Wood, K. L./Huang, W./Guo, C./Huang, H.-L. (2021):** Hybrid quantum-classical convolutional neural networks. In: *Science China Physics, Mechanics and Astronomy* 64.9, S. 290311. ISSN: 1869-1927. DOI: 10.1007/s11433-021-1734-3. URL: <https://doi.org/10.1007/s11433-021-1734-3>.
- Liu, Y. (2019):** Novel volatility forecasting using deep learning–Long Short Term Memory Recurrent Neural Networks. In: *Expert Systems with Applications* 132, S. 99–109. ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2019.04.038>. URL: <https://www.sciencedirect.com/science/article/pii/S0957417419302635>.
- Mari, A./Bromley, T. R./Izaac, J./Schuld, M./Killoran, N. (2020):** Transfer learning in hybrid classical-quantum neural networks. In: *Quantum* 4, S. 340. DOI: 10.22331/q-2020-10-09-340. URL: <https://doi.org/10.22331/q-2020-10-09-340>.
- McClean, J. R./Boixo, S./Smelyanskiy, V. N./Babbush, R./Neven, H. (2018):** Barren plateaus in quantum neural network training landscapes. In: *Nature Communications* 9.1, S. 4812. ISSN: 2041-1723. DOI: 10.1038/s41467-018-07090-4. URL: <https://doi.org/10.1038/s41467-018-07090-4>.
- Microsoft Azure (2023):** Azure Quantum. URL: <https://azure.microsoft.com/de-de/solutions/quantum-computing> (Abruf: 29.09.2023).

- Mitarai, K./Negoro, M./Kitagawa, M./Fujii, K. (2018): Quantum circuit learning. In: *Phys. Rev. A* 98 (3), S. 032309. DOI: 10.1103/PhysRevA.98.032309. URL: <https://link.aps.org/doi/10.1103/PhysRevA.98.032309>.
- Monaco, S./Kiss, O./Mandarino, A./Vallecora, S./Grossi, M. (2023): Quantum phase detection generalization from marginal quantum neural network models. In: *Phys. Rev. B* 107 (8), S. L081105. DOI: 10.1103/PhysRevB.107.L081105. URL: <https://link.aps.org/doi/10.1103/PhysRevB.107.L081105>.
- Ovalle-Magallanes, E./Alvarado-Carrillo, D. E./Avina-Cervantes, J. G./Cruz-Aceves, I./Ruiz-Pinales, J. (2023): Quantum angle encoding with learnable rotation applied to quantum-classical convolutional neural networks. In: *Applied Soft Computing* 141, S. 110307. ISSN: 1568-4946. DOI: <https://doi.org/10.1016/j.asoc.2023.110307>. URL: <https://www.sciencedirect.com/science/article/pii/S1568494623003253>.
- Panesar, A. (2021): Evaluating Machine Learning Models. In: *Machine Learning and AI for Healthcare : Big Data for Improved Health Outcomes*. Berkeley, CA: Apress, S. 189–205. ISBN: 978-1-4842-6537-6. DOI: 10.1007/978-1-4842-6537-6_7. URL: https://doi.org/10.1007/978-1-4842-6537-6_7.
- Qi, J./Yang, C.-H. H./Chen, P.-Y. (2021): QTN-VQC: An End-to-End Learning framework for Quantum Neural Networks. arXiv: 2110.03861 [quant-ph].
- Qiskit Dokumentation: TorchConnector (2023): TorchConnector. URL: https://qiskit.org/ecosystem/machine-learning/stubs/qiskit_machine_learning.connectors.TorchConnector.html (Abruf: 11.10.2023).
- Qiskit Textbook: QNN (2023): Quantum Neural Networks. URL: https://qiskit.org/ecosystem/machine-learning/tutorials/01_neural_networks.html (Abruf: 11.10.2023).
- Rodrigues, F./Markou, I./Pereira, F. C. (2019): Combining time-series and textual data for taxi demand prediction in event areas: A deep learning approach. In: *Information Fusion* 49, S. 120–129. ISSN: 1566-2535. DOI: <https://doi.org/10.1016/j.inffus.2018.07.007>. URL: <https://www.sciencedirect.com/science/article/pii/S1566253517308175>.
- Sagheer, A./Kotb, M. (2019): Time series forecasting of petroleum production using deep LSTM recurrent networks. In: *Neurocomputing* 323, S. 203–213. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2018.09.082>. URL: <https://www.sciencedirect.com/science/article/pii/S0925231218311639>.
- Sak, H./Senior, A. W./Beaufays, F. (2014): Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In: *INTERSPEECH*, S. 338–342.
- Schuld, M./Sweke, R./Meyer, J. J. (2021): Effect of data encoding on the expressive power of variational quantum-machine-learning models. In: *Phys. Rev. A* 103 (3), S. 032430. DOI: 10.1103/PhysRevA.103.032430. URL: <https://link.aps.org/doi/10.1103/PhysRevA.103.032430>.
- Sim, S./Johnson, P. D./Aspuru-Guzik, A. (2019): Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms. In: *Advanced Quantum Technologies* 2.12, S. 1900070. DOI: <https://doi.org/10.1002/qute.201900070>.

- eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/qute.201900070>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/qute.201900070>.
- Sutskever, I./Vinyals, O./Le, Q. V. (2014)**: Sequence to Sequence Learning with Neural Networks. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'14. Montreal, Canada: MIT Press, S. 3104–3112.
- TensorFlow Quantum Dokumentation (2023)**: TensorFlow Quantum: GitHub Repo. URL: <https://github.com/tensorflow/quantum> (Abruf: 11.10.2023).
- Van Houdt, G./Mosquera, C./Nápoles, G. (2020)**: A review on the long short-term memory model. In: *Artificial Intelligence Review* 53.8, S. 5929–5955. ISSN: 1573-7462. DOI: 10.1007/s10462-020-09838-1. URL: <https://doi.org/10.1007/s10462-020-09838-1>.
- Vaswani, A./Shazeer, N./Parmar, N./Uszkoreit, J./Jones, L./Gomez, A. N./Kaiser, Ł./Polosukhin, I. (2017)**: Attention is All You Need. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., S. 6000–6010. ISBN: 9781510860964.
- Venugopalan, S./Rohrbach, M./Donahue, J./Mooney, R./Darrell, T./Saenko, K. (2015)**: Sequence to Sequence – Video to Text. In: *2015 IEEE International Conference on Computer Vision (ICCV)*, S. 4534–4542. DOI: 10.1109/ICCV.2015.515.
- Webster, J./Watson, R. T. (2002)**: Analyzing the past to prepare for the future: Writing a literature review. In: *MIS quarterly*, S. xiii–xxiii.
- Weigold, M./Barzen, J./Leymann, F./Salm, M. (2021)**: Expanding Data Encoding Patterns For Quantum Algorithms. In: *2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*. Stuttgart, Germany: IEEE, S. 95–101. DOI: 10.1109/ICSA-C52384.2021.00025.
- **(2022)**: Data Encoding Patterns for Quantum Computing. In: *Proceedings of the 27th Conference on Pattern Languages of Programs*. PLoP '20. Virtual Event: The Hillside Group, S. 1–11. ISBN: 9781941652169.
- Werbos, P. (1990)**: Backpropagation through time: what it does and how to do it. In: *Proceedings of the IEEE* 78.10, S. 1550–1560. DOI: 10.1109/5.58337.
- Williams, C. P./Clearwater, S. H. u. a. (1998)**: Explorations in quantum computing. Springer.
- Wiseman, H. M./Milburn, G. J. (2010)**: Quantum measurement and control. Cambridge University Press.
- Yan, H./Ouyang, H. (2018)**: Financial Time Series Prediction Based on Deep Learning. In: *Wireless Personal Communications* 102.2, S. 683–700. ISSN: 1572-834X. DOI: 10.1007/s11277-017-5086-2. URL: <https://doi.org/10.1007/s11277-017-5086-2>.
- Yu, Y./Hu, G./Liu, C./Xiong, J./Wu, Z. (2023)**: Prediction of Solar Irradiance One Hour Ahead Based on Quantum Long Short-Term Memory Network. In: *IEEE Transactions on Quantum Engineering* 4, S. 1–15. DOI: 10.1109/TQE.2023.3271362.

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit mit dem Thema: *tbd.* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

(Ort, Datum)

(Unterschrift)