Institut für Architektur von Anwendungssystemen

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit

Implementierung und Analyse von Gradientenberechnung in Quantenalgorithmen

Moritz Schmidt

Studiengang:	Informatik
Prüfer/in:	Prof. Dr. Dr. h.c. Frank Leymann
Betreuer/in:	Dr. phil. Johanna Barzen, Philipp Wundrack, M.Sc.
Beginn am:	13. Dezember 2021
Beendet am:	13. Juni 2022

Kurzfassung

Quantencomputer bieten die theoretische Möglichkeit verschiedenste Probleme präziser und schneller zu lösen als klassische Computer. Auch im Gebiet des maschinellen Lernens, welches in den letzten Jahren in einem immer größer werdenden Spektrum an Disziplinen Anwendung findet, hofft man das Potenzial des Quantencomputers zu entfalten. Viele Algorithmen des maschinellen Lernens sind im Kern Optimierungsprobleme. Um eine möglichst genaue Lösung für diese Probleme zu finden, werden oft gradientenbasierte Verfahren als Kompromiss zwischen Rechenaufwand und Qualität der Lösung verwendet. In dieser Arbeit werden verschiedene Methoden zur Bestimmung von Gradienten von Quantenschaltkreisen analysiert und verglichen. Die abschließenden Ergebnisse zeigen, wie die inhärente Varianz von Messungen auf Quantencomputern zu einem Dilemma bei der Wahl von Hyperparametern von numerischen Verfahren führt, warum das analytische Parameter-Shift Verfahren einzelne Gradienten nicht nur exakt, sondern auch effizient berechnet und warum das SPSA Verfahren vor allem zur Gradientenberechnung auf großen Schaltkreisen mit vielen Parametern eine gute numerische Alternative sein kann. Dies kann als Entscheidungsgrundlage zur Gradientenberechnung für zukünftige Implementierungen von Algorithmen des maschinellen Lernens auf Quantencomputern dienen.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Quantencomputer ermöglichen eine neue Art des Rechnens. Quantenmechanische Konzepte wie Superpositionen von Zuständen und Verschränkung haben das Potenzial Probleme exponentiell schneller zu lösen, als herkömmliche Computer [RT19]. Sie können beispielsweise dabei helfen mögliche neue Medizin, Düngemittel oder Baumaterialien zu entwickeln [De 17], beim Handel in der Finanzwelt zum Einsatz kommen [SES+20] und verändern, was in der modernen Krpytographie als sicher gilt [GRTZ02].

Maschinelles Lernen versucht anhand von Daten komplexe Zusammenhänge zu erlernen. Leistungsstärkere Computer, sowie stark gewachsene Datenmengen [ZPWV17] [RHW21] haben in den letzten Jahren für einen Durchbruch von praktischen Anwendungen von maschinellem Lernen geführt: Unter anderem konnte ein Computer den besten Gospieler der Welt schlagen [SHM+16], Vorhersagen von Proteinstrukturen können für zukünftige Medizin genutzt werden [JEP+21] und Klassifizierungen von Bildern [KSH12] können unter anderem zur Krankheitsdiagnose [FLNH13] oder für autonomes Fahren verwendet werden [FHY19].

Es gibt bereits einige Ideen die Stärken des Quantencomputers auch für das maschinelle Lernen zu nutzen. So wurde zum Beispiel die Support Vector Machine [RML14] und die Hauptkomponentenanalyse [CSAC20] auf den Quantencomputer übertragen. Eine weitere prominente Klasse von Algorithmen sind variationelle Quantenalgorithmen, welche in ihrem Konzept künstlichen neuronalen Netzen ähneln [CAB+21].

Viele Probleme des maschinellen Lernens sind problemspezifische Optimierungsprobleme. Auch wenn einige Optimierungsverfahren, wie der Nelder-Mead Algorithmus [NM65], ohne Gradienteninformationen optimieren können, liefern gradientenbasierte Verfahren oft effizientere und bessere Lösungen. Ein Beispiel für einen solchen Algorithmus für Probleme mit großen Datenmengen ist der stochastische Gradientenabstieg [Bot10]. Um nun auch mit gradientenbasierten Optimierungsverfahren Probleme des maschinellen Lernens auf Quantencomputern zu lösen, gilt es zu klären, ob sich möglichst genaue Quantengradienten mit akzeptablem Aufwand berechnen lassen.

In dieser Arbeit wurden verschiedene numerische und quantenspezifische analytische Verfahren zur Gradientenberechnung implementiert und analysiert. Als Anwendungsfall wurde dazu der Variational Quantum Eigensolver (VQE) [PMS+14] [KMT+17] verwendet. Die Verfahren wurden an verschiedenen Schaltkreisansätzen und Observablen getestet. Auch wenn die Anzahl an möglichen Konfigurationen und Hyperparametern eine allgemeingültige, empirische Auswertung schwer zulassen, wurde in der Analyse versucht einige mathematisch fundierte Aussagen über die Verfahren zu treffen.

Zuerst werden in Kapitel 2 die für diese Arbeit relevanten Grundlagen des Quantencomputings sowie in Kapitel 3 des maschinellen Lernens besprochen. Danach wird in Kapitel 4 die Synthese der beiden Gebiete motiviert und die Klasse der variationellen Quantenalgorithmen vorgestellt. Fokussiert wird dabei der Variational Quantum Eigensolver, welcher in Abschnitt 6.1 und Abschnitt 6.4 als

Ausgangslage genutzt wurde. In Kapitel 5 werden die verschiedenen untersuchten Methoden zur Gradientenberechnung dargelegt und versucht deren Charakteristika herauszuarbeiten. Auf die Implementierung der einzelnen Gradientenberechnungsverfahren wird in Abschnitt 6.1 eingegangen. Schließlich werden in Kapitel 6 die durchgeführten Experimente zum Vergleich der Verfahren besprochen. Dazu wird in Abschnitt 6.3 auf die empirischen Ergebnisse der Experimente eingegangen und auf dessen Basis Hypothesen über die Gradientenverfahren aufgestellt. Danach werden in Abschnitt 6.4 mögliche Erklärungen für diese Hypothesen beschrieben. Um diese Erklärungen empirisch zu fundieren, wurden weitere Experimente durchgeführt, welche in Abschnitt 6.5 besprochen werden. Zuletzt werden die Ergebnisse der Arbeit in Kapitel 7 zusammengefasst und ein Ausblick gegeben.

2 Quantencomputing

Hochleistungsrechner rechnen wie herkömmliche Computer. Sie haben jedoch meist deutlich mehr Speicher, sowie leistungsstärkere Prozessoren und können so Probleme schneller lösen. Quantencomputer unterscheiden sich von diesen Supercomputern und herkömmlichen Computern, indem sie, anhand der Gesetzte der Quantenmechanik, auf grundlegend andere Weise rechnen.

Anstelle von klassischen Bits verwenden sie sogenannte Qubits. Diese sind jedoch in ihrer Anzahl auf heutigen Quantencomputern noch begrenzt und fehleranfällig. In der Theorie kann jedoch ein Quantencomputer alle Berechnungen eines klassischen Computers ausführen. Manche Probleme lassen sich sogar nur auf einem Quantencomputer in polynomieller Zeit lösen [RT19].

Ein Beispiel für die erhofften neuen Rechenmöglichkeiten des Quantencomputers ist der Shor Algorithmus [Sho94], welcher in polynomieller Zeit Zahlen faktorisiert. Hierfür ist bislang kein klassischer Polynomialzeitalgorithmus bekannt. Dies kann weitreichenden Einfluss haben, da das Kryptographieprotokoll RSA auf der Prämisse basiert, dass das faktorisieren von Zahlen für einen potenziellen Angreifer nur schwer durchführbar ist. Auch wenn die Anzahl an heute verfügbarer Qubits für einen Angriff nicht ausreicht, wird schon in der Post-Quantum Krpytographie [GRTZ02] daran gearbeitet, neue Verfahren zu entwickeln, die auch vor Quantencomputern sicher sind.

In diesem Kapitel wird ein Überblick über die grundlegende Funktionsweise eines Quantencomputers gegeben und für den weiteren Teil der Arbeit relevante Konzepte eingeführt, sowie Limitierungen heutiger Quantenhardware besprochen. Für einen tieferen Einblick in die Grundlagen des Quantencomputings wird auf [NC04] verwiesen.

2.1 Qubits

Klassische Computer nutzen als kleinst mögliche Einheit das Bit. Dies beschreibt genau zwei mögliche Zustände: 0 oder 1. Die kleinste Einheit des Quantencomputers, sogenannte Qubits, erweitern diese Idee: Das Qubit kann auch Zustände zwischen 0 und 1 annehmen.

Formal ist ein Qubit $|\psi\rangle$ ein Vektor im komplexen Hilbertraum $\mathbb{H} \cong \mathbb{C}^2$.

In der Basis der Zustände $|0\rangle$ und $|1\rangle$ beschreibt ein Vektor $|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ den Zustand $\alpha\,|0\rangle + \beta\,|1\rangle$.

Dieser Vektor muss auf Länge 1 normiert sein: $\langle\psi|\,|\psi\rangle = |\alpha|^2 + |\beta|^2 = 1$. Im Bezug auf ein klassisches Bit lassen sich dann die Amplitudenquadrate $|\alpha|^2, |\beta|^2$ als Wahrscheinlichkeiten betrachten im Zustand 0 oder 1 zu sein.

Als visuelles Modell lässt sich ein Qubit im $\mathbb{C}^2$ als Einheitskugel betrachten, bekannt als die Bloch-Sphäre. Die Punkte auf der Oberfläche der Bloch-Sphäre entsprechen allen möglichen Zuständen, die ein Qubit annehmen kann.

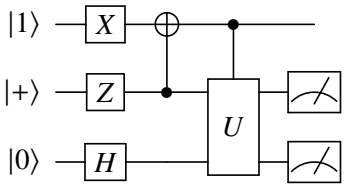

Abbildung 2.1: Darstellung von beispielhaften Operationen als Quantenschaltkreis

2.2 Operationen

Eine Abbildung f ist unitär genau dann wenn $\forall u, v \in \mathbb{H} : \langle f(v), f(w)\rangle = \langle v, w\rangle$.

Jede Operation auf einem Quantencomputer, die auf ein Qubit angewandt wird, ist unitär. Ein anschauliches Argument liefert hier die Bloch-Sphäre: Aus der Definition einer unitären Abbildung folgt, das jede Operation f auf einen Vektor v, dessen Länge nicht verändert. Wenn also eine Operation f auf einen Zustand $|\psi\rangle$ angewandt wird und f unitär ist, kann der resultierende Zustand $|\psi'\rangle = f(|\psi\rangle)$ nicht die Bloch-Sphäre verlassen, da die Abbildung längenerhaltend ist.

Unitäre Abbildungen sind linear und lassen sich somit als Matrizen formulieren. Wenn wir also den Zustandsvektor eines Qubits in einer beliebigen Basis betrachten, gibt es zu der unitären Abbildung f eine explizite unitäre Matrix U, die ebenfalls die Operation f beschreibt ($\forall |\psi\rangle : f(|\psi\rangle) = U|\psi\rangle$). Für unitäre Matrizen U gilt: $U^{-1} = U^\dagger \iff U^\dagger U = UU^\dagger = I$, wobei $U^\dagger$ die zu U adjungierte Matrix ist.

Zur Visualisierung wird ein Quantenalgorithmus oft als Schaltkreis dargestellt (siehe Abbildung 2.1). Dabei sind die Qubits horizontale Leitungen und Operationen, oft Gatter genannt, werden auf diesen als Blöcke eingezeichnet.

2.3 Messungen

Auf dem klassischen Computer kann der Zustand eines Bits deterministisch ausgelesen werden, indem man misst, ob gerade Strom fließt (Zustand 1) oder nicht (Zustand 0). Messen auf dem Quantencomputer muss nicht deterministisch sein.

Informell entscheidet man sich bei Messung eines Qubits für ein Zustandspaar, das sich auf der Bloch-Sphäre gegenüberliegt, bezüglich wessen man messen will. Den Zustand $\alpha|0\rangle + \beta|1\rangle$ aus Abschnitt 2.1 betrachtend, befindet das Qubit sich nach einer Messung in der Basis $\{|0\rangle, |1\rangle\}$ mit Wahrscheinlichkeit $|\alpha|^2$ im Zustand $|0\rangle$ und mit Wahrscheinlichkeit $|\beta|^2$ im Zustand $|1\rangle$. Nach der Messung ist der Zustand des Qubits bekannt, der ursprüngliche Zustand vor der Messung kann jedoch nicht durch diese einzelne Messung rekonstruiert werden.

Formal ist eine Messung die Projektion auf einen der Eigenräume[1] einer Observablen O. Die Matrix O ist hermitesch, es gilt also $O = O^\dagger$. Hermitesche Matrizen haben nur reelle Eigenwerte und die Eigenvektoren bilden eine Orthonormal-Basis. Jedem Zustand, der einem Eigenvektor von O entspricht (auch Eigenzustand genannt), kann so eine reelle Zahl zugeordnet werden. So lässt sich

[1] die jeweils von den Eigenvektoren $|\lambda_i\rangle$ jedes Eigenwerts λ aufgespannten Unterräume

die Messung eines Zustands $|\psi\rangle$ in O als Zufallsvariable interpretieren. Unter Verwendung der Eigendekomposition von $O = \sum_{\lambda\in\sigma(O)} \lambda \sum_i |\lambda_i\rangle \langle\lambda_i|$ ergibt sich dann der Erwartungswert $\mathbb{E}[O]$ der Messung, auch geschrieben als $\langle O\rangle_{|\psi\rangle}$ wie folgt:

$$\langle\psi| O |\psi\rangle = \sum_{\lambda\in\sigma(O)} \lambda \sum_i \langle\psi| |\lambda_i\rangle \langle\lambda_i| |\psi\rangle = \sum_{\lambda\in\sigma(O)} \lambda p(\lambda) = \mathbb{E}[O] \tag{2.1}$$

Messungen können als vielseitiges Werkzeug in Quantenalgorithmen eingesetzt werden. Da durch eine Messung der Zustand von Qubits verändert werden kann, ist es möglich nur anhand von Messungen beliebige Operationen ausführen [RBB03].

Die für diese Arbeit relevanten Algorithmen verwenden Messungen nur als Abschätzung des Erwartungswerts $\langle O\rangle_{|\psi\rangle}$. Mit einem Messergebnis ist daher immer der Erwartungswert oder eine Abschätzung dessen in Form eines Stichprobenmittelwerts gemeint.

2.4 Quantenregister und Quantenverschränkung

Ein Quantenregister ist, analog zu seinem klassischen Gegenstück, ein Zusammenschluss von mehreren Qubits. Formal ist ein Quantenregister aus N Qubits ein Vektor im Tensorprodukt $\mathbb{H} \otimes \mathbb{H} \cdots \otimes \mathbb{H} = \mathbb{H}^N$ aus N Hilberträumen $\mathbb{H}$, wobei auch hier weiterhin der Vektor auf die Länge 1 normiert sein muss. Die Dimensionalität wächst dabei exponentiell: $\mathbb{H}^N \cong (\mathbb{C}^2)^N = \mathbb{C}^{2^N}$. Da Tensorprodukte von unitären Matrizen wieder unitäre Matrizen und Tensorprodukte von hermiteschen Matrizen wieder hermitesche Matrizen sind, können Operationen und Observablen für ein einzelnes Qubit per Tensorprodukt für die Anwendung auf Quantenregistern erweitert werden. Es gibt jedoch auch Operationen die sich nicht als Tensorprodukt von Operationen auf einzelnen Qubits darstellen lassen.

Ein wichtiges Beispiel dafür ist das 2-Qubit CNOT Gatter: Ist das Kontroll-Qubit im Zustand $|1\rangle$ wird der Zustand des Ziel-Qubits negiert. Als Beispiel wendet man CNOT auf den Zustand $|+\rangle |0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$ an und erhält den Zustand $|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Der Zustand $|\phi^+\rangle$ hat eine Eigenschaft, die kein klassisches Gegenstück hat: Wenn man auf dem ersten Qubit den Zustand $|0\rangle$ misst, ist das zweite Qubit ebenfalls im Zustand $|0\rangle$. Im anderen Fall, wenn $|1\rangle$ gemessen wird, ist das zweite Qubit im Zustand $|1\rangle$. Die beiden Qubits sind nicht unabhängig voneinander. Die Messung des einen Qubits verändert den Zustand des anderen Qubits. Dieses Verhalten nennt man Quantenverschränkung.

2.5 Pauli-Matrizen

Die drei Pauli-Matrizen spielen eine vielseitige Rolle im Quantencomputing:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.2}$$

Sie sind unitär und haben wichtige Funktionen als Operationen. Zum Beispiel ist Pauli X die Negation eines einzelnen Qubits, daher lässt sich das CNOT Gatter auch als CX (kontrolliertes Pauli-X) verstehen.

Sie sind aber auch hermitesch und können somit als Observablen dienen. Will man bezüglich der Zustände $|0\rangle$ und $|1\rangle$ messen, kann man Pauli-Z als Observable verwenden, da die beiden Zustände Eigenvektoren von Z sind. Außerdem sind die Pauli-Matrizen zusammen mit der 2×2 Einheitsmatrix I eine Basis aller hermiteschen Matrizen in $\mathbb{C}^{2\times 2}$. Dies lässt sich für mehr Qubits erweitern: Die Menge aller sogenannten Pauli-Strings der Länge N, also die Menge $\{X, Y, Z, I\}^{\otimes N}$, ist ebenfalls eine Basis aller hermiteschen Matrizen in $\mathbb{H}^N = \mathbb{C}^{2^N}$.

Jede unitäre Matrix U lässt sich schreiben als Matrixexponential e^{iH} einer hermiteschen Matrix H, bezeichnet als der Generator von U. Die Pauli-Matrizen sind Generatoren von Rotationen auf der Bloch-Sphäre um jeweils die X, Y und Z Achse. Mithilfe dieser drei Rotationen lässt sich jedes 1-Qubit Gatter realisieren, zusätzlich mit dem CNOT Gatter sogar alle möglichen N-Qubit Gatter [BBC+95].

2.6 NISQ

Quantencomputing lässt sich auf einem klassischen Computer simulieren, indem man alle Operationen in dem exponentiell wachsenden Vektorraum $\mathbb{C}^{2^N}$ berechnet. Exponentielle Beschleunigung von Algorithmen lassen sich durch Anwendung auf Hardware erreichen, die nativ auf dem linear wachsenden Vektorraum $\mathbb{H}^N$ arbeiten. Beispiele für heutige Quantencomputer sind Ionenfallen, sowie superleitende oder photonenbasierte Quantencomputer.

Auch wenn das Feld an Hardware sehr divers ist, einen die verschiedenen Ansätze ähnliche Limitationen.

Ein großes Problem ist Skalierung. Sowohl die Anzahl an Qubits, als auch die mögliche Anzahl an nacheinander ausgeführten Gattern, bezeichnet als die Tiefe des Schaltkreises, ist stark begrenzt. Bei Letzterem kommt hinzu, dass oft nur eine geringe Anzahl an Basisgattern nativ unterstützt wird und alle anderen Gatter in einem Kompilierungsschritt erst in eine Darstellung in diesen Basis-Gattern zerlegt werden müssen, was die Tiefe des Schaltkreises weiter aufbläht. Außerdem werden die Gatter meist mit Fehlern ausgeführt, die sich über die Dauer des Ausführens verschlimmern. Es können Verfahren zur Verringerung von Fehlern angewandt werden, doch auch diese benötigen meist zusätzliche Qubits oder Gatter.

Heutige Quantencomputer können oft auf einzelnen Qubits nur bezüglich $|0\rangle$ und $|1\rangle$ messen, also Z als Observable nutzen. Oft will man jedoch in beliebigen Observablen messen. Hierzu kann man die in Abschnitt 2.5 beschriebene Zerlegung einer Hermiteschen Matrix in Pauli-Strings nutzen. Durch das Anfügen von Gattern können leicht die Eigenzustände von X und Y auf $|0\rangle$ und $|1\rangle$ abgebildet werden. So kann eine Messung in X oder Y künstlich über eine Messung in Z erzeugt werden. Wenn man die Zerlegung einer Observablen in Pauli-Strings kennt, kann man nun jeden Pauli-String messen und die Ergebnisse zum Gesamtergebnis zusammenfügen. Nur kommutierende Pauli-Strings können in einer Ausführung des Schaltkreises zusammen gemessen werden. Dementsprechend kann eine Messung in beliebiger Observable sich in mehrere Schaltkreisausführungen auffächern.

Die jetzige Zeit, in der Quantencomputing noch durch diese beschriebenen Hardwarelimitationen eingeschränkt ist, wird als Noisy Intermediate-Scale Quantum (NISQ) Ära bezeichnet [Pre18] [LB20]. Für aufwendige Algorithmen wie den Shor-Algorithmus reichen NISQ-Quantencomputer noch nicht aus, viele neue Algorithmen berücksichtigen jedoch deren Grenzen und sind schon auf heutigen Quantencomputern anwendbar.

3 Maschinelles Lernen

Im Zentrum von maschinellem Lernen stehen Daten. Es wird versucht, Erkenntnisse über Daten zu gewinnen und mithilfe dieser Erkenntnisse Rückschlüsse auf die Methode zu ziehen, mit der die Daten generiert wurden. Aus einer Menge von Bildern von Verkehrsschildern können so eigenständig Regeln abgeleitet werden, um diese zu unterscheiden und danach auch auf neuen, vorher ungesehenen Bildern wiedererkannt werden [SSSI11]. So kann zum Beispiel auch in medizinischen Bilddaten gelernt werden, ob ein Patient an Krebs erkrankt ist [FLNH13] oder in Spielsituationen in Go der bestmögliche Zug vorausgesagt werden [SHM+16].

Maschinelles Lernen ist eine große und vielseitige Disziplin. In diesem Kapitel wird lediglich eine Einführung in die Konzepte des Supervised und Unsupervised Learnings gegeben, sowie der Bezug zu Optimierungsproblemen hergestellt. Für eine weitreichendere Einführung in das Gebiet des maschinellen Lernens wird auf [Dom12] verwiesen. Außerdem wird für eine ausführliche Vorstellung von Algorithmen auf [Bis06], sowie für eine ausführliche Erklärung der unterliegenden mathematischen Konzepte auf [DFO20] verwiesen.

3.1 Supervised Learning

Gegeben sei eine Menge von D Paaren $\{(x_i, y_i)\}_{i=1}^{D}$, jedes $x_i \in \mathbb{R}^d$ ist ein d dimensionaler Datenpunkt und y_i der zu x_i gehörende Zielwert. Im Supervised Learning befasst man sich mit dem Problem, mithilfe der gegebenen Datenpunkte, die Funktion $f : x \mapsto y$ zu lernen. Falls der Zielwert kontinuierlich (z.B. $y_i \in \mathbb{R}$) ist, spricht man von einem Regressionsproblem. Wird stattdessen jedem Datenpunkt ein Element aus einer Menge von Klassennamen zugeordnet, spricht man von einem Klassifizierungsproblem. Mithilfe eines Modells, das zusätzlich zum Eingabedatum x noch von Modellparametern θ abhängt, wird versucht die Funktion f bestmöglich zu approximieren. Im besten Fall schafft man es optimale Parameterwerte θ^* zu finden, sodass möglichst viele Datenpunkte vom Modell korrekt auf y_i abgebildet werden. Zusätzlich hofft man, dass das resultierende Modell möglichst allgemeingültig ist, also auch für neue, unbekannte Eingabedaten richtige Zielwerte voraussagt.

3.2 Unsupervised Learning

Im Unsupervised Learning hat man lediglich eine Datenmenge $\{x_i\}_{i=1}^{N}$ gegeben. Da es hier keine Zielwerte zur Orientierung gibt, definiert man stattdessen eigene Ziele, die beschreiben, welche Erkenntnisse man aus den Daten gewinnen möchte. Ein bekanntes Beispiel dafür sind Clusteringalgorithmen. Hierbei wird versucht Daten in Gruppen (Cluster) aus zusammenhängenden

Datenpunkten einzuteilen. Ebenfalls prominent sind Komprimierungsverfahren. Hier wird untersucht, wie sich Datenpunkte mit möglichst wenig Informationsverlust in einem Vektorraum niedrigerer Dimensionalität darstellen lassen.

3.3 Verlustfunktion

Um eine optimale Funktionsapproximation zu finden, benötigt man ein Gütemaß. Im Fall von Supervised Learning bietet sich meist ein Abstandsmaß mithilfe einer Metrik zwischen dem Zielwert y_i und der Approximation des Modells an. Beim Unsupervised Learning muss man kreativ für das selbst gesetzte Ziel ein Maß modellieren. Üblicherweise wird die Abbildung $L(\theta)$, die für Parameter θ des Modells das zugehörige Gütemaß angibt, als Kosten- oder Verlustfunktion aufgefasst. Gut gewählte Parameter θ minimieren die Kosten $L(\theta)$. Eine optimale Lösung ist dabei aber immer nur so gut wie die Problemmodellierung der Kostenfunktion.

3.4 Optimierung

Wenn man eine Kostenfunktion gegeben hat, bleibt die Frage wie man die optimalen Parameter θ^* findet, oder formal:

$$\arg\min_{\theta} L(\theta) = \theta^* \tag{3.1}$$

Für viele Kostenfunktionen lässt sich das Minimum nicht analytisch bestimmen. Stattdessen kann man iterative Verfahren einsetzen, bei denen man an einem zufälligen Punkt im Parameterraum startet und sich dem Minimum Schritt für Schritt annähert. Bei diesen Verfahren weiß man jedoch nicht, ob man das globale Minimum erreicht hat, sondern kann nur die Kosten an einem Punkt mit den Kosten an bereits vorher betrachteten Parameterpunkten vergleichen. Daher kann es sein, dass man in lokalen Minima oder Plateaus mit dauerhaft geringfügig unterschiedlichen Kosten stecken bleibt. Wenn die Kostenfunktion jedoch konvex oder sogar streng konvex ist, treten diese Fälle nicht auf. Es kann jedoch trotzdem viele Iterationen dauern das Ziel zu erreichen.

3.4.1 Blackbox Optimierung

Blackbox Optimierungsverfahren betrachten die Verlustfunktion als Blackbox, man kann an beliebigen Stellen den Funktionswert abfragen, aber hat keinerlei weitere Informationen (z.B. Stetigkeit oder Differenzierbarkeit) über die Funktion. Dies ist zum einen ein Vorteil, da man diese Algorithmen auf beliebige Funktionen anwenden kann. Man muss jedoch meist an deutlich mehr Punkten den Funktionswert abfragen als bei Algorithmen, die weitere Informationen über die Funktion ausnutzen können. Ein Beispiel für diese Klasse von Optimierungsalgorithmen ist das Nelder-Mead Verfahren [NM65].

3.4.2 Gradienten Basierte Optimierung

Der Gradient $\nabla f(x)$ einer Funktion an einem Punkt x ist proportional zum steilsten Anstieg an diesem Punkt[1]. Wenn man in die entgegengesetzte Richtung geht, entspricht dies der lokal bestmöglichen Minimierung des Funktionswertes. Der klassische Gradientenabstiegsalgorithmus geht in jeder Iteration genau so vor, wobei vom Nutzer als Hyperparameter die Schrittweite festgelegt werden muss (die Länge des Gradienten ist dafür nicht geeignet). Alternativ kann man sogenannte Optimizer wie ADAM [KB14] nutzen, die die Schrittweite basierend auf den Messwerten und Gradienten vorheriger Iterationen anpassen. In der Praxis wird im maschinellen Lernen oft ein stochastischer Gradientenabstieg [SWM+20] verwendet, bei dem in jeder Iteration der Gradient nur über eine zufällig gewählte Teilmenge der Daten berechnet wird. Dadurch kann bei großen Datenmengen der Rechenaufwand reduziert werden.

3.4.3 Newton-Verfahren

Das Newton-Verfahren ist ursprünglich ein Verfahren um Nullstellen einer Funktion zu finden. Da Extrema Nullstellen der ersten Ableitung sind, kann man das Verfahren aber auch zur Optimierung nutzen. Analog zum Gradienten als Gegenrichtung des steilsten Abstiegs, lässt sich der Newton-Schritt $-\nabla^2 f(x)^{-1} \nabla f(x)$ als der Schritt zum Minimum der Taylor-Approximation zweiter Ordnung der Kostenfunktion am Punkt x verstehen. Der große Vorteil gegenüber dem Gradientenabstieg besteht darin, dass ein Schritt und keine Richtung berechnet wird. Es ist also kein Hyperparameter für die Schrittweite mehr nötig. Während für den Gradienten $\nabla f(x)$ insgesamt d Werte bestimmt werden müssen (eine partielle Ableitung pro Dimension), müssen für die Hesse-Matrix $\nabla^2 f(x)$ $O(d^2)$ Werte bestimmt werden. Die Hesse-Matrix muss zusätzlich noch invertiert werden, was gerade in hochdimensionalen Parameterräumen sehr aufwendig sein kann.

Oft benötigen die komplexeren Optimierungsverfahren weniger Iterationen, um ein Minimum zu erreichen, dafür kann jede Iteration aber deutlich aufwendiger sein. Je nachdem wie aufwendig (oder überhaupt möglich) das Messen von Funktion, Ableitung und zweiter Ableitung ist, muss man abwägen, welches Verfahren am besten geeignet ist. Es lassen sich Ableitungen auch numerisch annähern, was Blackbox Optimierungsalgorithmen oft implizit machen, und somit komplexere Optimierungsverfahren anwenden, selbst wenn das explizite Messen von Ableitungen nicht möglich ist.

[1] Hier wird von einer euklidischer Norm ausgegangen. Ansonsten muss der Gradient noch mit dem invertierten metrischen Tensor skaliert werden.

4 Maschinelles Lernen auf Quantencomputern

Im Feld des maschinellen Lernens wird bereits an verschiedensten Stellen versucht, die Stärken des Quantencomputers auszunutzen. Zum einen wird in Hybriden Ansätzen wie Quantenkerneln [HCT+19] oder hybriden Quantum-Klassischen neuronalen Netzen [MBI+20] [ZG21] versucht klassische Algorithmen mit Quantenkomponenten zu verbessern. Ebenfalls wurden Konzepte entwickelt, die es erlauben künstliche Neuronen direkt auf dem Quantencomputer zu implementieren [CGA17] [TMGB19]. Quanteneigene Konzepte wie Superpositionen und Verschränkung lassen vermuten, dass sich auch im maschinellen Lernen ein Quantenvorteil erzielen lässt. So musste beispielsweise das No-Free-Lunch Theorem, welches die Limitationen des maschinellen Lernens aufzeigt, für Quantencomputer neu formuliert werden [SCH+22].

4.1 Variationelle Quantenalgorithmen

Variationelle Quantenalgorithmen (VQAs) sind eine Klasse von Quantenalgorithmen, die Schaltkreise aus parametrisierten Gattern verwenden, sogenannte *Ansätze* oder Variational Forms, deren Parameter mit klassischen Methoden optimiert werden. Als Verlustfunktion fungiert dabei die Messung in einer problemspezifischen Observable, beziehungsweise eine Funktion über mehrere solcher Messungen.

Ein als gut angesehener Ansatz kann mit wenig Gattern eine große Menge unitärer Operationen generieren [DHLT20]. Man muss sich also zu jedem Problem nicht spezifisch einen konkreten Schaltkreis überlegen, sondern hofft, dass die gewünschte unitäre Operation durch den Ansatz erzeugt und die dazugehörige Parameterkonfiguration durch Optimierung gefunden werden kann. Auf das maschinelle Lernen bezogen lassen sich Parallelen zum klassischen Feature bzw. Representation Learning erkennen: Beispielsweise soll ein neuronales Netzwerk bei einem Klassifizierungsproblem eine Funktion darstellen, die Eingabedaten Klassen zuordnet, wobei man hofft, dass die gewünschte Funktion durch die Netzwerkarchitektur möglichst akkurat dargestellt werden kann.

Dementsprechend liegt es nahe VQAs auch im maschinellen Lernen einzusetzen. Wichtige Fragen sind dabei, wie man Eingabedaten als Quantenzustand kodiert, wie man eine Verlustfunktion als Observable konstruiert und welchen Ansatz man wählt.

Ein bekannter VQA ist der Quantum Approximate Optimization Algorithmus (QAOA), der beispielsweise angewandt auf das Max-Cut Problem in Clusteringalgorithmen Verwendung findet [FGG14].

Ein anderer bekannter Algorithmus ist der Variational Quantum Eigensolver (VQE) [PMS+14], der in modifizierter Form [CSAC20][LTO+19] für die Hauptkomponentenanalyse (PCA) verwendet werden kann. Der VQE Algorithmus wurde als Anwendungsfall für alle Experimente in dieser Arbeit verwendet und wird im Abschnitt 4.2 näher erläutert.

VQAs sind in der heutigen NISQ Ära besonders prominent, da in einer Iteration ein einzelner Schaltkreis meist kurz ist, was Dekohärenz vorbeugt und einen wachsenden Fehler von ungenauen Gattern verringert. Außerdem bestehen die Ansätze meist aus Gattern, die von vielen Quantencomputern als Basisgatter verwendet werden.

Zudem haben viele Quantenalgorithmen das Problem, dass sie viele zusätzliche Qubits benötigen. Die Ansätze von VQAs sind dagegen meist sehr flexibel und lassen sich auf beliebige Qubitzahlen anpassen. Die Größe der problemspezifischen Observable, sowie die mögliche Kodierung von Eingabedaten als Quantenzustand, bestimmen üblicherweise bei VQAs die Anzahl an benötigten Qubits.

Da VQAs iterative Optimierungsalgorithmen sind, können je nach Verlustfunktion, Ansatz, Observable und Optimierungsalgorithmus viele Ausführungen nötig sein, bis ein Minimum gefunden wurde. Falls die Verlustfunktion mehrere Minima besitzt, kann der Algorithmus auch in einem lokalen Minimum stecken bleiben. Außerdem wächst mit zunehmender Anzahl Qubits die Chance auf sogenannte Barren Plateaus [MBS+18], also große Abschnitte mit fast gleichem Funktionswert, in welchen Optimierungsalgorithmen leicht stecken bleiben können.

Auch wenn VQAs meist simple Schaltkreise haben, sind dafür die problemspezifischen Observablen oft kompliziert. Dies kann zu zusätzlichen Ausführungen des Schaltkreises pro Iterationen führen, wie in Kapitel 2.6 bereits besprochenen.

4.2 Variational Quantum Eigensolver

Der Variational Quantum Eigensolver ist ein VQA aus dem Bereich der Quantenchemie [KMT+17] [PMS+14],[OBK+16]. In der Quantenphysik und Quantenchemie beschreibt ein Quantenregister meist den Zustand eines physikalischen Systems [SF14]. Während in der Quanteninformatik ein Quantenregister $|\psi\rangle$ als statisch betrachtet wird, solange nicht Gatter darauf angewandt werden, wird ein Zustand $|\psi(t)\rangle$ in der Quantenphysik als dynamisch betrachtet. Der Hamiltonoperator $\hat{H}$ beschreibt in der zeitabhängigen Schrödingergleichung diese Veränderung des Zustands des Systems mit der Zeit:

$$i\hbar \frac{\partial |\psi(t)\rangle}{\partial t} = \hat{H} |\psi(t)\rangle \tag{4.1}$$

Zusätzlich geben die Eigenwerte der Eigenzustände $|E_j\rangle$ des Hamiltonoperators die Energie E_j dieser Zustände an, beschrieben in der zeitunabhängigen Schrödingergleichung:

$$\hat{H} |E_j\rangle = E_j |E_j\rangle \tag{4.2}$$

So fungiert der Hamiltonoperator als Observable, um die Energien von physikalischen Zuständen zu messen.

In der Quantenchemie gibt es nun die Problemstellung, dass der Hamiltonoperator eines Moleküls bekannt ist, jedoch nicht seine Eigenzustände und Eigenenergien. Im Grundzustandsproblem will man die geringste Energie, die das System haben kann (also wenn das System sich im Grundzustand befindet), bestimmen.

Mathematisch formuliert bedeutet das, dass man den kleinsten Eigenwert der hermiteschen Matrix $\hat{H}$ bestimmen will. Der Variational Eigenvalue Solver ist ein VQA, der dieses Problem lösen soll. Wie bei einem VQA üblich, ist der Schaltkreis ein parametrisierter Ansatz $U(\theta)$. Man startet im Zustand $|0\rangle$, wendet den Ansatz mit zufällig gewählten initialen Parameterwerten θ an und misst mit dem Hamiltonoperator $\hat{H}$ als Observable. Das Ergebnis der Messung entspricht der Energie des Zustands. Nun adaptiert man die Parameter θ klassisch mit einem Optimierungsverfahren und minimiert so die gemessene Energie. Im besten Fall bildet der Ansatz $U(\theta)$ nach abgeschlossener Optimierung für die optimalen Parameter θ^* den Startzustand $|0\rangle$ auf den Grundzustand mit kleinst möglicher Energie ab.

Es gibt auch andere Quantenalgorithmen zur Bestimmung von Eigenwerten, zum Beispiel mithilfe von Quantum Phase Estimation [OTT19], die sich ebenfalls für das Grundzustandsproblem, sowie auf Probleme des maschinellen Lernens verwenden lassen [LMR14][HLL+22]. Diese benötigen jedoch meistens große Schaltkreise, kompliziertere Gatter und viele zusätzliche Qubits. Dagegen ist der VQE Algorithmus, der die besprochenen Vorteile von VQAs ausnutzen kann, für heutige NISQ-Quantencomputer geeigneter.

5 Quantum Gradienten

Um auch gradientenbasiert in VQAs zu optimieren, muss man die Gradienten von parametrisierten Quantenschaltkreisen bestimmen können. Zuerst werden numerische Gradientenapproximationsverfahren vorgestellt, die unabhängig von der Problemstellung auf beliebigen Funktionen angewandt werden können. Danach wird eine Auswahl von analytischen Gradientenberechnungsverfahren besprochen, die spezifisch für die Berechnung von Gradienten von parametrisierten Quantenschaltkreisen konzipiert sind. Dies bedeutet nicht, dass diese Verfahren spezifisch für den VQE Algorithmus entworfen wurden. Alle Quantenalgorithmen, die der Problemdefinition in 5.1 entsprechen, können diese Verfahren verwenden.

Bevor die Gradientenberechnungsverfahren besprochen werden, gibt es noch einen kurzer Einschub: Eine Technik, die zur Beschleunigung von Gradientenberechnungen in neuronalen Netzen verwendet wird, ist Backpropagation. Backpropagation ist ein Verfahren der automatischen Differenzierung [Mac16][BPRS17]. Dabei werden relevante Zwischenergebnisse bei der Auswertung der Funktion zwischengespeichert und für die Berechnung des Gradienten wiederverwendet. Gerade bei rekursiven oder in Schichten aufteilbaren Funktionen können so oft doppelte Berechnungen vermieden werden. Auch Quantenschaltkreise lassen sich als Schichten von Gattern interpretieren, die nacheinander auf den Startzustand ausgeführt werden. In klassischen Algorithmen ist es kein Problem den Wert eines gerade verwendeten Registers zwischenzuspeichern. Auf dem Quantencomputern dagegen entsprechen die Zwischenergebnisse den unbekannten Zuständen nach jeder Schicht von Gattern[1]. Zum Zwischenspeichern müssten diese Zustände mithilfe aufwändiger Quantentomographieverfahren [DPS03] aus mehreren Messungen rekonstruiert werden. Nach jeder Messung muss dabei der Schaltkreis neu gestartet werden, da, wie in Abschnitt 2.3 besprochen, der Zustand durch Messungen potenziell verändert wird. Daher ist automatische Differenzierung innerhalb eines Quantenschaltkreises in der Praxis nicht sinnvoll. Ausnahmen bilden Statevector Simulatoren und hybride Anwendungen, bei denen ganze, gekapselte Quanten- und klassische Berechnungen als Schichten interpretiert werden können [BIS+18].

5.1 Problemdefinition

Gegeben sei ein Schaltkreis $U(\theta)$ auf beliebig vielen Qubits, der auf den Startzustand $|0\rangle^{\otimes N}$ (alle Qubits sind im Zustand 0) angewandt wird[2]. Die Gatter des Schaltkreises können dabei von den k Parametern θ_i des Parametervektors $\theta \in \mathbb{R}^k$ abhängen. Der resultierende Zustand nach Anwendung

[1] Hier wird davon ausgegangen, dass, wie bei VQAs üblich, nur einmal am Ende des Schaltkreises ein Erwartungswert gemessen wird.

[2] Falls man stattdessen in einem beliebigen Zustand $|\psi_0\rangle$ starten möchte, kann ein Operator U_{ψ_0}, für den $U_{\psi_0}|0\rangle = |\psi_0\rangle$ gilt, in $U(\theta)$ absorbiert werden.

von $U(\theta)$ ist der Zustand $|\psi(\theta)\rangle$. Danach misst man in Observable $O(\omega)$, die von m Parametern ω_i des Parametervektors $\omega \in \mathbb{R}^m$ abhängt. Die Verlustfunktion $L(\theta, \omega)$ entspricht dem Erwartungswert $\langle\psi(\theta)| O(\omega) |\psi(\theta)\rangle$ dieser Messung.

Man kann den Gradienten bezüglich Observablenparametern ω oder Schaltkreisparametern θ berechnen. Gradientenberechnungen bezüglich ω gestalten sich simpler und werden kurz in 5.2 erläutert. Der Fokus der Arbeit liegt auf Gradientenberechnung bezüglich Schaltkreisparametern θ, dementsprechend wird im Weiteren die Verlustfunktion nur noch als $L(\theta)$ bezeichnet. Alle besprochenen Verfahren und Experimente beziehen sich auf dieses Szenario.

Dass der Erwartungswert der Messung als Verlustfunktion verwendet wird, entspricht unter anderem dem VQE Szenario. Die Verlustfunktion kann aber im Prinzip auch eine Funktion eines oder mehrerer Erwartungswerte sein. Ein Beispiel dafür ist eine übliche Verlustfunktion des maschinellen Lernens, bei der für jedes Eingabedatum der Erwartungswert des Schaltkreises ausgewertet wird und der mittlere quadratische Fehler zu gewünschten Zieldatenpunkten als Verlustfunktion $L(\theta)$ fungiert. Die konkrete Ableitung kann dann über die Kettenregel bestimmt werden, wobei zusätzliche klassische Berechnungen hinzukommen. Da unsere Messergebnisse nur Schätzer der Erwartungswerte sind, kann es hier zu Schwierigkeiten kommen, falls nicht-lineare Funktionen verwendet werden [SWM+20].

Die hier vorgestellten Verfahren berechnen bzw. approximieren partielle Ableitungen bezüglich einzelner Parameter. Diese partiellen Ableitungen müssen dann zum Gradienten zusammengesetzt werden. Im weiteren Verlauf werden Approximationen der partiellen Ableitung $\frac{\partial L(\theta)}{\partial \theta_i}$ des i-ten Parameters θ_i als g_i bezeichnet. In den genannten Verfahren kommt es häufiger vor, dass bei der Berechnung von g_i am Parameterpunkt θ die Verlustfunktion an der Stelle $\theta + e_i h$ gemessen wird, wobei e_i hier den i-ten Einheitsvektor bezeichnet. Es wird also an einer Stelle gemessen, die nur in der Dimension des betrachteten Parameters um einen Skalar h vom Ausgangspunkt verschoben wurde. Zur einfacheren Notation wird eine solche Messung anstelle von $L(\theta + e_i h)$ mit $L(\theta + h)$ abgekürzt.

5.2 Gradienten bzgl. Observablenparametern

Um zu verstehen, wie Gradientenberechnung bezüglich Observablenparametern funktioniert, muss zuerst geklärt werden, wie eine Observable parametrisiert werden kann. Wichtig ist, dass $O(\omega)$ weiterhin eine gültige Observable ist, also hermitesch bleibt. Gültige Parametrisierungen lassen sich wie folgt schreiben:

$$O(\omega) = \sum_{O_i \in B} f_i(\omega) O_i \tag{5.1}$$

Die Menge B ist eine beliebige Basis der hermiteschen Matrizen mit Basiselementen O_i, beispielsweise die Menge aller Pauli-Strings (in $\mathbb{C}^{2^N \times 2^N}$), wie in 2.5 besprochen. Die Funktionen f_i sind beliebige Funktionen $\mathbb{R}^m \to \mathbb{R}$, die den Parametervektor ω auf die jeweiligen Koeffizienten der Basiselemente abbilden.

Wenn sich $O(\omega)$ so darstellen lässt, kann man die Sesquilinearität des Skalarprodukts bei der Gradientenberechnung bezüglich eines ω_j ausnutzen:

$$\begin{aligned}\frac{\partial}{\partial\omega_j}L(\theta,\omega) &= \frac{\partial}{\partial\omega_j}\langle\psi(\theta)|\sum_{O_i\in B} f_i(\omega)O_i\,|\psi(\theta)\rangle \\ &= \frac{\partial}{\partial\omega_j}\sum_{O_i\in B} f_i(\omega)\,\langle\psi(\theta)|\,O_i\,|\psi(\theta)\rangle \\ &= \sum_{O_i\in B}\frac{\partial f_i(\omega)}{\partial\omega_j}\,\langle\psi(\theta)|\,O_i\,|\psi(\theta)\rangle\end{aligned}$$

Die Messungen des Quantenschaltkreises $\langle\psi(\theta)|\,O_i\,|\psi(\theta)\rangle$ sind komplett unabhängig von der Ableitung. Der zusätzliche Aufwand besteht in der Berechnung von $\frac{\partial f_i(\omega)}{\partial\omega_j}$, den Ableitungen bezüglich der Koeffizientenfunktionen, einem rein klassischen Berechnungsschritt. Dieses Reduzieren der Gradientenberechnungen auf klassische Berechnungen ist bei Ableitungen bezüglich Schaltkreisparametern nicht möglich. Dort gestaltet sich die Gradientenberechnung schwieriger. Es werden nun einige Verfahren vorgestellt, die den Gradienten bezüglich Schaltkreisparametern berechnen bzw. approximieren.

5.3 Numerische Verfahren

Die Definition der Ableitung an einem Punkt x_0 ist gegeben durch den Differenzenquotienten:

$$\frac{\partial f(x)}{\partial x} = \lim_{h\to 0}\frac{f(x_0+h)-f(x_0)}{h} \tag{5.2}$$

Viele numerische Verfahren nutzen dies aus, um Ableitungen zu approximieren. Dafür rechnet man den Differenzenquotienten für einen kleinen Abstand ϵ explizit aus. Auch wenn das Ergebnis nur eine Approximation der Ableitung ist, haben numerische Verfahren den Vorteil, dass sie keine explizite Formulierung der abzuleitenden Funktion benötigen, sondern, ähnlich zu Blackbox Optimierungsverfahren, auf jede Funktion angewandt werden können. Man kann also einen Quantenschaltkreis als Blackbox betrachten und mit den hier vorgestellten numerischen Verfahren Gradienten annähern.

5.3.1 Finite Differenzen

Finite Differenzen Verfahren finden vor allem Anwendung in numerischen Methoden zur Lösung von Differenzialgleichungen [SK13][Smi85]. Für eine ausführliche Erläuterung der hier vorgestellten Verfahren wird auf [MM05] und [Col12] verwiesen.

Das wichtigste Werkzeug zur Approximation von Gradienten mittels finiter Differenzen ist die Taylor-Entwicklung:

$$f(x+h) \approx f(x) + \frac{h}{1}f'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \cdots = \sum_{i=0}^{\infty} \frac{h^i}{i!}\frac{\partial^i f(x)}{\partial x^i} \tag{5.3}$$

Wenn man die Reihe bezüglich der ersten Ableitung umstellt, erhält man die vorwärts finiten Differenzen:

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h^2) \tag{5.4}$$

Man beachte die Ähnlichkeit zur Ableitungsdefinition 5.2.

Analog dazu erhält man mit einem negativen Abstand, also am Punkt $f'(x-h)$, die rückwärts finiten Differenzen:

$$f(x-h) = f(x) - hf'(x) + O(h^2) \implies f'(x) = \frac{f(x) - f(x-h)}{h} + O(h^2) \tag{5.5}$$

Man kann auch die Taylor-Reihen beider Stellen $x+h$ und $x-h$ kombinieren, was in den sogenannten zentralen finiten Differenzen resultiert:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2) \tag{5.6}$$

Mithilfe dieser Verfahren kann jede partielle Ableitung $\frac{\partial f}{\partial x_i}$ innerhalb des Gradienten $\nabla f(x)$ approximiert werden. Vorwärts und rückwärts finite Differenzen haben im Gegensatz zu zentralen finiten Differenzen den Vorteil, dass der Funktionswert $f(x)$ für jede partielle Ableitung wiederverwendet werden kann. Während man bei zentralen finiten Differenzen für jede der k partiellen Ableitungen zwei neue Funktionswerte, also insgesamt $2k$ Funktionswerte, messen muss, benötigen vorwärts und rückwärts finite Differenzen nur $k+1$ Funktionsaufrufe.

Beliebige Ableitungen mit beliebigen Punkten kann man mithilfe eines Koeffizientenvergleichs approximieren. Seien F_i die Funktionswerte an N verschiedenen Punkten mit Abstand h_i von der Samplestelle x. Hier soll die erste Ableitung approximiert werden. Es müssen also Koeffizienten α_i gefunden werden, für die $\sum_{i=1}^{N} \alpha_i F_i \approx \frac{\partial f(x)}{\partial x}$ gilt.

Wenn man jedes F_i nun als Taylor-Erweiterung bis $\frac{\partial^N f(x)}{\partial x^N}$ ausschreibt, lässt sich ein lineares Gleichungssystem bezüglich der Koeffizienten der Ableitungsterme aufstellen:

$$\begin{pmatrix} 1 & 1 & 1 & \cdots \\ h_1 & h_2 & h_3 & \cdots \\ \frac{h_1^2}{2} & \frac{h_2^2}{2} & \frac{h_3^2}{2} & \cdots \\ \vdots & \vdots & \vdots & \\ \frac{h_1^N}{(N)!} & \frac{h_2^N}{(N)!} & \frac{h_3^N}{(N)!} & \cdots \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \vdots \\ \alpha_N \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{5.7}$$

Jede Zeile entspricht einem Ableitungsterm, jede Spalte einem Abtastpunkt. Durch Lösung des Gleichungssystems erhält man die gewünschten Koeffizienten.

Da bei konkreter Anwendung von finite Differenzen Verfahren meist kleine Abstände $0 < \epsilon \ll 1$ genutzt werden, bedeuten Fehlerterme höherer Ordnung $O(h_i^N)$ kleinere Fehler. Bei niedrigeren Ableitungen können mehr Abtastpunkte zu einem Fehlerterm höherer Ordnung und somit zu einem geringeren Fehler führen. Der genaue Fehler lässt sich bestimmen, indem man in $\sum_{i=1}^{N} \alpha_i F_i$ die berechneten Koeffizienten und einzelnen Taylor-Erweiterungen wieder einsetzt.

Für die konkrete Problemstellung dieser Arbeit beschreiben die folgenden Gleichungen die Approximationen g_i der partiellen Ableitungen $\frac{\partial L(\theta)}{\partial \theta_i}$ mittels finiter Differenzen:

$$g_{i(VFD)} = \frac{L(\theta + h) - L(\theta)}{h} \tag{5.8}$$

$$g_{i(RFD)} = \frac{L(\theta) - L(\theta - h)}{h} \tag{5.9}$$

$$g_{i(ZFD)} = \frac{L(\theta + h) + L(\theta - h)}{2h} \tag{5.10}$$

5.3.2 Simultaneous Perturbation Stochastic Approximation

Ein Nachteil der finiten Differenzen Verfahren ist, dass es bei der Bestimmung eines vollständigen Gradienten $\nabla L(\theta_1, \ldots, \theta_k)$ auf jeden Parameter einzeln angewandt werden muss. Für die Ableitungen $\frac{\partial L}{\partial \theta_i}$ bezüglich k Parametern mit dem zentralen finite Differenzen Verfahren benötigt man beispielsweise die Funktionswerte an $2k$ Samplestellen.

Das Simultaneous Perturbation Stochastic Approximation (SPSA) Verfahren [Spa87] benötigt hingegen nur 2 Funktionsaufrufe für beliebig viele Parameter. Dazu wählt man einen zufällig gewählten Vektor Δ, der, mit einem festen Abstand h skaliert, als Abstandsvektor fungiert. Die einzelnen Gradiententerme werden dann ähnlich zu zentralen finiten Differenzen berechnet, wobei die zwei Funktionswerte für alle Ableitungen unverändert bleiben und nur die Skalierungen anhand der einzelnen Elemente $h\Delta_i$ des Abstandsvektors variiert werden:

$$g_{i(SPSA)} = \frac{L(\theta + h\Delta) - L(\theta - h\Delta)}{2h\Delta_i} \tag{5.11}$$

Dies ist meist eine gröbere Abschätzung des Gradienten, jedoch handelt es sich bei g_{SPSA}, sofern die Wahrscheinlichkeitsverteilung für Δ passend gewählt wird, um einen erwartungstreuen Schätzer des Gradienten. In dieser Arbeit wurden die Einträge Δ_i unabhängig voneinander zufällig aus $\{\pm 1\}$ anhand einer Bernoulliverteilung mit Wahrscheinlichkeit $\frac{1}{2}$ für beide Fälle gezogen. SPSA findet vor allem in iterativen Optimierungsalgorithmen Verwendung. Dort können die ungenaueren Gradienten oft über mehrere Iterationen ausgeglichen werden, wodurch der Algorithmus mit geringerem Gesamtaufwand ein Optimum erreicht.

Da bei Quantenalgorithmen Funktionsaufrufe aufwändig sind, wäre eine konstante Skalierung bezüglich der Parameteranzahl von Vorteil. Ein weiteres Problem ist, dass das Messen von Funktionswerten, gerade auf NISQ Hardware, sehr verrauscht sein kann. Dazu wird in [Spa98] vorgeschlagen in jeder Iteration den Gradienten präziser zu approximieren, indem man den Durchschnitt mehrerer solcher Gradientenapproximationen berechnet. Dies sei gerade bei verrauschten Funktionen vorteilhaft.

5.4 Analytische Verfahren

Analytische Gradientenberechnungsverfahren berechnen für eine Funktion die exakte Ableitung mithilfe von analytischen Ableitungsregeln wie der Ketten- oder Produktregel. Dazu benötigt man jedoch eine explizite Formulierung der Funktion.

5.4.1 Parameter-Shift

Das wohl bekannteste Verfahren zur analytischen Gradientenberechnung auf Quantencomputern ist das Parameter-Shift Verfahren [SBG+19][MNKF18].

Sei lediglich das Gatter $\mathbb{G}(\theta)$ vom Parameter θ_i abhängig[3]. Die auf den Startzustand $|0\rangle^{\otimes N}$ ausgeführten unitären Operationen $U(\theta)$ lassen sich dann, lediglich Parameter θ_i betrachtend, aufschlüsseln als $U(\theta_i) = V\mathbb{G}(\theta_i)W$. Wenn man V in die Observable absorbiert und W in den Startzustand absorbiert, ist der Erwartungswert der Messung gegeben als $\langle\psi|\,\mathbb{G}^\dagger\hat{O}\mathbb{G}\,|\psi\rangle$. Für die Ableitung folgt dann:

$$\frac{\partial L(\theta)}{\partial\theta_i} = \frac{\partial}{\partial\theta_i}\langle\psi|\,\mathbb{G}^\dagger\hat{O}\mathbb{G}\,|\psi\rangle = \langle\psi|\,\frac{\partial}{\partial\theta_i}(\mathbb{G}^\dagger)\hat{O}\mathbb{G}\,|\psi\rangle + \langle\psi|\,\mathbb{G}^\dagger\hat{O}\frac{\partial}{\partial\theta_i}(\mathbb{G})\,|\psi\rangle \tag{5.12}$$

Wie in 2.5 beschrieben, lässt sich jede unitäre Matrix als Matrixexponential e^{iG} der hermiteschen Generatormatrix G schreiben. Nutzt man zusätzlich noch eine Identität der Matrizenrechnung, erhält man:

$$\begin{aligned}\frac{\partial L(\theta)}{\partial\theta_i} &= \frac{1}{2}(\langle\psi|\,(\mathbb{G}+\frac{\partial}{\partial\theta_i}\mathbb{G})^\dagger\hat{O}(\mathbb{G}+\frac{\partial}{\partial\theta_i}\mathbb{G})\,|\psi\rangle - \langle\psi|\,(\mathbb{G}-\frac{\partial}{\partial\theta_i}\mathbb{G})^\dagger\hat{O}(\mathbb{G}-\frac{\partial}{\partial\theta_i}\mathbb{G})\,|\psi\rangle)\\ &= \frac{1}{2}(\langle\psi|\,\mathbb{G}^\dagger(I-iG)^\dagger\hat{O}(I-iG)\mathbb{G}\,|\psi\rangle - \langle\psi|\,\mathbb{G}^\dagger(I+iG)^\dagger\hat{O}(I+iG)\mathbb{G}\,|\psi\rangle)\end{aligned}$$

Wenn sich $\mathbb{G}(\theta_i)$ schreiben lässt als $e^{i\theta_i G}$ und für den Generator G $G^2 = I$ gilt, folgt aus der Reihenentwicklung von Sinus und Kosinus, dass $\frac{1}{\sqrt{2}}(I \pm iG) = \mathbb{G}(\pm\frac{\pi}{4})$. Da für $\mathbb{G}$ gilt, dass $\mathbb{G}(a)\mathbb{G}(b) = \mathbb{G}(a+b)$ ist, ergibt sich schlussendlich:

$$\frac{\partial L(\theta)}{\partial\theta_i} = L(\theta+\frac{\pi}{4}) - L(\theta-\frac{\pi}{4}) \tag{5.13}$$

[3] Falls mehrere Gatter von einem Parameter abhängen, muss die Produktregel angewandt werden.

Der Name des Verfahrens ergibt sich daraus, dass nur die Parameterwerte verschoben werden, der sonstige Schaltkreis aber komplett identisch bleibt. Man kann in diesem Fall also den Gradienten eines Schaltkreises mit zwei Anwendungen desselben Schaltkreises berechnen.

Die Voraussetzung, dass $G^2 = I$ für das Gatter $\mathbb{G}(\theta)$ gilt, kann noch weiter abgeschwächt werden [HFQE22]. Es genügt, wenn G genau zwei Eigenwerte λ_1, λ_2 besitzt. Sei $2r = \lambda_1 - \lambda_2$. Falls es nicht schon der Fall ist, werden mittels Phaseshift die Eigenwerte auf $\lambda_1 = -\lambda_2 = r$ gesetzt. Dann gilt $G^2 = r^2 I$ und die Parameter-Shift Regel kann ebenfalls angewandt werden, jedoch mit anderen Shifts und anderem Skalierungsfaktor:

$$g_{i(PS)} = \frac{\partial L(\theta)}{\partial \theta_i} = r[L(\theta + \frac{\pi}{4r}) - L(\theta - \frac{\pi}{4r})] \tag{5.14}$$

Alle einzelnen Paulistrings erfüllen diese Eigenschaft als Generator, somit lässt sich beispielsweise der Gradient von den 1-Qubit Rotationsmatrizen R_x, R_y, R_z mittels Parameter-Shift berechnen.

Mit der allgemeinen (zentrierten) Parameter Shift Regel [MBK21][HFQE22] wurde gezeigt, dass man auch beliebig um $s \neq k\pi, k \in \mathbb{Z}$, unabhängig von r, den Parameter verschieben kann, solange man den Skalierungsfaktor richtig wählt:

$$g_{i(APS)} = \frac{\partial L(\theta)}{\partial \theta_i} = \frac{r[L(\theta + s) - L(\theta - s)]}{sin(2rs)} \tag{5.15}$$

Interessant ist die Ähnlichkeit aller vorgestellten Parameter-Shift Regeln zu zentralen finiten Differenzen Gleichung (5.6). Die beiden Verfahren unterscheiden sich nur im Bezug auf den Skalierungsfaktor der Differenz der beiden Samplewerte. In Abschnitt 6.4 wird dies weiter untersucht.

Wenn es Verfahren ähnlich zu zentralen Differenzen gibt, wären Verfahren ähnlich zu vorwärts oder rückwärts Differenzen von großem Vorteil, da, wie in Abschnitt 5.3.1 besprochen, der Messwert $L(\theta)$ an der Samplestelle dann für alle partiellen Ableitungen verwendet werden kann, wodurch man sich die Hälfte aller Messungen sparen könnte. In [HFQE22] wurde jedoch gezeigt, dass ein solches Verfahren nicht existieren kann.

5.4.2 Gatterzerlegung

Das Parameter-Shift Verfahren lässt sich nicht auf alle Gatter anwenden. Eine Möglichkeit trotzdem bezüglich eines Parameters θ_i eines beliebigen Gatters abzuleiten, besteht darin, das Gatter zu zerlegen [Cro19]. Wie in Abschnitt 2.5 beschrieben, lässt sich jedes n-Qubit Gatter in 1- und 2-Qubit Gatter zerlegen. Jedes 1-Qubit Gatter lässt sich wiederum in die drei Rotationsmatrizen R_x, R_y, R_z zerlegen, auf welche sich das Parameter-Shift Verfahren anwenden lässt.

Das kanonische 2-Qubit Gatter aus Gleichung (5.16) kann mit einzelnen 1-Qubit Gattern vor und hinter diesem Gatter jedes 2-Qubit Gatter darstellen [ZVSW03].

$$U_{CAN}(t_x, t_y, t_z) = e^{\frac{-i\pi}{2}(t_x X \otimes X + t_y Y \otimes Y + t_z Z \otimes Z)} \tag{5.16}$$

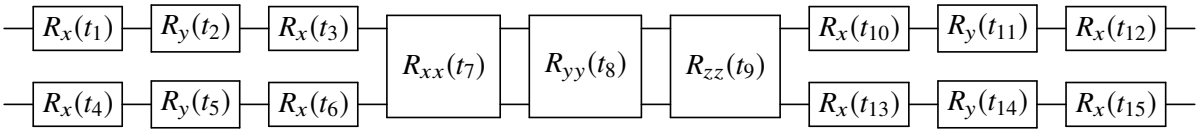

Abbildung 5.1: Zerlegung eines beliebigen 2-Qubit Gatters

Der Generator von U_{CAN} hat mehr als zwei unterschiedliche Eigenwerte. Da $X \otimes X$, $Y \otimes Y$ und $Z \otimes Z$ aber kommutieren, gilt $e^{A+B} = e^A e^B$. Somit kann U_{CAN} als Sequenz aufgefasst werden:

$$U_{CAN}(t_x, t_y, t_z) = e^{\frac{-i\pi}{2}(t_x X\otimes X)} e^{\frac{-i\pi}{2}(t_y Y\otimes Y)} e^{\frac{-i\pi}{2}(t_z Z\otimes Z)} = R_{xx}(t_x) R_{yy}(t_y) R_{zz}(t_z) \tag{5.17}$$

Auf jedes dieser Gatter ist das Parameter-Shift Verfahren anwendbar. Wenn man jede der 1-Qubit Operationen als drei Rotationen darstellt, benötigt man also insgesamt 15 Parameter für ein 2 Qubit Gatter (wie in Abbildung 5.1 dargestellt).

Wenn der Zusammenhang zwischen dem Parameter θ_i des ursprünglichen Gatters $U(\theta_i)$ und den Parametern t_i der Zerlegung $U_{CAN}(t_1(\theta_i), \ldots, t_{15}(\theta_i))$ bekannt ist, kann die Ableitung per Kettenregel bestimmt werden:

$$\frac{dL(\theta)}{d\theta_i} = \sum_{i=1}^{15} \frac{\partial L_{CAN}(t_1(\theta), \ldots, t_{15}(\theta))}{\partial t_i} \frac{dt_i}{d\theta_i} \tag{5.18}$$

Jeder $\frac{\partial L_{CAN}}{\partial t_i}$ Term lässt sich mit einer Anwendung von Parameter-Shift bestimmen. Die $\frac{dt_i}{d\theta_i}$ Terme sind klassische Berechnungen.

5.4.3 Linearkombination

Für eine andere Möglichkeit beliebige Gatter abzuleiten, wird in [SBG+19] $\frac{\partial U(\theta)}{\partial \theta_i}$ explizit als komplexe Matrix bestimmt und diese in eine Linearkombination aus unitären Matrizen A_k zerlegt. Eine Möglichkeit besteht darin, das Gatter $U(\theta)$ wieder als Matrixexponential $e^{iG(\theta)}$ aufzufassen. Nun geht man davon aus, dass der Generator als Linearkombination von unitären Matrizen G_k zerlegt wird (z.B. wie in Abschnitt 5.2 beschrieben in die Basis aus Paulistrings). Daraus ergibt sich dann folgende Ableitung für einen Parameter θ_i:

$$\frac{\partial U(\theta)}{\partial \theta_i} = \frac{\partial}{\partial \theta_i} \exp(\sum_k f_k(\theta) i G_k) = \sum_k \frac{\partial f_k(\theta)}{\partial \theta_i} i G_k U(\theta) \tag{5.19}$$

In Gleichung (5.12) eingesetzt ergibt sich daraus für die Ableitung des gesamten Schaltkreises:

$$\begin{aligned}\frac{\partial L(\theta)}{\partial \theta_i} &= \sum_k i \frac{\partial f_k(\theta)}{\partial \theta_i}(\langle\psi| \mathbb{G}^\dagger G_k^\dagger \hat{O} \mathbb{G} |\psi\rangle + \langle\psi| \mathbb{G}^\dagger \hat{O} G_k \mathbb{G} |\psi\rangle) \\ &= \sum_k \frac{i}{2} \frac{\partial f_k(\theta)}{\partial \theta_i}(\langle\psi| \mathbb{G}^\dagger (G_k + I)^\dagger \hat{O} (G_k + I) \mathbb{G} |\psi\rangle + \langle\psi| \mathbb{G}^\dagger (G_k - I)^\dagger \hat{O} (G_k - I) \mathbb{G} |\psi\rangle)\end{aligned}$$

Die Terme $\langle\psi| \mathbb{G}^\dagger (G_k + I)^\dagger \hat{O} (G_k + I) \mathbb{G} |\psi\rangle$ und $\langle\psi| \mathbb{G}^\dagger (G_k - I)^\dagger \hat{O} (G_k - I) \mathbb{G} |\psi\rangle$ lassen sich mithilfe eines Ancilla Qubits bestimmen [CW12]. Dabei müssen kontrollierte Versionen der G_k Gatter angewandt werden. Die $\frac{\partial f_k(\theta)}{\partial \theta_i}$ Terme sind klassische Berechnungen. Bei der Wahl der Zerlegung der Generatormatrix muss man abwägen zwischen dem Aufwand zur Bestimmung der Linearkombination, der Anzahl der Terme der Linearkombination (mehr Terme führen zu mehr nötigen Schaltkreisausführungen) und mit wie vielen Basisgattern sich einzelne G_k implementieren.

Neben den zwei vorgestellten Verfahren zur Ableitung bezüglich Parametern in beliebigen Gattern, gibt es noch weitere Verfahren, wie z.B. die stochastische Parameter-Shift Regel [BC21] bzw. die allgemeine stochastische Parameter-Shift Regel [WIWL21].

6 Experimente

Um die vorgestellten Gradientenverfahren hinsichtlich Performanz und Güte der Approximationen zu vergleichen, wurden Experimente durchgeführt und ausgewertet. Zuerst wird in Abschnitt 6.1 auf vorhandene Implementierungen zur Gradientenberechnung in Qiskit und Pennylane eingegangen. Danach werden in Abschnitt 6.2 im Detail die genutzten Ansätze, Observablen und Hyperparameter der in den Experimenten verwendeten Gradientenverfahren beschrieben. In Abschnitt 6.3 werden die Ergebnisse der Experimente ausgewertet und basierend darauf Hypothesen über die Verfahren aufgestellt. Zur Analyse dieser Hypothesen werden in Abschnitt 6.4 mathematisch fundierte Aussagen über die möglichen Berechnungsfehler getroffen. Zuletzt werden in Abschnitt 6.5 die Ergebnisse weiterer Experimente zur empirischen Validierung der Analyse besprochen.

6.1 Implementierung

Alle Experimente wurden in Qiskit [AAA+21] implementiert. Qiskit selbst stellt mit dem Qiskit Gradient Framework bereits drei Gradientenverfahren zur Verfügung. Zum einen gibt es das Parameter-Shift Verfahren, bei dem in diesem Fall der Parameter immer um $\pi/2$ verschoben wird. Als finite Differenzen Verfahren werden nur zentrale finite Differenzen unterstützt (intern wird ebenfalls die Parameter-Shift Klasse verwendet). Außerdem beträgt hier der Standardwert für den Abstand $h = 10^{-6}$. Wie sich später in Abschnitt 6.4 herausstellen wird, ist dies ein ungeeigneter Wert zur Anwendung auf NISQ Quantencomputern. Als Drittes gibt es noch ein Linearkombinationsverfahren welches 22 Gatter unterstützt. Falls im Schaltkreis andere Gatter verwendet werden, wird der Schaltkreis ähnlich zum Gatterzerlegungsverfahren transpiliert.

Pennylane [BIS+18] bietet nur finite Differenzen und Parameter-Shift Gradientenverfahren an, welche jedoch deutlich mehr Konfigurationsmöglichkeiten bieten. Im finite Differenzen Verfahren können zwar auch nicht beliebige Samplestellen verwendet werden, es kann jedoch eine höhere Approximationsordnung gewählt werden, bei der dann entweder eine vorwärts, rückwärts oder zentrale finite Differenzen Strategie zur Auswahl stehen. Zwei Samplestellen liegen dabei mit gleichbleibendem Abstand h auseinander. Beim Parameter-Shift Verfahren können eigene Shiftregeln als Tupel $[a_j, s_j, c_j]$ angegeben werden, die dann, wie in Gleichung (6.1) beschrieben, ausgewertet werden.

$$g_{iPS} = \sum_j c_j L(a_j \theta_i + s_j) \tag{6.1}$$

Falls keine eigenen Shift Regeln angegeben werden, wird versucht mithilfe eines, auf der Fourier-Transformation basierenden Verfahrens [WIWL21], Shift Regeln zu bestimmen. Schlägt auch dieser Versuch fehl, werden die Standardregeln $[1/2, 1, \pi/2], [-1/2, 1, -\pi/2]$, welche dem klassischen

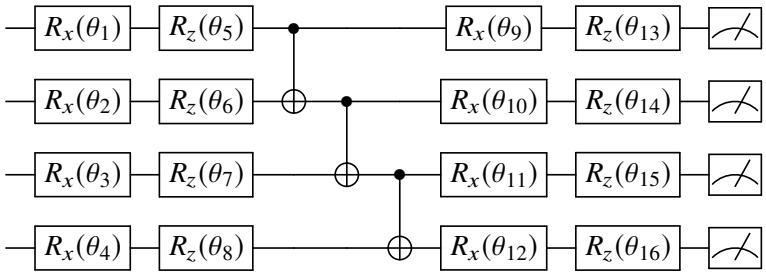

Abbildung 6.1: EfficientSU2 Schaltkreis mit linearer Verschränkungsschicht per CNOT Gatter, sowie R_x und R_z Gatter in der SU2-Schicht

Parameter-Shift Verfahren bezüglich z.B. des R_x Gatters entsprechen, verwendet. Zusätzlich unterstützt Pennylane automatische Differenzierung, falls der Schaltkreis auf einem passenden Simulator ausgeführt wird.

Da beide Frameworks sowohl einige der besprochenen Gradientenverfahren als auch für die unterstützten Verfahren benötigte Analysefunktionalitäten bisher nicht unterstützen, wurden diese in einem eigenen Qiskit Tool implementiert: `https://github.com/SchmidtMoritz/IAGQ`

6.2 Versuchsaufbau

Viele verschiedene Faktoren spielen bei der Gradientenberechnung eine Rolle. Der Ansatz bestimmt bezüglich welcher Gatter Ableitungen berechnet werden müssen. Die Observable impliziert über ihre Dimensionalität die Mindestanzahl an benötigten Qubits und auch die Anzahl und Orientierungen der Eigenvektoren, sowie die Größenordnungen der Eigenwerte beeinflussen die Gradientenberechnung. Außerdem können diese beiden Komponenten auch wieder Hyperparameter haben: Zum Beispiel sind Ansätze häufig in Form von sich wiederholenden Schichten aufgebaut, mit der man die Tiefe des Schaltkreises skalieren kann. Sowohl die Anzahl der Schichten, als auch die Gatter, die in den Schichten verwendet werden, lassen sich als Hyperparameter konfigurieren.

In den Experimenten wurde der in Qiskit als EfficientSU2 bezeichnete Schaltkreis verwendet, der konzeptionell dem Harware Efficient Ansatz [KMT+17] ähnelt, der für den VQE Algorithmus konzipiert wurde. Für eine genauere Analyse des Ansatzes wird auf [FHJ+21] verwiesen. Der Schaltkreis besteht aus l Schichten aus Gattern. Jede Schicht besteht wiederum aus einer Verschränkungsschicht gefolgt von einer Schicht aus 1-Qubit Gattern. Dabei wird immer eine Schicht aus 1-Qubit Gattern als nullte Schicht vorangestellt, da Verschränkungsschichten angewandt auf den Startzustand $|0\rangle^{\otimes N}$ keinen Effekt hätten. Als 1-Qubit Gatter wurden die Rotationsgatter R_x und R_z, sowie CNOT-Gatter als Verschränkungsgatter angewandt. Bei der Art der Verschränkungsgatter wurden sowohl lineare (siehe Abbildung 6.1), als auch vollständige Schichten (siehe Abbildung 6.2) verwendet.

Als Observable wurde in den Experimenten eine einfache Diagonalmatrix, wie in Gleichung (6.2) dargestellt, verwendet. Die Eigenwerte wurden als $\lambda_i = 2i$ gewählt, ähnlich zur globalen Observable im Variational Quantum State Eigensolver [CSAC20], bei der die Eigenwerte ebenfalls monoton steigen $\lambda_i < \lambda_{i+1}$.

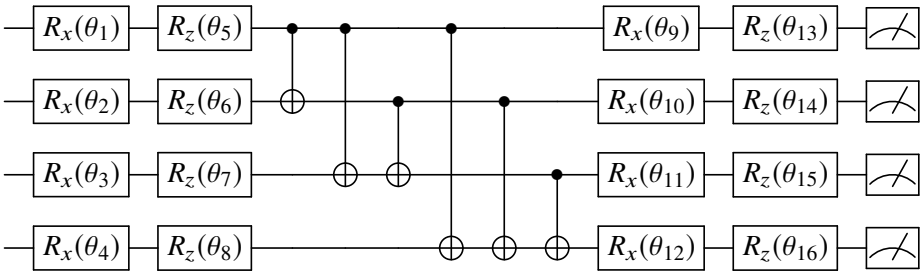

Abbildung 6.2: EfficientSU2 Schaltkreis mit voller Verschränkungsschicht per CNOT Gatter, sowie R_x und R_z Gatter in der SU2-Schicht

$$O_{Diag}(\lambda_1, \ldots, \lambda_N) = \begin{pmatrix} \lambda_1 & 0 & \ldots & \ldots & 0 \\ 0 & \lambda_2 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & \ldots & 0 & \lambda_N \end{pmatrix} \tag{6.2}$$

In der Implementierung lassen sich aber auch alternativ die H_2 und LiH Observablen aus [KMT+17], sowie die $He-H^+$ Observable aus [PMS+14] (für ausgewählte Bindungslängen $R = 0.05, 0.1, 1, 2, 3$) als Observable verwenden. Für diese Observablen wurden ebenfalls Experimente durchgeführt. Da es keine signifikanten Unterschiede bei den Ergebnissen gab, werden im Weiteren die Ergebnisse der Experimente vorgestellt, bei denen die Observable aus Gleichung (6.2) verwendet wurde. Insgesamt entspricht der Versuchsaufbau damit einem typischen Aufbau eines VQE Algorithmus.

Der Gradient ist immer abhängig von der Parameterposition, an welcher er gemessen wird. Eine Abtastung des gesamten Parameterraumes mit einem feinen Gitter, um allgemeine Aussagen treffen zu können, ist anhand der Laufzeiten einzelner Messungen praktisch nicht möglich. In dieser Arbeit wurden daher zufällige Parameterpunkte für die Experimente genutzt.

Diese Faktoren werden ergänzt durch die Hyperparameter der Gradientenverfahren selbst. Dazu gehören zum Beispiel die Anzahl und Abstände der Samplepunkte bei finiten Differenzen oder der Abstand und die Anzahl an Gradientenapproximationen, über die bei SPSA gemittelt werden soll. Was bei der Wahl dieser Hyperparameter beachtet werden muss, lag im Fokus der Experimente und Analyse und wird später detailliert besprochen.

Da die Experimente als Grundlage zur Anwendung auf NISQ Quantencomputern dienen sollen, spielen noch zusätzliche Faktoren, wie eine begrenzte Anzahl Shots und ein verrauschtes Ausführen der Schaltkreise, eine Rolle. In der Praxis sollten die simpleren Verfahren in ähnlichem Maße von Rauschen und ungenauen Gattern betroffen sein, da die untersuchten Methoden meist die gleichen Schaltkreise verwenden und sich nur in Parameterwerten und klassischen Nachberechnungen unterscheiden. Die auf Gatterzerlegung und Linearkombination basierten Verfahren, bei denen der Schaltkreis verändert wird, könnten unterschiedlich stark von Gatterfehlern und Rauschen betroffen sein. Dabei muss aber immer der Einzelfall betrachtet werden, da es von den genutzten Gattertypen abhängt. Weil in den Experimenten nur die Verfahren untersucht wurden, die keine Schaltkreismodifikationen benötigen, wurden zur einfacheren Durchführung alle Experimente mittels QASM-Simulator ausgeführt. Wie gut die Ergebnisse der Verfahren für verschiedene Shotzahlen sind, wurde im Detail untersucht.

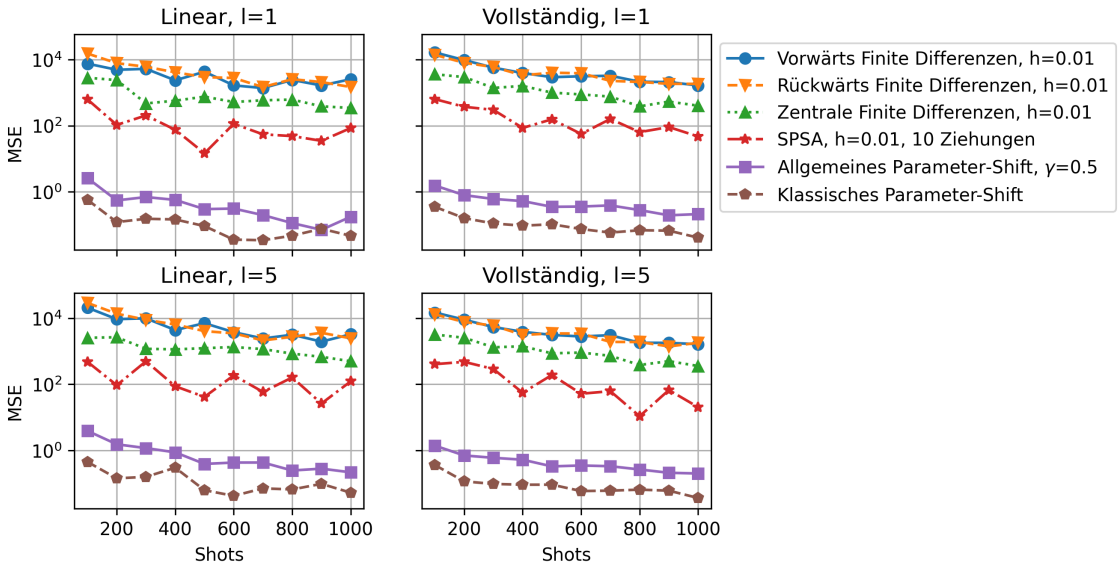

Abbildung 6.3: Vergleich der mittleren quadratischen Fehler an zufälligem Parameterpunkt mit Anzahl Ansatzschichten jeweils $l = 1$ und $l = 5$ und linearer sowie vollständiger Verschränkungsschicht.

Für den gewählten Ansatz lässt sich auf allen Gattern das Parameter-Shift Verfahren anwenden. Die Experimente fokussieren sich auf die vorgestellten numerischen und Parameter-Shift Verfahren. Die allgemeinen Gradientenverfahren, also die Verfahren per Linearkombination und Gatterzerlegung, wurden in den Experimenten nicht angewandt. Was sich mithilfe der Ergebnisse dieser Arbeit über diese Verfahren aussagen lässt, wird in Abschnitt 6.5 besprochen.

6.3 Versuchsergebnisse

Um einen ersten Eindruck der Güte der einzelnen Verfahren zu gewinnen, wurden an einem zufälligen Parameterpunkt alle Verfahren angewandt. Als analytische Verfahren wurden Parameter-Shift, sowie das allgemeine Parameter-Shift Verfahren mit Abstand von $\gamma = 0.5$ genutzt. Als numerische Verfahren wurden zentrale, vorwärts und rückwärts finite Differenzen, sowie SPSA genutzt, wobei bei allen $h = 0.01$ gewählt wurde. Da SPSA weniger Shots als die anderen Verfahren verwendet, wurde, wie in Abschnitt 5.3.2 besprochen, der Mittelwert über 10 Ausführungen gebildet.

Es wurden bei 4 Qubits verschiedene Anzahl an Schichten mit vollständiger und linearer Verschränkungsschicht getestet. Pro Messung wurden in den ersten Experimenten bis zu 1000 Shots verwendet. Auch in den folgenden Abbildungen beschreibt die Shotanzahl immer die Anzahl Shots die für jede individuelle Messung angewandt wurden.

Wie in Abbildung 6.3 zu sehen, ist das Verhältnis zwischen den Fehlern der verschiedenen Gradientenverfahren für alle Varianten von Verschränkungsschichten und Anzahl der Schichten ähnlich. Die Anzahl der Schichten des Ansatzes, sowie die Art der Verschränkungsschichten, schien auf die Güte der Gradientenverfahren keinen Einfluss gehabt zu haben. Bei genauerer Betrachtung dieser ersten Experimente lassen sich einige Hypothesen bezüglich der Verfahren formulieren:

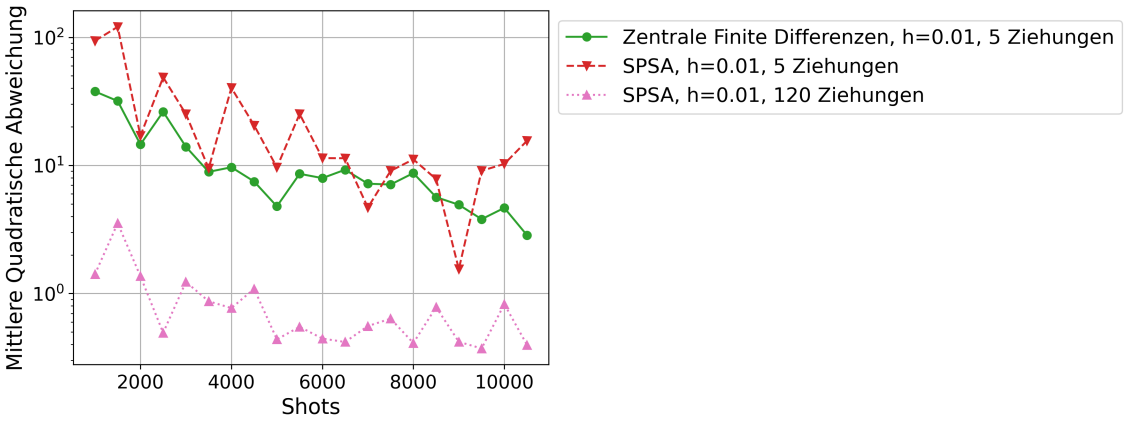

Abbildung 6.4: Vergleich der Fehler zwischen SPSA und Finite Differenzen mit gleichem Abstand $h = 0.01$ und mit bzw. ohne ausgeglichener gesamter Shotanzahl

Hypothese 1: Das klassische und allgemeine Parameter-Shift Verfahren berechnen deutlich bessere Approximationen als alle numerischen Verfahren.

Hypothese 2: Vorwärts und rückwärts finite Differenzen berechnen schlechtere Approximationen als zentrale finite Differenzen bei gleichem Abstand h.

Hypothese 3: SPSA berechnet bessere Approximationen als finite Differenzen.

Hypothese 3 entspricht dabei nicht den Erwartungen zu SPSA in anderen Problemstellungen, da der eigentliche Vorteil von SPSA darin besteht, schlechtere, aber dafür weniger aufwendige, Approximationen iterativ auszugleichen. Auch hat SPSA bei den Versuchen insgesamt weniger Shots benötigt als alle anderen Verfahren, nämlich nur zwei Messungen pro Gradientenapproximation, während die anderen Verfahren $2n$ Messungen benötigten. Schon bei nur einer Ansatzschicht und 4 Qubits gibt es 16 Parameter. Alle Verfahren außer SPSA benötigten also 32 Messungen. Da bei den Experimenten bei SPSA über 10 Approximationen gemittelt wurde, waren dafür insgesamt trotzdem nur 20 Messungen nötig.

Um dies weiter zu untersuchen, wurde bei $l = 2$, also $k = 24$ Parametern, sowohl SPSA, als auch zentrale finite Differenzen mit Abstand $h = 0.01$ über 5 Approximationen gemittelt. Wie in Abbildung 6.4 zu sehen, scheinen dann beide Verfahren ähnliche Fehler zu liefern. Wenn man die gesamte Shotanzahl jedoch ausgleicht, also finite Differenzen weiterhin fünf mal mittelt, SPSA aber $5 * k = 120$ mal, liefert SPSA bessere Ergebnisse.

Bisher wurden alle numerischen Verfahren mit dem gleichen Abstand $h = 0.01$ getestet. Im Allgemeinen will man ein möglichst kleines h wählen, um den Fehlerterm $O(h^2)$ zu verringern. Wenn man spezifisch zentrale finite Differenzen als den Differenzenquotienten der Ableitungsdefinition betrachtet, ergibt die Wahl eines kleineren Abstands ebenfalls Sinn, da ein kleineres h einem besseren Annähern des Grenzwertes und damit der exakten Ableitung entspricht.

Experimente dazu zeigen aber das genaue Gegenteil. Wie in Abbildung 6.5 zu sehen, liefern größere Abstände bessere Ergebnisse. Auch mehr Samplestellen mit geringeren Abständen scheinen dies nicht zu verbessern.

Hypothese 4: Finite Differenzen mit kleinerem h berechnen schlechtere Approximationen.

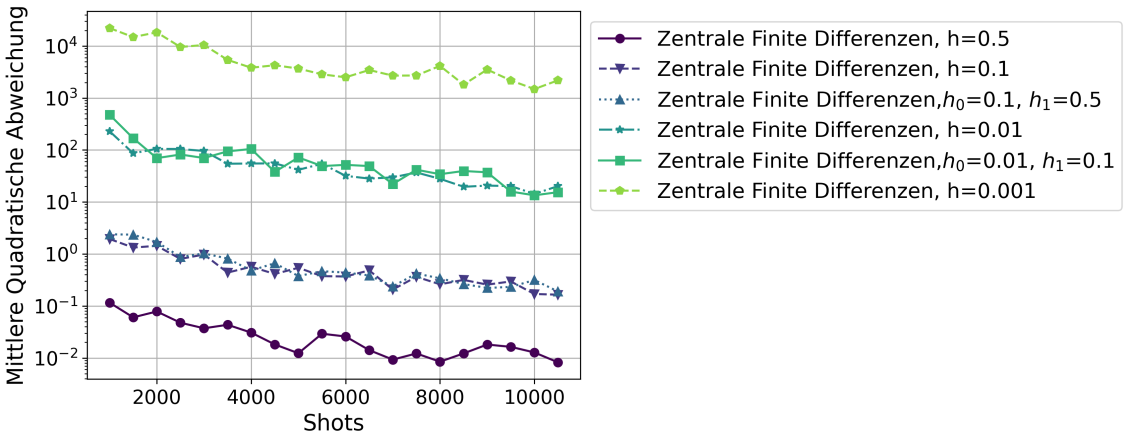

Abbildung 6.5: Vergleich der Fehler von zentralen finiten Differenzen mit verschiedenen Abständen

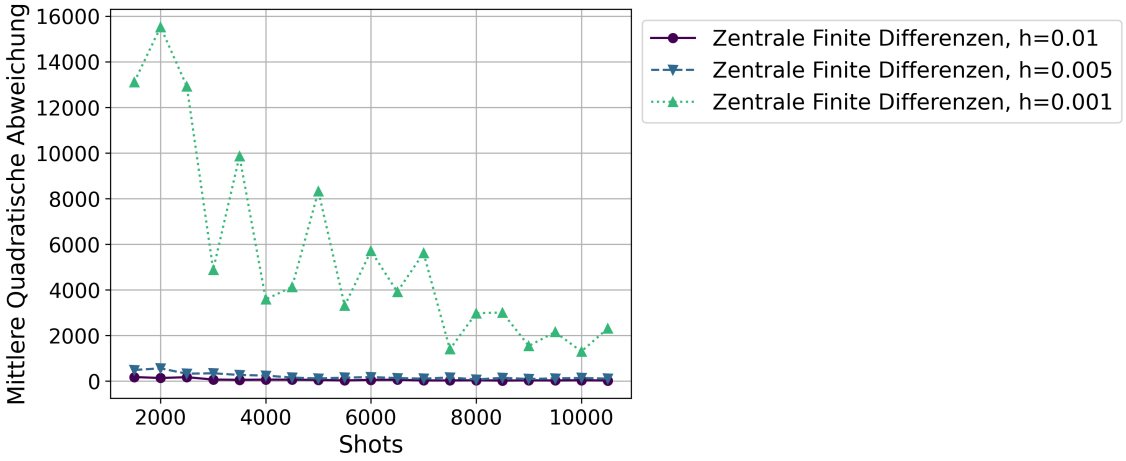

Abbildung 6.6: Entwicklung des Fehlers von zentralen finiten Differenzen mit $h = 0.01$, $h = 0.005$ und $h = 0.001$ bei großen Shotzahlen

Auch wenn geringere Abstände größere Fehler produzieren, sinkt dieser Fehler kontinuierlich mit höherer Anzahl Shots. Weitere Experimente (siehe Abbildung 6.6) zeigen, dass kleinere Abstände h bei geringen Shotzahlen nicht nur einen großen Fehler erzielen, sondern auch deutlich mehr Shots benötigen, um ähnliche Fehler wie die analytischen Verfahren zu erreichen.

Hypothese 5: Finite Differenzen benötigen mehr Shots je kleiner h gewählt wird, um ähnliche Fehlerraten wie Parameter-Shift zu erreichen.

In den ersten Experimenten ließ sich vor allem der Unterschied der Approximationsqualität der numerischen zu den analytischen Verfahren erkennen. Aber auch bei den analytischen Verfahren gibt es Unterschiede. Das allgemeine Parameter-Shift Verfahren sollte eigentlich, genau wie das klassische Parameter-Shift Verfahren, den exakten Gradienten bestimmen. Wenn man nur die beiden getesteten analytischen Verfahren mit bis zu 10000 Shots vergleicht (siehe Abbildung 6.7), fällt auf, dass das allgemeine Verfahren mit $h = 0.5$ zwar auch einen sehr kleinen quadratischen Fehler erzielt, es dafür aber mehr Shots aufwenden muss als das einfache Parameter-Shift Verfahren.

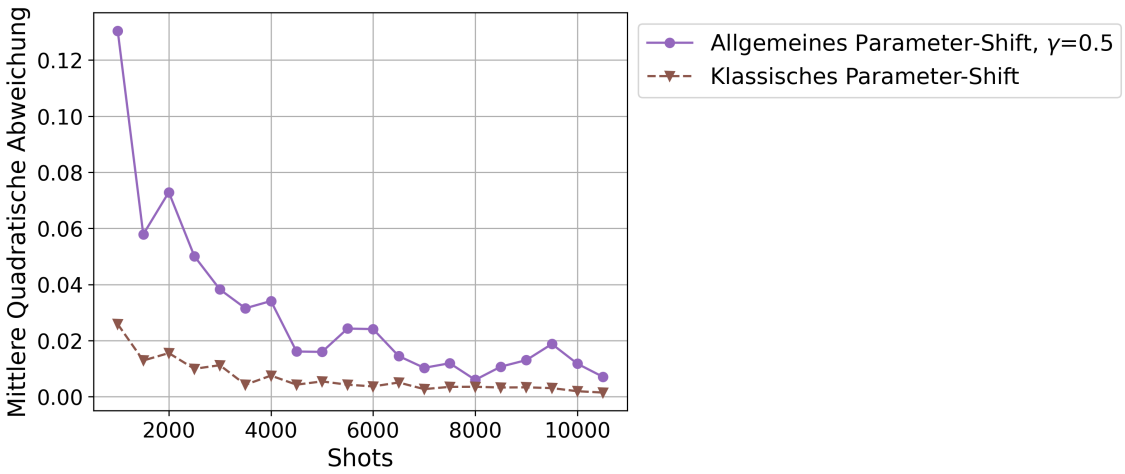

Abbildung 6.7: Vergleich der Fehler vom allgemeinen und klassischen Parameter Shift Verfahren

Hypothese 6: Bei dem allgemeinen und klassischen Parameter-Shift Verfahren gibt es leichte Unterschiede hinsichtlich der Anzahl Shots die benötigt werden, um den exakte Gradienten zu berechnen.

In der bisherigen Auswertung wurden die Fehler der Verfahren vor allem untereinander verglichen. Um zu verstehen, wie stark die Auswirkungen von großen Fehlern in möglichen Anwendungen wären, sollten die Ergebnisse noch mit der Größe des Gradienten in Kontext gesetzt werden. Die durchschnittliche Größe einer partiellen Ableitung an dem verwendeten, zufälligem Parameterpunkt war circa 0.75. Die Fehler der analytischen Verfahren erreichen nach 1000 Shots eine geringere Größenordnung, sind jedoch in noch keinem vernachlässigbaren Bereich. Im Gegensatz dazu entsprechen die Fehler der numerischen Verfahren nach 1000 Shots dem 5- bis 50-fachen der durchschnittlichen Gradientengröße. Die Approximationen der numerischen Verfahren sollten daher zur Optimierung nicht verwendet werden, da das Rauschen des Fehlers der tatsächlichen Gradienteninformation überwiegt.

6.4 Analyse

Die Ergebnisse der Experimente werden nun weiter aufgeschlüsselt und mathematisch analysiert, um für die aufgestellten Hypothesen mögliche Erklärungen zu finden. Aufgrund des durch die Messungen entstehenden Zufalls, lassen sich die konkreten Gradientenberechnungen als Zufallsvariablen auffassen und deren quadratische Fehler in Bias und Varianz aufteilen. Der Bias entspricht dabei dem Fehler des Verfahrens, wenn bei jeder Messung der exakte Erwartungswert gemessen würde. Die Varianz beschreibt die zu erwartende quadratische Abweichung vom Bias. Messungen mit mehr Shots verringern die Varianz des Messergebnisses. Für unendlich Shots geht die Varianz

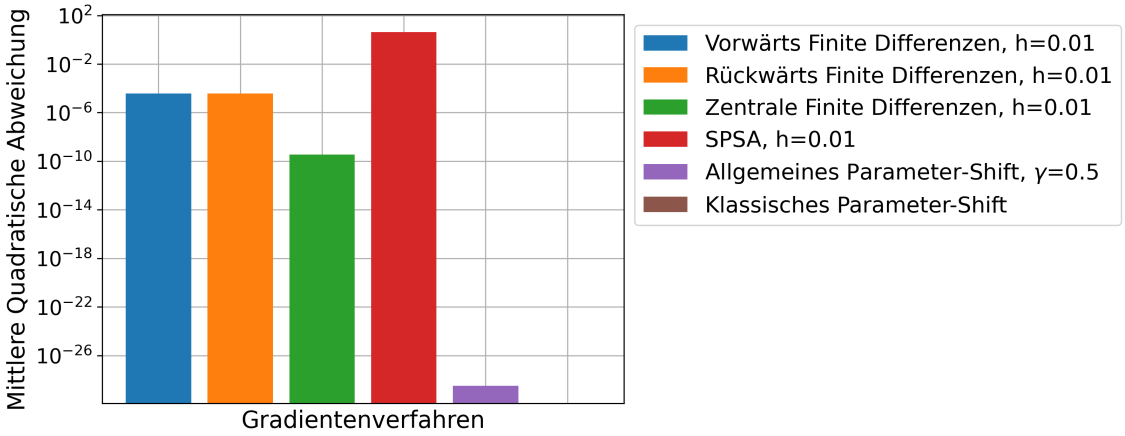

Abbildung 6.8: Vergleich des Bias der Verfahren aus den vorherigen Experimenten

gegen 0, man misst also sicher den Erwartungswert und es bleibt lediglich der Bias als Restfehler. Gleichung (6.3) zeigt diese Zusammensetzung des quadratischen Fehlers als Bias und Varianz für partielle Ableitung $\frac{\partial L(\theta)}{\partial \theta_i}$ und Approximation g_i.

$$\mathbb{E}\left[\left(g_i - \frac{\partial L(\theta)}{\partial \theta_i}\right)^2\right] = \mathbb{E}[(\mathbb{E}[g_i] - g_i)^2] + \left(\mathbb{E}[g_i] - \frac{\partial L(\theta)}{\partial \theta_i}\right)^2 = Var(g_i) + Bias\left(g_i, \frac{\partial L(\theta)}{\partial \theta_i}\right)^2 \tag{6.3}$$

Mit dem Qiskit Opflow Modul lassen sich die genauen Erwartungswerte $\mathbb{E}[L(\theta)]$ von Messungen bestimmen. Aufgrund der Linearität des Erwartungswerts lassen sich die Erwartungswerte $\mathbb{E}[g_i]$ für alle Verfahren aus den ersten Experimenten berechnen, also die Gradienten, zu denen die Verfahren schlussendlich unter Verwendung von unendlich vielen Shots konvergieren würden. In Gleichung (6.4) wird dies beispielhaft für das klassische Paramete-Shift Verfahren demonstriert.

$$\mathbb{E}[g_{i(PS)}] = \mathbb{E}[\frac{1}{2}(L(\theta + \frac{\pi}{2}) - L(\theta - \frac{\pi}{2}))] = \frac{1}{2}(\mathbb{E}[L(\theta + \frac{\pi}{2})] - \mathbb{E}[L(\theta - \frac{\pi}{2})]) \tag{6.4}$$

Für die Rotationsgatter entsprechen die Erwartungswerte des klassischen Parameter-Shift Verfahrens den exakten Ableitungen $\frac{\partial L(\theta)}{\partial \theta_i}$. Der Bias eines Verfahrens lässt sich also bestimmen, indem deren Erwartungswert mit dem vom klassischen Parameter Shift verglichen wird. Abbildung 6.8 zeigt den quadrierten Bias für alle Verfahren der ersten Experimente.

Wichtig ist hierbei, dass der Erwartungswert von SPSA nur über die spezifischen 10 gewählten Abstandsvektoren Δ gebildet wird und nicht der allgemeine Erwartungswert des Verfahrens ist. Die Analyse von SPSA ist insgesamt komplizierter, da sowohl anhand der Auswahl des Abstandsvektors, als auch durch die Messungen selbst ein verschachtelter, doppelter Zufall entsteht. Daher wird die Analyse von SPSA gesondert in Abschnitt 6.4.3 besprochen.

Es zeigt sich, dass das allgemeine Parameter-Shift Verfahren wie erwartet, bis auf Maschinenpräzision genau, ebenfalls zum korrekten Gradienten konvergiert, also keinen Bias hat. Die numerischen Verfahren sind dagegen nicht ohne Bias. SPSA sticht hier, den Erwartungen entsprechend am

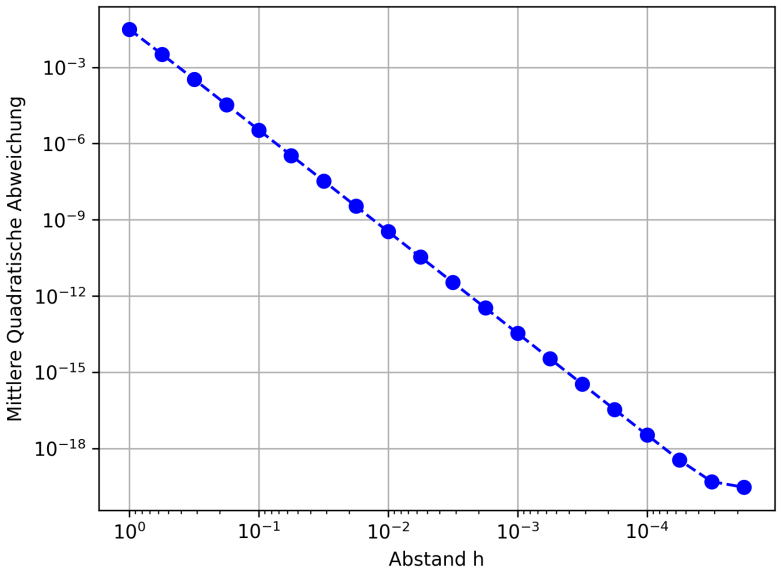

Abbildung 6.9: Vergleich des Bias von zentralen finiten Differenzen für verschiedene Abstände h an dem Parameterpunkt der ersten Experimente

schlechtesten ab. Bei den finiten Differenzen Verfahren hat das zentrale Verfahren bei selbem Abstand einen geringeren Bias als die anderen beiden Verfahren. Wie in Abbildung 6.9 zu sehen, spiegelt sich für den Bias auch das eigentlich erwartete Verhalten wider, dass kleinere Abstände zu geringeren Fehlern führen[1].

Allgemein fällt auf, dass der Bias nur einen sehr geringen Teil des Fehlers der ersten Experimente ausmacht. Der Großteil des Fehlers muss also von der Varianz stammen.

6.4.1 Varianz

Wie in Abschnitt 2.3 besprochen, ist im allgemeinen Fall der Erwartungswert einer Messung gegeben durch $\langle O \rangle_{|\psi\rangle} = \langle \psi | O | \psi \rangle$. Die Varianz der Messung lässt sich, analog zu ihrer Definition $Var(X) = \mathbb{E}[(X - \mathbb{E}[X])^2]$, als $\langle \psi | (O - \langle O \rangle_{|\psi\rangle} I)^2 | \psi \rangle$ berechnen. Eine alternative Berechnung ergibt sich, analog zur bekannten Identität $Var(X) = E[X^2] - E[X]^2$, als $\langle \psi | (O^2) | \psi \rangle - (\langle \psi | O | \psi \rangle)^2$. In [HFQE22] und [MBK21] wird diese Varianz als One-Shot bzw. Single-Shot Varianz bezeichnet, da es die Varianz einer Messung beschreibt, bei der nur ein einziger Shot verwendet wurde.

Die Varianz ist nicht linear. Es werden also einige weitere Varianzidentitäten benötigt, um von der Varianz einzelner Messungen auf die Varianz der verschiedenen Gradientenverfahren zu schließen. Eine Linearkombination zweier Zufallsvariablen lässt sich, wie in Gleichung (6.5) beschrieben, in die einzelnen Varianzen der Zufallsvariablen sowie deren Kovarianz zerlegen.

$$Var(aX + bY) = a^2 Var(X) + b^2 Var(Y) + 2abCov(X, Y) \tag{6.5}$$

[1] In [HFQE22] wird gezeigt, dass der Bias der finiten Differenzen Verfahren mithilfe der allgemeinen Parameter-Shift Regel als Linearkombination von erster und zweiter Ableitung dargestellt werden kann. Auch dort zeigt sich, dass die Approximationen für $\lim_{h \to 0}$ exakt werden.

Für unabhängige Zufallsvariablen folgt daraus, dass die Varianz der Linearkombination lediglich die Summe der einzelnen Varianzen mit quadrierten Koeffizienten ist (siehe Gleichung (6.6)), da in diesem Fall die Kovarianz 0 ist.

$$Var(aX + bY) = a^2 Var(X) + b^2 Var(Y) \tag{6.6}$$

Wenn in einem Quantenschaltkreis mehrmals gemessen wird, kann eine frühe Messung den Zustand des Quantencomputers verändern und dadurch spätere Messungen beeinflussen. In dem Fall sind die Zufallsvariablen, die die Messergebnisse dieser Messungen beschreiben, nicht unabhängig voneinander. Für das hier besprochene Szenario ist dies jedoch nicht der Fall, da nur am Ende eines Schaltkreises einmal gemessen wird. Jede Messung der Verlustfunktion entspricht einer neuen Ausführung des Schaltkreises. Daher sind die Messungen $L(\theta)$ hier voneinander unabhängig. Die Varianzen der Gradientenverfahren ergeben sich dann wie folgt:

$$Var(g_{i(PS)}) = r^2(Var(L(\theta + \frac{\pi}{4r})) + Var(L(\theta - \frac{\pi}{4r}))) \tag{6.7}$$

$$Var(g_{i(VFD)}) = Var\left(\frac{1}{h}(L(\theta + h) - L(\theta))\right) = \frac{1}{h^2}(Var(L(\theta + h)) + Var(L(\theta))) \tag{6.8}$$

$$Var(g_{i(RFD)}) = Var\left(\frac{1}{h}(L(\theta) - L(\theta - h))\right) = \frac{1}{h^2}(Var(L(\theta)) + Var(L(\theta - h))) \tag{6.9}$$

$$Var(g_{i(ZFD)}) = \frac{1}{4h^2}(Var(L(\theta + h)) + Var(L(\theta - h))) \tag{6.10}$$

$$Var(g_{i(APS)}) = \frac{r^2}{sin(2r\gamma)^2}(Var(L(\theta + \gamma)) + Var(L(\theta - \gamma))) \tag{6.11}$$

Mithilfe der One-Shot Varianz ergibt sich auch die Varianz nach beliebig vielen Shots. Jeder Shot einer Messung lässt sich als eine Zufallsvariable $L(\theta)_s$ darstellen. Dabei ist jedes $L(\theta)_s$ identisch verteilt mit $\mathbb{E}[L(\theta)_s] = \mu$ und $Var(L(\theta)_s) = \sigma^2$, aber unabhängig von allen anderen Shots. Das Gesamtergebnis nach n Shots ist dann der Mittelwertschätzer $\bar{L}(\theta)$ über alle $L(\theta)_s$, also $\bar{L}(\theta) = \frac{1}{n}\sum_{s=0}^{n} L(\theta)_s$. Der Erwartungswert des Mittelwertschätzers entspricht dem der Originalverteilung unabhängig von der Anzahl an Shots (siehe Gleichung (6.12)).

$$\mathbb{E}[\bar{L}(\theta)] = \mathbb{E}[\frac{1}{n}\sum_{s=0}^{n} L(\theta)_s] = \frac{1}{n}\sum_{s=0}^{n} \mathbb{E}[L(\theta)_s] = \frac{1}{n} n\mu = \mu \tag{6.12}$$

Die Varianz des Mittelwertschätzers nimmt hingegen mit der Anzahl an Shots ab (siehe Gleichung (6.13)).

$$Var(\bar{L}(\theta)) = Var(\frac{1}{n}\sum_{s=0}^{n} L(\theta)_s) = \frac{1}{n^2}\sum_{s=0}^{n} Var(L(\theta)_s) = \frac{1}{n^2} n\sigma^2 = \frac{\sigma^2}{n} \tag{6.13}$$

Wenn also die One-Shot Varianz an den Messpunkten bekannt ist, lässt sich die Varianz der Verfahren für beliebig viele Shots berechnen. Solange für alle Messungen dieselbe Anzahl n an Shots verwendet wird, ergibt sich die verringerte Varianz als $\frac{Var(g_i)}{n}$.

Die Verfahren messen meist an unterschiedlichen Parameterpunkten. Man kann deren Varianzen daher eigentlich nicht direkt vergleichen, da sich die Varianzen der Messungen je nach Parameterpunkt unterscheiden können. Folgende Annahme aus [MBK21] erlaubt jedoch eine andere Perspektive:

Annahme 1: Die Varianz der Messung in einer Observable hängt nur schwach von der Verschiebung des Parameterwertes ab, sodass $Var(L(\theta + h)) + Var(L(\theta - h)) \backsimeq 2Var(L(\theta))$ für alle Werte von h gilt.

Für diese Annahme lässt sich leicht ein Gegenbeispiel konstruieren: Betrachtet man ein einzelnes Qubit im Zustand $|0\rangle$ auf das ein $R_y(\theta)$ Gatter angewandt wird und misst es in der Standardbasis, also bezüglich der Zustände $|0\rangle$ und $|1\rangle$, gibt es im Fall $\theta = 0$ keine Varianz. Wenn man nun aber an $\theta = 0$ den Parameter um den Abstand $h = \frac{\pi}{2}$ verschiebt, ist der resultierende Zustand $|+\rangle$ bei $R_y(\theta + h)$, beziehungsweise $|-\rangle$ bei $R_y(\theta - h)$. Diese beiden Zustände sorgen für die größte Varianz, die in dieser Konstruktion möglich ist. Hier gilt also $Var(L(\theta + h)) + Var(L(\theta - h)) \backsimeq 2Var(L(\theta))$ nicht. Es scheint also zunächst nicht sinnvoll diese Annahme zu akzeptieren.

Wie oben besprochen ist die Varianz messbar. Eine empirische Untersuchung in Form einer Stichprobe an verschiedenen zufälligen Parameterpunkten unterstützt die Annahme. Gerade bei kleinen Abständen unterscheiden sich $Var(L(\theta + h))$, $Var(L(\theta - h))$ und $Var(L(\theta))$ kaum voneinander.

Eine mögliche Erklärung für dieses Verhalten könnte das Barren Plateau Phänomen [MBS+18] sein. Es weist Parallelen auf zu dem Phänomen des Curse of Dimensionality [Bel57] auf, welches unter anderem im maschinellen Lernen bekannt ist. Die Anzahl an Zuständen, deren Erwartungswerte sich vom durchschnittlichen Erwartungswert signifikant unterscheiden, sinkt exponentiell mit der Anzahl an Qubits. In Plateaus aus Zuständen mit durchschnittlichen Erwartungswerten ist dann der Gradient sehr gering, jedoch kein Extremum vorhanden.

Eine Idee ist daher, auch wenn das Problem mit 4 Qubits noch relativ klein ist, das dieses Phänomen hier bereits Auswirkungen hat. Erst wenn man sich an dem Punkt θ, beziehungsweise den Verschiebungen um h, einem Eigenzustand der Observablen nahe genug annähert, unterscheidet sich die Varianz der Messung deutlich von der durchschnittlichen Varianz. Wenn man davon ausgeht, dass ein gleich verteilter zufälliger Parameterpunkt circa zu einem gleich verteilten zufälligen Zustand führt, ist die Chance deutlich höher einen Zustand mit durchschnittlicher Varianz zu treffen, als einen Zustand in der Nähe eines Eigenvektors. Dieses Phänomen bietet definitiv Möglichkeiten für eine zukünftige gesonderte Analyse. Aufgrund der empirischen Ergebnisse wird die Annahme als zutreffend angenommen und für die weitere Analyse verwendet.

Die Varianzen der Verfahren aus Gleichung (6.7)-Gleichung (6.11) werden dementsprechend angepasst. Alle Varianzen von Messungen werden als konstante Varianz σ^2 angenommen: $Var(L(\theta)) \simeq Var(L(\theta + h)) \simeq \sigma^2$.

$$Var(g_{i(PS)}) \simeq 2\sigma^2 r^2 \tag{6.14}$$

$$Var(g_{i(VFD)}) \simeq \frac{2\sigma^2}{h^2} \tag{6.15}$$

$$Var(g_{i(RFD)}) \simeq \frac{2\sigma^2}{h^2} \tag{6.16}$$

$$Var(g_{i(ZFD)}) \simeq \frac{\sigma^2}{2h^2} \tag{6.17}$$

$$Var(g_{i(APS)}) \simeq \frac{2\sigma^2 r^2}{sin(2r\gamma)^2} \tag{6.18}$$

Wir haben nun also eine Abschätzung der Varianz und wissen, dass diese für den Großteil des Fehlers in den Experimenten verantwortlich ist. Als Nächstes gilt es, mithilfe dieser Erkenntnisse die in Abschnitt 6.3 aufgestellten Hypothesen zu interpretieren.

6.4.2 Analyse der Hypothesen

Wie in Gleichung (6.14) - Gleichung (6.18) zu sehen, hängt die Varianz nun nur von den Faktoren ab, die in den ursprünglichen Gradientenformeln die Messwerte skalieren. Für die Rotationsgatter des benutzten Ansatzes ist $r = \frac{1}{2}$, folglich ist die Varianz des klassischen Parameter-Shifts $\frac{\sigma^2}{2}$. Für die numerischen Verfahren hängt der Vorfaktor vom Abstand h mit $O(h^{-2})$ ab. Da üblicherweise ein Abstand $0 < h \ll 1$ gewählt wird, um einen geringeren Bias zu produzieren, wird die Varianz aufgebläht. Beispielsweise für das rückwärts finite Differenzen Verfahren aus den Experimenten mit dem Abstand $h = 0.01$ wird σ^2 bereits mit $2(\frac{1}{0.01})^2 = 20000$ skaliert. Für das zentrale, finite Differenzen Verfahren aus den Experimenten mit dem Abstand $h = 0.01$ wird σ^2 bereits mit $\frac{1}{2 \cdot 0.01^2} = 5000$ skaliert. Es erzielen also erst 10000 Shots pro Messung die Varianz, die Parameter-Shift mit einem einzigen Shot pro Messung generiert.

Hypothese 4: Finite Differenzen mit kleinerem h berechnen schlechtere Approximationen.

Hypothese 5: Finite Differenzen benötigen mehr Shots je kleiner h gewählt wird, um ähnliche Fehlerraten wie Parameter-Shift zu erreichen.

Diese zwei Hypothesen lassen sich dadurch erklären: Kleinere Abstände führen zu größeren Varianzen, die mehr Shots benötigen, um sich dem Erwartungswert anzunähern. Da in den Experimenten der Fehler aus der Varianz dem des Bias überwiegt, lieferten die größeren Abstände bessere Ergebnisse.

Hypothese 2: Vorwärts und rückwärts finite Differenzen liefern schlechtere Ergebnisse als zentrale finite Differenzen bei gleichem Abstand h.

Auch hier kann die Varianzanalyse eine Erklärung liefern. Dadurch, dass bei zentralen finiten Differenzen nicht durch h, sondern durch $2h$ geteilt wird, ist die Varianz nur ein Viertel der Varianz von vorwärts und rückwärts finiten Differenzen.

Da bei vorwärts und rückwärts finiten Differenzen der Messwert $L(\theta)$ für alle partiellen Ableitungen wiederverwendet werden kann, muss hier nur an $k + 1$ anstatt $2k$ Punkten gemessen werden. Dies wurde in den ersten Experimenten, im Gegensatz zu SPSA, nicht ausgeglichen. Wenn man bei beiden Verfahren insgesamt dieselbe Anzahl Shots verwenden will, kann man bei vorwärts und rückwärts finiten Differenzen also circa doppelt so viele Shots pro Messung verwenden. Damit ist die Varianz bei gleichem Abstand h dann nicht mehr viermal so groß wie bei zentralen finiten Differenzen, sondern nur noch circa doppelt so groß.

Hypothese 1: Das klassische und allgemeine Parameter-Shift Verfahren berechnen deutlich bessere Approximationen als alle numerischen Verfahren.

Hypothese 6: Bei dem allgemeinen und klassischen Parameter-Shift Verfahren gibt es leichte Unterschiede hinsichtlich der Anzahl Shots die benötigt werden, um den exakte Gradienten zu berechnen.

Diese beiden Hypothesen scheinen sich nur begrenzt zu bestätigen. Der Skalierungsfaktor $\frac{2r^2}{sin(2r\gamma)^2}$ des allgemeinen Parameter-Shift Verfahrens ist für die Experimente $\frac{1}{2sin(\gamma)^2}$. Für γ nahe 0 gilt $sin(\gamma) \approx \gamma$, also $\frac{1}{2sin(\gamma)^2} \approx \frac{1}{2\gamma^2}$. Der Skalierungsfaktor ist also identisch mit dem von zentralen finiten Differenzen für kleine Abstände. Es kann also nicht nur leichte, sondern erhebliche Unterschiede bei der benötigten Anzahl Shots zwischen allgemeinem und klassischem Parameter-Shift Verfahren geben. Der entscheidende Punkt ist hier aber, dass der Abstand im allgemeinen Parameter Shift beliebig gewählt werden kann und trotzdem keinen Bias hat. Man kann also den Abstand so wählen, dass der Nenner des Skalierungsfaktors maximiert wird und so die Varianz minimieren. Interessanterweise ist der Abstand $\gamma = \frac{\pi}{4r}$ des klassischen Parameter-Shift Verfahrens der bestmögliche Skalierungsfaktor. Für große Abstände scheint Hypothese 1 also weiterhin zu gelten.

Als Letztes bleibt noch die Hypothese zu SPSA:

Hypothese 3: SPSA berechnet bessere Approximationen als finite Differenzen.

Diese wird nun im nächsten Abschnitt detailliert untersucht.

6.4.3 SPSA

In SPSA ist eine mögliche Approximation g_i des i-ten Elements des Gradienten gegeben durch Gleichung (5.11) aus Abschnitt 5.3.2. Je nach Abstandsvektor Δ unterscheidet sich der Erwartungswert und die Varianz von g_i, da an unterschiedlichen Parameterpunkten gemessen wird. Wie viele verschiedene Abstandsvektoren es gibt, skaliert exponentiell mit der Anzahl an Parametern k. Für

jeden Parameter ist der Abstand entweder $+h$ oder $-h$. Es gibt also $K = 2^k$ mögliche Vektoren. Anschaulich lassen sich die aus dem jeweiligen Abstandsvektor Δ resultierenden Samplestellen als zwei gegenüberliegende Ecken eines N-dimensionalen Würfels mit Seitenlänge $2h$ verstehen[2].

Im Folgenden wird Δ als diskrete Zufallsvariable betrachtet mit einzelnen Δ_j als mögliche Abstandsvektoren auf die Δ abbildet. Aufgrund der gewählten Verteilung sind alle Δ_j gleich wahrscheinlich mit $p(\Delta_j) = \frac{1}{K}$. Es muss unterschieden werden zwischen einer Approximation $g_i(\Delta_j)$, wenn bereits ein spezifischer Abstandsvektor Δ_j gewählt wurde, und g_i, der allgemeinen Betrachtung der Approximation, vor der Wahl von Δ. Analog zum Vorgehen im vorherigen Abschnitt lassen sich für eine Approximation $g_i(\Delta_j)$ der Erwartungswert $\mu_i(\Delta_j)$ und die Varianz $\sigma_i^2(\Delta_j)$ bestimmen, siehe dazu Gleichung (6.19), Gleichung (6.20).

$$\mu_i(\Delta_j) = \mathbb{E}\left[\frac{L(\theta + h\Delta_j) - L(\theta - h\Delta_j)}{2h\Delta_{j\,i}}\right] = \frac{\mathbb{E}[L(\theta + h\Delta_j)] - \mathbb{E}[L(\theta - h\Delta_j)]}{2h\Delta_{j\,i}} \tag{6.19}$$

$$\sigma_i{}^2(\Delta_j) = Var\left(\frac{L(\theta + h\Delta_j) - L(\theta - h\Delta_j)}{2h\Delta_{j\,i}}\right) = \frac{Var(L(\theta + h\Delta_j)) + Var(L(\theta - h\Delta_j))}{4h^2} \tag{6.20}$$

Die Frage ist jedoch nicht, wie die Varianzen solcher einzelnen Approximationen aussehen, schließlich könnte man zufällig einen sehr guten oder sehr schlechten Abstandsvektor gewählt haben, sondern wie sich die Varianz der allgemeinen Approximation g_i zusammensetzt, die alle möglichen Abstandsvektoren berücksichtigt. Mithilfe des Satzes über die totale Wahrscheinlichkeit lässt sich der allgemeine Erwartungswert μ_i von g_i ausrechnen (Gleichung (6.21)).

$$\mu_i = \mathbb{E}_\Delta[\mathbb{E}[g_i|\Delta]] = \sum_{j=0}^{K} p(\Delta_j)\mathbb{E}[g_i|\Delta = \Delta_j] = \sum_{j=0}^{K} \frac{\mu_i(\Delta_j)}{K} \tag{6.21}$$

SPSA hat zwei Arten von Varianzen. Die erste Art ist die eben besprochene Varianz $\sigma_i{}^2(\Delta_j)$ einer Gradientenapproximation für ein spezifisches Δ_j. Diese Varianz ergibt sich aus den Varianzen der Messungen an $L(\theta \pm h\Delta_j)$. Aber auch die Erwartungswerte $\mu_i(\Delta_j)$ der Approximation für ein spezifisches Δ_j schwanken um den gesamten Erwartungswert μ_i. Diese Varianz σ_Δ^2 wird in Gleichung (6.22) beschrieben. Sie sagt aus wie stark die Approximation aufgrund der Wahl von Δ schwanken kann, wenn alle Messungen perfekt wären.

$$\sigma_\Delta^2 = \mathbb{E}_\Delta[(\mathbb{E}[g_i|\Delta] - \mu_i)^2] = \sum_{j=0}^{K} p(\Delta_j)(\mu_i(\Delta_j) - \mu_i)^2 = \frac{1}{K}\sum_{j=0}^{K}(\mu_i(\Delta_j) - \mu_i)^2 \tag{6.22}$$

[2] Die Samplestellen von zentralen finiten Differenzen entsprechen den Mittelpunkten gegenüberliegender Flächen dieses Würfels.

Es bleibt zu klären, wie sich diese beiden Varianzen zur allgemeinen Varianz $Var(g_{iSPSA}) = \sigma_i^2$ zusammensetzen. Als Ausgangspunkt wird diese allgemeine Varianz σ_i^2 ausformuliert:

$$\sigma_i^2 = Var(g_i) = \mathbb{E}_\Delta[(g_i - \mu_i)^2] = \mathbb{E}[g_i^2] - \mu_i^2 \tag{6.23}$$

Das zweite Moment $\mathbb{E}[g_i^2]$ kann weiter aufgeschlüsselt werden.

$$\mathbb{E}[g_i^2] = \mathbb{E}_\Delta[\mathbb{E}[g_i^2|\Delta]] = \sum_{j=0}^{K} p(\Delta_j)\mathbb{E}[g_i^2|\Delta = \Delta_j] = \frac{1}{K}\sum_{j=0}^{K}({\sigma_i}^2(\Delta_j) + {\mu_i}^2(\Delta_j)) \tag{6.24}$$

Dabei wurde in Gleichung (6.24) ausgenutzt, dass ${\sigma_i}^2(\Delta_j) = \mathbb{E}[g_i^2|\Delta = \Delta_j] - {\mu_i}^2(\Delta_j)$ gilt.

Sei nun $\sigma_M^2 = \frac{1}{K}\sum_{j=0}^{K}{\sigma_i}^2(\Delta_j)$ die durchschnittliche Varianz, die durch Messungen induziert wird. Durch Umformulierung von σ_Δ^2 als $(\frac{1}{K}\sum_{i=0}^{K}{\mu_i}^2(\Delta_j)) - \mu_i^2$ erhält man schlussendlich den Zusammenhang der beiden Varianzen:

$$\sigma_i^2 = \frac{1}{K}\left(\sum_{j=0}^{K}({\sigma_i}^2(\Delta_j) + {\mu_i}^2(\Delta_j))\right) - \mu_i^2 = \left(\frac{1}{K}\sum_{j=0}^{K}{\sigma_i}^2(\Delta_j)\right) + \left((\frac{1}{K}\sum_{j=0}^{K}{\mu_i}^2(\Delta_j)) - \mu_i^2\right) = \sigma_M^2 + \sigma_\Delta^2 \tag{6.25}$$

Dies lässt sich als ein Spezialfall des Gesetzes der totalen Varianz verstehen. Die gesamte Varianz ergibt sich als Summe der messungsunabhängigen Varianz σ_Δ^2 und der durchschnittlichen messungsabhängigen Varianz σ_M^2.

Über σ_Δ^2 können aus den bisherigen Experimenten und Analysen noch keine Schlüsse gezogen werden. σ_M^2 lässt sich jedoch auch unter der vorherigen Annahme, dass die Varianz einer Messung konstant mit σ^2 angenommen werden kann, betrachten. Die Annahme wurde nur für Shifts in einem Parameter formuliert. Hier werden jedoch alle Parameter gleichzeitig verschoben. In Experimenten mit $h = 0.01$ für verschiedene Δ galt aber auch $Var(L(\theta + \Delta h)) + Var(L(\theta - \Delta h)) \simeq 2Var(L(\theta))$. Basierend auf diesen empirischen Resultaten wird nun auch von $Var(L(\theta \pm \Delta h)) = \sigma^2$ ausgegangen:

$$\sigma_M^2 = \frac{1}{K}\sum_{j=0}^{K}{\sigma_i}^2(\Delta_j) = \frac{1}{K}\sum_{j=0}^{K}\frac{2\sigma^2}{4h^2} \simeq \frac{1}{K}\frac{K\sigma^2}{2h^2} = \frac{\sigma^2}{2h^2} \tag{6.26}$$

σ_M^2 ist unter der Annahme somit identisch zur Varianz von zentralen finiten Differenzen. Obwohl für SPSA noch die Varianz σ_Δ^2 hinzukommt, hat es aber trotzdem in den Experimenten bessere Approximationen berechnet als zentrale finite Differenzen. Um dies zu klären, muss noch betrachtet werden, wie sich die Anzahl Shots auf die Varianz von SPSA auswirkt.

	SPSA	Finite Differenzen			Parameter Shift	
Varianten		**Vorwärts**	**Rückwärts**	**Zentral**	**Einfach**	**Allgemein**
Samplestellen	2	$k+1$	$k+1$	$2k$	$2k$	$2k$
Varianz	$\sigma_\Delta^2 + \frac{\sigma^2}{2h^2}$	$\frac{2\sigma^2}{h^2}$	$\frac{2\sigma^2}{h^2}$	$\frac{\sigma^2}{2h^2}$	$2\,\sigma^2 r^2$	$\frac{2\sigma^2 r^2}{sin(2r\gamma)^2}$

Tabelle 6.1

Bei den bisherigen Verfahren macht es keinen Unterschied, ob man n mal mit einem Shot approximiert und über die Ergebnisse den Mittelwert bildet, oder einmal mit n Shots approximiert. Bei SPSA spielt dies jedoch eine Rolle. Wenn man nur einen spezifischen Abstandsvektor Δ_j wählt und bei den Messungen n Shots verwendet, entspricht dies einer Mittelwertschätzung von $\mu_i(\Delta_j)$ mit Messvarianz $\frac{\sigma_M^2}{n}$. Die Varianz σ_Δ^2 bleibt aber unverändert. Wenn man stattdessen den Mittelwert über n Approximationen mit unterschiedlichen Abstandsvektoren bildet, bei denen nur ein Shot pro Messung verwendet wurde, wird die gesamte Varianz auf $\frac{\sigma_i^2}{n} = \frac{\sigma_M^2}{n} + \frac{\sigma_\Delta^2}{n}$ reduziert.

Allgemein für einen Durchschnitt aus p Ziehungen von Δ, wobei Messungen mit jeweils n Shots ausgeführt werden, ergibt sich die Varianz:

$$\sigma_i^2(|\Delta| = p, \text{Shots} = n) = \frac{\frac{\sigma_M^2}{n} + \sigma_\Delta^2}{p} = \frac{\sigma_M^2}{np} + \frac{\sigma_\Delta^2}{p} \tag{6.27}$$

Daraus resultiert, dass möglichst mit nur geringen Shotzahlen über möglichst viele verschiedene Abstandsvektoren gemittelt werden sollte, da so beide Arten von Varianzen verringert werden können. Jedoch kann dadurch die Laufzeit erhöht werden. So war für den Versuchsaufbau in Qiskit das Ausführen von einem Experiment mit mehreren Shots deutlich schneller als das Ausführen mehrerer Experimente mit einem Shot.

Der entscheidende Punkt, warum SPSA bei den Experimenten bessere Ergebnisse lieferte, besteht darin, dass mit nur *zwei* Messungen die Varianz in *allen* Dimensionen verringert wird. Angenommen SPSA und zentrale finite Differenzen verwenden beide $2np$ viele Shots, wobei $p = k$ auf die Anzahl der Parameter gesetzt wird. Finite Differenzen verwendet für jeden der k Parameter $2n$ Shots und erreicht damit eine Varianz von $\frac{\sigma_M^2}{n}$ für jeden Parameter. SPSA vermittelt über k Ziehungen von Δ und verwendet bei den Messungen ebenfalls $2n$ Shots. Die Varianz in jedem Parameter wird dadurch auf $\frac{\sigma_M^2}{nk} + \frac{\sigma_\Delta^2}{k}$ gesenkt. Solange $\sigma_\Delta^2 < \frac{k-1}{n}\sigma_M^2$ gilt, hat SPSA in diesem Fall eine geringere Varianz als zentrale finite Differenzen.

6.5 Validierung der Analyse

Zur Überprüfung der Analyse wurden weitere Experimente durchgeführt. Für alle Verfahren bis auf SPSA lassen sich Bias und Varianz leicht bestimmen: Mithilfe von Opflow können die exakten Approximationen g_i, sowie mit zwei zusätzlichen Messungen pro Samplestelle (einmal mit Observable O und einmal mit O^2) auch die exakten Varianzen $Var(g_i)$ bestimmt werden.

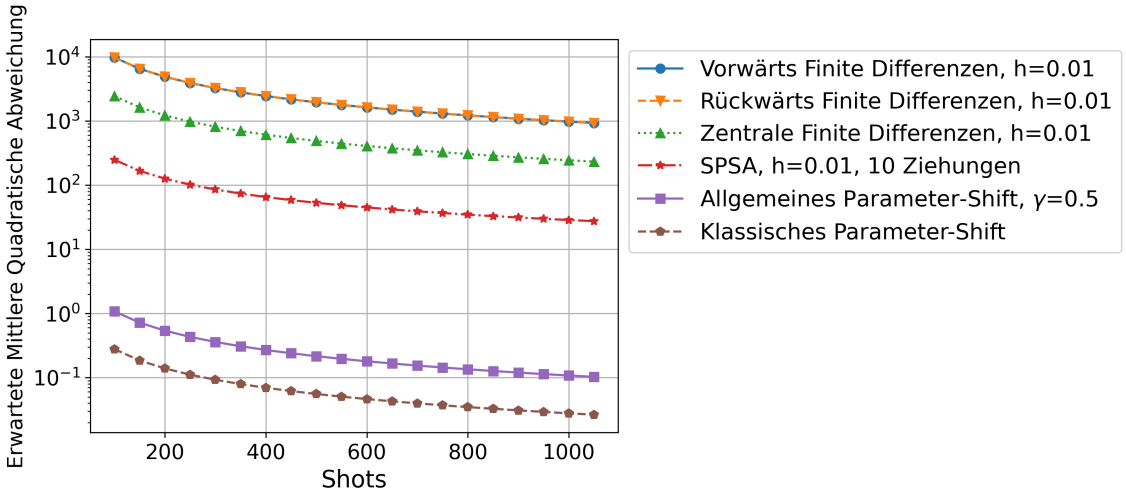

Abbildung 6.10: Vergleich der exakten Varianzen für die Verfahren aus den ersten Experimenten

Bei SPSA lassen sich die einzelnen $g_i(\Delta)$ und deren Varianz ebenfalls so berechnen. Das reicht aber noch nicht, um σ_M^2 und σ_Δ^2 zu bestimmen. Für diese Varianzen müssen Erwartungswerte bezüglich Δ berechnet werden. Es müssen also Summen über alle möglichen Δ Vektoren gebildet werden.

Wie bereits in Abschnitt 6.4.3 erwähnt, gibt es insgesamt 2^k mögliche Δ Vektoren bei k Parametern im Schaltkreis. Der Vektor $-\Delta_j$ liefert jedoch die gleiche Approximation wie Δ_j. Damit können die Hälfte der Abstandsvektoren ignoriert werden und es bleiben nur noch 2^{k-1} mögliche Vektoren übrig. Bei dem verwendeten Ansatz der Experimente ($k = 24$) führte dies bereits zu einer inakzeptablen Laufzeit um einmal $\mathbb{E}_\Delta[g_i(\Delta)]$, sowie σ_M^2 und σ_Δ^2 zu berechnen. Zur Beschleunigung wurden die exakten Berechnungen durch folgende Approximationen ersetzt: Es werden zufällig Δ Vektoren gewählt und iterativ der Stichprobenmittelwert ermittelt. Während dieses Prozesses wird die Änderungsrate über die letzten 20 Δ Vektoren verfolgt. Fällt dieser unter einen Grenzwert ϵ, wird der Stichprobenmittelwert als Approximation akzeptiert.

Die Grenzwerte ϵ wurden dabei für die verschiedenen Approximationen nicht gleich groß gewählt. Da bei der Approximation von σ_Δ^2 der bereits approximierte Erwartungswert $\mathbb{E}_\Delta[g_i(\Delta)]$ genutzt wird, wurde hier ein größeres ϵ gewählt, um einer längeren Laufzeit aufgrund des Fehlers aus der ersten Approximation vorzubeugen.

Aus Abbildung 6.10 lässt sich herauslesen, dass die erwarteten Fehler unter Berücksichtigung der tatsächlichen Varianzen die Vermutungen aus der Analyse bestätigen. Es scheint also, dass die tatsächlichen Varianzen an den verschiedenen Samplestellen die Annahme 1 aus der Analyse erfüllen. Für SPSA zeigte sich, dass in den ersten Experimenten $\sigma_M^2 \gg \sigma_\Delta^2$ galt und σ_M^2 circa der Varianz $Var(g_{i(ZFD)})$ von zentralen finiten Differenzen entsprach. Da bei allen Verfahren für einzelne Messungen gleich viele Shots verwendet wurden und bei SPSA über zehn Approximationen der Durchschnitt gebildet wurde, ist hier für n Shots $Var(g_{i(SPSA)}) = \frac{\sigma_M^2}{10n} + \frac{\sigma_\Delta^2}{10} \approx \frac{\sigma^2(g_{i(ZFD)})}{10}$. Wenn man die gesamte Anzahl an Shots ausgleicht, ist die Varianz von SPSA circa um einen Faktor k geringer als die Varianz von zentralen finiten Differenzen. Als abschließende Frage bleibt offen, welche Hyperparameter das Verhältnis von σ_M^2 und σ_Δ^2 beeinträchtigen können, aber aus den Experimenten in dieser Arbeit scheint SPSA gerade für zukünftig größere Schaltkreise mit mehr Parametern das beste der besprochenen numerischen Verfahren zu sein.

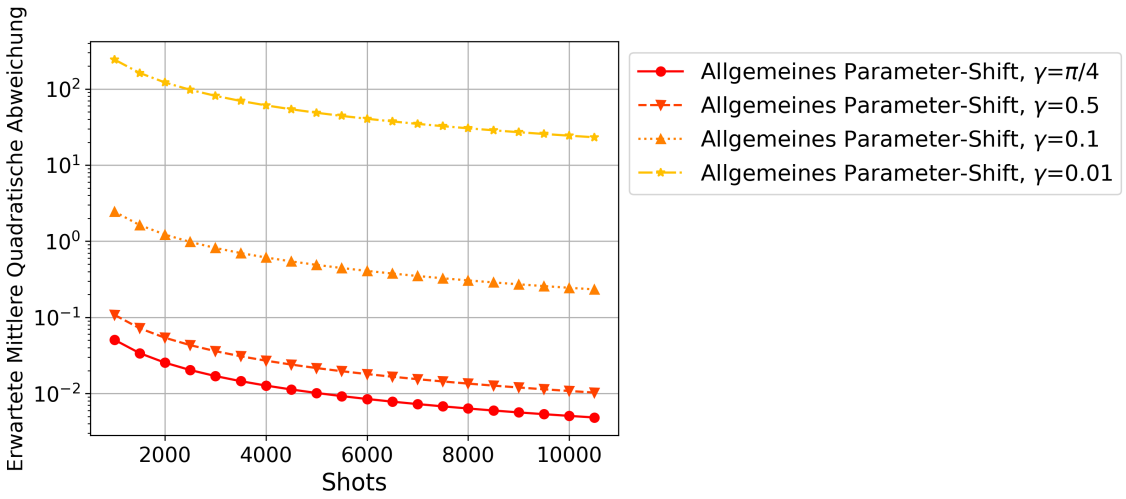

Abbildung 6.11: Vergleich der exakten Varianzen für das allgemeine Parameter-Shift Verfahren mit verschiedenen Shift-Werten an dem Parameterpunkt der ersten Experimente

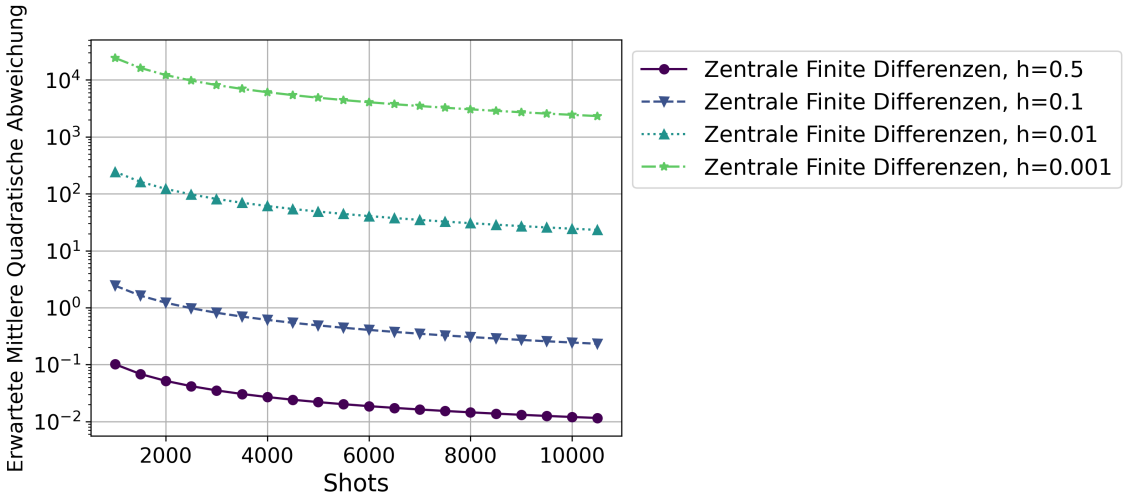

Abbildung 6.12: Vergleich der exakten Varianzen für zentrale finite Differenzen mit verschiedenen Abständen h an dem Parameterpunkt der ersten Experimente

Wie in 6.11 zusehen, bestätigt sich auch die Analyse für das allgemeine Parameter-Shift Verfahren. Je kleiner der Abstand γ gewählt wird, desto größer wird die Varianz.

Auch für die verschiedenen finiten Differenzen Verfahren bestätigen sich die Ideen der Analyse, wie in Abbildung 6.12 zu sehen. Für die Varianz von vorwärts und rückwärts finiten Differenzen zu zentralen finiten Differenzen ergab sich, wie aus der Analyse erwartet, bei gleichem Abstand ein Unterschied von circa dem Faktor 4.

Wie ursprünglich in Abschnitt 5.3.1 erwähnt, lassen sich auch finite Differenzen mit mehr als zwei Samplestellen konstruieren. Dies führte aber zu keinen guten Ergebnissen (siehe Abbildung 6.5). Auch wenn mehr Samplestellen den Bias reduzieren sollten, fügt jede Samplestelle zusätzliche Varianz hinzu. Finite Differenzen mit mehr Samplestellen scheinen also nicht sinnvoll.

Zuletzt noch eine Bemerkung bezüglich der in den Experimenten außer Acht gelassenen analytischen Verfahren, die sich auf beliebige Gatter anwenden lassen. Für eine Einschätzung der Performanz der Gradientenberechnung per Gatterzerlegung lassen sich die hier vorgestellten Ergebnisse auch nutzen. Die Gradienten bezüglich einzelner Gatter der Zerlegung sollten per Parameter Shift ähnlich gute Ergebnisse liefern wie in den Experimenten in dieser Arbeit. Es stellt sich die Frage, wie kompliziert die benutzen parametrisierten Gatter zerlegt werden müssen. Als Schranke ist bekannt, dass sich ein n-Qubit Operator als Komposition von $O(n^2 \cdot 4^n)$ 1-Qbit Operatoren und CNOTs schreiben lässt [NC04]. Diese Schranke konnte auf $O(4^n)$ [VMS04] [LRY13] reduziert werden. Wenn eine beliebige Zerlegung mithilfe des kanonischen Gatters ähnlich skaliert, kann die Anzahl benötigter Parameter-Shift Verfahren exponentiell wachsen. Hierzu sei noch gesagt, dass das Problem der Dekomposition aufgrund der begrenzten Anzahl an Basis-Gattern von heutigen Quantencomputern natürlich auch schon bei der allgemeinen Ausführung des Schaltkreises auftritt. Dies hat aber keine Auswirkung auf die Anzahl an benötigten Messungen, die Dekomposition des Gatterzerlegungsverfahrens jedoch schon. Bei einer exponentiell wachsenden Anzahl an Messungen per Gatterzerlegung könnten die numerischen Verfahren wieder zu besseren Ergebnissen führen. Vielleicht kann durch ein ähnliches Verfahren, das eine Approximation der Gatter [DN06] verwendet, dieses Problem bei komplizierteren Gattern umgangen werden.

Sowohl auf das Linearkombinationsverfahren als auch auf das stochastische Parameter Shift Verfahren [BC21] lassen sich die Ergebnisse dieser Arbeit nicht übertragen. Auch bei der Linearkombination ist unklar, wie stark das Verfahren mit größeren Operatoren skaliert. Schließlich gibt es dort eine exponentiell wachsende Anzahl an möglichen Basiselementen des Generators, die potenziell von Parametern abhängen könnten. Zusätzlich modifizieren beide Verfahren den Schaltkreis und führen weitere Probabilistiken ein. Es bleibt abzuwarten, welche Gatter in zukünftigen Ansätzen eine Rolle spielen werden.

7 Zusammenfassung und Ausblick

In dieser Arbeit wurden verschiedene Verfahren zur Gradientenberechnung auf Quantencomputern untersucht. Es wurde eine Übersicht gegeben, wie Gradienten bezüglich parametrisierter Observablen und Schaltkreisen berechnet werden können, wobei der Fokus der weiteren Arbeit auf Gradientenberechnung bezüglich parametrisierten Schaltkreisen lag.

Als numerische Verfahren wurden finite Differenzen sowie das Gradientenapproximationsverfahren des SPSA Algorithmus vorgestellt. Als quantenspezifische analytische Verfahren wurde das Parameter-Shift Verfahren vorgestellt, sowie das allgemeine Parameter-Shift Verfahren. Diese beiden Verfahren können nur auf parametrisierten Gattern eingesetzt werden, wenn deren Generatoren bestimmte Bedingungen erfüllen. Zusätzlich wurde eine Übersicht über analytische Verfahren gegeben, die auf beliebigen Gattern angewandt werden können.

In Experimenten haben die numerischen Verfahren schlechtere Ergebnisse als ihre analytischen Gegenstücke produziert. Aber auch innerhalb der numerischen Verfahren gab es Unterschiede zwischen vorwärts, rückwärts und zentralen finiten Differenzen. Außerdem hat SPSA überraschenderweise bessere Ergebnisse als alle finiten Differenzen Verfahren geliefert.

Diese Beobachtungen wurden als Hypothesen formuliert und weiter analysiert. Zuerst fiel auf, dass der größte Fehler (bei NISQ üblichen Shotzahlen) durch Varianz entsteht. Gerade für die numerischen Verfahren, aber je nach Wahl des Shifts auch beim allgemeinen Parameter-Shift Verfahren, werden so erheblich mehr Shots für gute Approximationen benötigt. Direkt aus den Berechnungsformeln folgt, dass die inhärente Varianz der Messungen bei finiten Differenzen mit $O(h^{-2})$, sowie bei dem allgemeinen Parameter-Shift Verfahren mit $O(\sin(h)^{-2})$ skaliert wird. Für finite Differenzen ergibt sich daraus ein Dilemma, da ein möglichst kleiner Abstand den Bias verringert, was im Widerspruch zu einem möglichst großen Abstand zur Varianzminimierung steht. Im Gegenzug führt beim allgemeinen Parameter-Shift Verfahren jeder Abstand zum exakten Ergebnis, sodass der Abstand hier so groß wie möglich gewählt werden kann. Diese Behauptungen stützen sich auf eine Annahme aus [MBK21], wonach die Varianzen von Messungen an einem Parameterpunkt ähnlich zu denen an Parameterpunkten nach beliebigen Shifts sind. Die Experimente dieser Arbeit stützen diese Annahme. Eine mögliche Erklärung für dieses Verhalten könnte in dem Barren Plateau Phänomen [MBS+18] liegen.

Im Detail wurde die Zusammensetzung der Varianz von SPSA besprochen. Auch wenn die Varianz aufgrund der Messungen dort ebenfalls mit $O(h^{-2})$ skaliert, lässt sie sich in allen Parametern mit nur zwei Messungen generieren. Dadurch können im Verhältnis mehr Shots auf diese Messungen angewandt werden, was die Varianz in allen Parametern reduziert. Diese Unabhängigkeit von der Parameteranzahl bietet gerade für zukünftige, größere, variationelle Quantenalgorithmen Potenzial.

Ausblick

Mit den Ergebnissen dieser Arbeit als Grundlage steht noch ein Vergleich mit den analytischen, auf beliebige Gatter anwendbaren Verfahren aus. Dabei interessiert die Güte der Ergebnisse im Verhältnis zum Aufwand der Verfahren in Bezug zu den numerischen Verfahren aus dieser Arbeit. Bei dem Linearkombinationsverfahren und stochastischem Parameter Shift wird wahrscheinlich, aufgrund der doppelten Probablilistik, eine Analyse ähnlich der hier besprochenen SPSA Analyse möglich sein. Für das Linearkombinationsverfahren und das Gatterzerlegungsverfahren bleibt ebenfalls die Frage, wie die Zerlegung in Gatter bzw. Terme effizient für beliebige Gatter automatisiert werden kann. Auch das in Pennylane benutzte allgemeine Parameter Shift Verfahren [WIWL21] sollte für Vergleiche hinzugezogen werden.

Ein weiteres offenes Thema besteht darin, die Ergebnisse dieser Arbeit mit dem Barren Plateau Phänomen in Kontext zu setzen. Zum einen gehört dazu eine Analyse der Annahme aus [MBK21], für die wahrscheinlich die Werkzeuge verwendet werden können, mit denen in der Barren Plateau Arbeit [MBS+18] argumentiert wurde. Interessant ist dabei, inwieweit die beiden Phänomene zusammenhängen: Gibt es Barren Plateaus, in denen die Annahme nicht gilt? Gibt es Regionen, in denen die Annahme gilt, die aber keine Barren Plateaus sind?

Ebenfalls stellt sich die Frage inwiefern Barren Plateaus vom gewählten Ansatz abhängen. Vielleicht sind nicht universelle Ansätze, wie der in dieser Arbeit verwendete EfficientSU2 Ansatz, der richtige Weg, sondern problemspezifische Ansätze. Beispielsweise wurde in der ersten VQE Veröffentlichung [PMS+14] der Unitary Coupled Cluster Ansatz [TB06][OBK+16] aus der Quantenchemie verwendet. Kombiniert mit einem problemspezifischen Initialzustand könnte dies zu einer lokaleren Optimierung führen, die nicht von Barren Plateaus betroffen ist. Das Warm Starting Verfahren [EMW21] [TBB+22] zur Lösung klassischer Optimierungsprobleme, bei dem der Initialzustand des Quantenalgorithmus aus einer approximierten Lösung durch klassische Optimierung erzeugt wird, ist ein Beispiel für ein solches Vorgehen.

Eine weitere Möglichkeit, wie man das Barren Plateau Phänomen umgehen kann, besteht darin Barren Plateaus buchstäblich zu umgehen [SMM+22]. Durch Entropiemessungen kann festgestellt werden, ob man ein schwaches Barren Plateau betritt. Ist dies während eines Optimierungsschritts der Fall, kann man einen Schritt zurückgehen und dem Plateau ausweichen. Auch beim Initialzustand kann dieses Verfahren genutzt werden, um einen Start in einem Plateau zu verhindern.

Eine hier nicht besprochene Idee ist eine Übertragung des Konzepts des Natural Gradients von tiefen neuronalen Netzen [Ama98] auf den Quantencomputer: der Quantum Natural Gradient [SIKC20] [KB19]. Dabei wird die Inverse der Quantum Fisher Information Matrix [PG11] als metrischer Tensor zur Skalierung des Gradienten verwendet, was einer approximierten Optimierung zweiter Ordnung entspricht. Zum einen gilt es die potenziell besseren und schneller konvergierenden Ergebnisse mit dem zusätzlichen Aufwand in Verhältnis zu setzen und zu klären, ob auch dort ähnliche Varianzprobleme auftreten können. Außerdem ist auch hier die Frage, ob der Quantum Natural Gradient ein hilfreiches Werkzeug sein könnte, um ein Steckenbleiben in Barren Plateaus zu verhindern. Wenn der Gradient im Plateau nur sehr klein wird, kann eine Skalierung dafür sorgen, dass das Plateau schneller wieder verlassen wird. Falls im schlimmsten Fall der Gradient jedoch exakt der Nullvektor ist, hilft auch keine Skalierung, um dem Plateau zu entkommen.

Literaturverzeichnis

[AAA+21] M. S. ANIS et al. *Qiskit: An Open-source Framework for Quantum Computing*. 2021. DOI: 10.5281/zenodo.2573505 (zitiert auf S. 35).

[Ama98] S.-i. Amari. „Natural Gradient Works Efficiently in Learning“. In: *Neural Computation* 10.2 (Feb. 1998), S. 251–276. DOI: 10.1162/089976698300017746 (zitiert auf S. 56).

[BBC+95] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, H. Weinfurter. „Elementary gates for quantum computation“. In: *Phys. Rev. A* 52 (5 Nov. 1995), S. 3457–3467. DOI: 10.1103/PhysRevA.52.3457 (zitiert auf S. 14).

[BC21] L. Banchi, G. E. Crooks. „Measuring Analytic Gradients of General Quantum Evolution with the Stochastic Parameter Shift Rule“. In: *Quantum* 5 (2021), S. 386. DOI: 10.22331/q-2021-01-25-386 (zitiert auf S. 33, 53).

[Bel57] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957. ISBN: 9780691146683 (zitiert auf S. 45).

[BIS+18] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M. S. Alam, S. Ahmed, J. M. Arrazola, C. Blank, A. Delgado, S. Jahangiri et al. „Pennylane: Automatic differentiation of hybrid quantum-classical computations“. In: *arXiv preprint arXiv:1811.04968* (2018). DOI: 10.48550/ARXIV.1811.04968 (zitiert auf S. 25, 35).

[Bis06] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006. ISBN: 0387310738. DOI: 10.5555/1162264 (zitiert auf S. 17).

[Bot10] L. Bottou. „Large-scale machine learning with stochastic gradient descent“. In: *Proceedings of COMPSTAT'2010*. Springer, 2010, S. 177–186. ISBN: 978-3-7908-2604-3. DOI: 10.1007/978-3-7908-2604-3_16 (zitiert auf S. 9).

[BPRS17] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, J. M. Siskind. „Automatic differentiation in machine learning: a survey“. In: *Journal of Marchine Learning Research* 18.1 (2017), S. 5595–5637. DOI: 10.5555/3122009.3242010 (zitiert auf S. 25).

[CAB+21] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio et al. „Variational quantum algorithms“. In: *Nature Reviews Physics* 3.9 (2021), S. 625–644. DOI: 10.1038/s42254-021-00348-9 (zitiert auf S. 9).

[CGA17] Y. Cao, G. G. Guerreschi, A. Aspuru-Guzik. „Quantum Neuron: an elementary building block for machine learning on quantum computers“. In: (2017). DOI: 10.48550/ARXIV.1711.11240 (zitiert auf S. 21).

[Col12] L. Collatz. *The numerical treatment of differential equations*. Springer-Verlag, 2012. ISBN: 978-3-662-05456-7. DOI: 10.1007/978-3-662-05500-7 (zitiert auf S. 27).

[Cro19] G. E. Crooks. „Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition“. In: *arXiv preprint arXiv:1905.13311* (2019). DOI: 10.48550/ARXIV.1905.13311 (zitiert auf S. 31).

[CSAC20] M. Cerezo, K. Sharma, A. Arrasmith, P. J. Coles. „Variational quantum state eigensolver“. In: *arXiv preprint arXiv:2004.01372* (2020). DOI: 10.48550/ARXIV.2004.01372 (zitiert auf S. 9, 21, 36).

[CW12] A. M. Childs, N. Wiebe. „Hamiltonian simulation using linear combinations of unitary operations“. In: *arXiv preprint arXiv:1202.5822* (2012) (zitiert auf S. 33).

[De 17] R. De Wolf. „The potential impact of quantum computers on society“. In: *Ethics and Information Technology* 19.4 (2017), S. 271–276. DOI: 10.1007/s10676-017-9439-z (zitiert auf S. 9).

[DFO20] M. Deisenroth, A. Faisal, C. Ong. *Mathematics for Machine Learning*. Feb. 2020. ISBN: 9781108470049. DOI: 10.1017/9781108679930 (zitiert auf S. 17).

[DHLT20] Y. Du, M.-H. Hsieh, T. Liu, D. Tao. „Expressive power of parametrized quantum circuits“. In: *Phys. Rev. Research* 2 (3 Juli 2020). DOI: 10.1103/PhysRevResearch.2.033125 (zitiert auf S. 21).

[DN06] C. M. Dawson, M. A. Nielsen. „The Solovay-Kitaev Algorithm“. In: *Quantum Info. Comput.* 6.1 (Jan. 2006), S. 81–95. DOI: 10.5555/2011679.2011685 (zitiert auf S. 53).

[Dom12] P. Domingos. „A few useful things to know about machine learning“. In: *Communications of the ACM* 55.10 (2012), S. 78–87. DOI: 10.1145/2347736.2347755 (zitiert auf S. 17).

[DPS03] G. M. D'Ariano, M. G. Paris, M. F. Sacchi. „Quantum tomography“. In: *Advances in Imaging and Electron Physics* 128 (2003), S. 206–309. DOI: 10.1016/S1076-5670(03)80065-4 (zitiert auf S. 25).

[EMW21] D. J. Egger, J. Mareček, S. Woerner. „Warm-starting quantum optimization“. In: *Quantum* 5 (Juni 2021), S. 479. DOI: 10.22331/q-2021-06-17-479 (zitiert auf S. 56).

[FGG14] E. Farhi, J. Goldstone, S. Gutmann. „A quantum approximate optimization algorithm“. In: *arXiv preprint arXiv:1411.4028* (2014). DOI: 10.48550/ARXIV.1411.4028 (zitiert auf S. 21).

[FHJ+21] L. Funcke, T. Hartung, K. Jansen, S. Kühn, P. Stornati. „Dimensional Expressivity Analysis of Parametric Quantum Circuits“. In: *Quantum* 5 (März 2021), S. 422. DOI: 10.22331/q-2021-03-29-422 (zitiert auf S. 36).

[FHY19] H. Fujiyoshi, T. Hirakawa, T. Yamashita. „Deep learning-based image recognition for autonomous driving“. In: *IATSS research* 43.4 (2019), S. 244–252. DOI: https://doi.org/10.1016/j.iatssr.2019.11.008 (zitiert auf S. 9).

[FLNH13] R. Fakoor, F. Ladhak, A. Nazi, M. Huber. „Using deep learning to enhance cancer diagnosis and classification“. In: *Proceedings of the international conference on machine learning*. Bd. 28. ACM, New York, USA. 2013, S. 3937–3949 (zitiert auf S. 9, 17).

[GRTZ02] N. Gisin, G. Ribordy, W. Tittel, H. Zbinden. „Quantum cryptography“. In: *Reviews of Modern Physics* 74.1 (März 2002), S. 145–195. DOI: 10.1103/RevModPhys.74.145 (zitiert auf S. 9, 11).

[HCT+19] V. Havlíček, A. Córcoles, K. Temme, A. Harrow, A. Kandala, J. Chow, J. Gambetta. „Supervised learning with quantum-enhanced feature spaces“. In: *Nature* 567 (März 2019), S. 209–212. DOI: 10.1038/s41586-019-0980-2 (zitiert auf S. 21).

[HFQE22] T. Hubregtsen, W. Frederik, S. Qasim, J. Eisert. „Single-component gradient rules for variational quantum algorithms“. In: *Quantum Science and Technology* 7 (Apr. 2022). DOI: 10.1088/2058-9565/ac6824 (zitiert auf S. 31, 43).

[HLL+22] C. He, J. Li, W. Liu, J. Peng, Z. J. Wang. „A Low-Complexity Quantum Principal Component Analysis Algorithm“. In: *IEEE Transactions on Quantum Engineering* 3 (2022), S. 1–13. DOI: 10.1109/TQE.2021.3140152 (zitiert auf S. 23).

[JEP+21] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko et al. „Highly accurate protein structure prediction with AlphaFold“. In: *Nature* 596.7873 (2021), S. 583–589. DOI: 10.1038/s41586-021-03819-2 (zitiert auf S. 9).

[KB14] D. P. Kingma, J. Ba. „Adam: A method for stochastic optimization“. In: *arXiv preprint arXiv:1412.6980* (2014). DOI: 10.48550/ARXIV.1412.6980 (zitiert auf S. 19).

[KB19] B. Koczor, S. C. Benjamin. „Quantum natural gradient generalised to non-unitary circuits“. In: *arXiv preprint arXiv:1912.08660* (2019). DOI: 10.48550/ARXIV.1912.08660 (zitiert auf S. 56).

[KMT+17] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. Chow, J. Gambetta. „Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets“. In: *Nature* 549 (Sep. 2017), S. 242–246. DOI: 10.1038/nature23879 (zitiert auf S. 9, 22, 36, 37).

[KSH12] A. Krizhevsky, I. Sutskever, G. E. Hinton. „ImageNet Classification with Deep Convolutional Neural Networks“. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. Curran Associates Inc., 2012, S. 1097–1105. DOI: 10.5555/2999134.2999257 (zitiert auf S. 9).

[LB20] F. Leymann, J. Barzen. „The bitter truth about gate-based quantum algorithms in the NISQ era“. In: *Quantum Science and Technology* 5 (Aug. 2020). DOI: 10.1088/2058-9565/abae7d (zitiert auf S. 15).

[LMR14] S. Lloyd, M. Mohseni, P. Rebentrost. „Quantum principal component analysis“. In: *Nature Physics* 10.9 (2014), S. 631–633. DOI: 10.1038/nphys3029 (zitiert auf S. 23).

[LRY13] C.-K. Li, R. Roberts, X. Yin. „Decomposition of unitary matrices and quantum gates“. In: *International Journal of Quantum Information* 11.01 (2013). DOI: 10.1142/S0219749913500159 (zitiert auf S. 53).

[LTO+19] R. LaRose, A. Tikku, É. O'Neel-Judy, L. Cincio, P. J. Coles. „Variational quantum state diagonalization“. In: *npj Quantum Information* 5.1 (2019), S. 1–10. DOI: 10.1038/s41534-019-0167-6 (zitiert auf S. 21).

[Mac16] D. Maclaurin. „Modeling, inference and optimization with composable differentiable procedures“. Diss. 2016. URL: http://nrs.harvard.edu/urn-3:HUL.InstRepos:33493599 (zitiert auf S. 25).

[MBI+20] A. Mari, T. Bromley, J. Izaac, M. Schuld, N. Killoran. „Transfer learning in hybrid classical-quantum neural networks“. In: *Quantum* 4 (Okt. 2020), S. 340. DOI: 10.22331/q-2020-10-09-340 (zitiert auf S. 21).

[MBK21] A. Mari, T. R. Bromley, N. Killoran. „Estimating the gradient and higher-order derivatives on quantum hardware“. In: *Phys. Rev. A* 103 (1 Jan. 2021). DOI: 10.1103/PhysRevA.103.012405 (zitiert auf S. 31, 43, 45, 55, 56).

[MBS+18] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, H. Neven. „Barren plateaus in quantum neural network training landscapes“. In: *Nature communications* 9.1 (2018). DOI: 10.1038/s41467-018-07090-4 (zitiert auf S. 22, 45, 55, 56).

[MM05] K. W. Morton, D. F. Mayers. *Numerical solution of partial differential equations: an introduction*. Cambridge university press, 2005. ISBN: 0521607930. DOI: 10.5555/1121701 (zitiert auf S. 27).

[MNKF18] K. Mitarai, M. Negoro, M. Kitagawa, K. Fujii. „Quantum circuit learning“. In: *Phys. Rev. A* 98 (3 Sep. 2018). DOI: 10.1103/PhysRevA.98.032309 (zitiert auf S. 30).

[NC04] M. Nielsen, I. Chuang. *Quantum Computation and Quantum Information*. Bd. 70. Jan. 2004. DOI: 10.1063/1.1428442 (zitiert auf S. 11, 53).

[NM65] J. A. Nelder, R. Mead. „A simplex method for function minimization“. In: *The computer journal* 7.4 (1965), S. 308–313. DOI: 10.1093/comjnl/7.4.308 (zitiert auf S. 9, 18).

[OBK+16] P. O'Malley, R. Babbush, I. Kivlichan, J. Romero, J. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding et al. „Scalable Quantum Simulation of Molecular Energies“. In: *Physical Review X* 6.3 (2016). DOI: 10.1103/PhysRevX.6.031007 (zitiert auf S. 22, 56).

[OTT19] T. O'Brien, B. Tarasinski, B. Terhal. „Quantum phase estimation of multiple eigenvalues for small-scale (noisy) experiments“. In: *New Journal of Physics* 21 (Feb. 2019). DOI: 10.1088/1367-2630/aafb8e (zitiert auf S. 23).

[PG11] D. Petz, C. Ghinea. „Introduction to quantum Fisher information“. In: *Quantum probability and related topics*. World Scientific, 2011, S. 261–281. DOI: 10.1142/9789814338745_0015 (zitiert auf S. 56).

[PMS+14] A. Peruzzo, J. Mcclean, P. Shadbolt, M. H. Yung, X. Zhou, P. Love, A. Aspuru-Guzik, J. O'Brien. „A variational eigenvalue solver on a photonic quantum processor“. In: *Nature communications* 5.1 (2014). DOI: 10.1038/ncomms5213 (zitiert auf S. 9, 21, 22, 37, 56).

[Pre18] J. Preskill. „Quantum Computing in the NISQ era and beyond“. In: *Quantum* 2 (Aug. 2018), S. 79. DOI: 10.22331/q-2018-08-06-79 (zitiert auf S. 15).

[RBB03] R. Raussendorf, D. E. Browne, H. J. Briegel. „Measurement-based quantum computation on cluster states“. In: *Phys. Rev. A* 68 (2 Aug. 2003). DOI: 10.1103/PhysRevA.68.022312 (zitiert auf S. 13).

[RHW21] Y. Roh, G. Heo, S. E. Whang. „A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective“. In: *IEEE Transactions on Knowledge and Data Engineering* 33.4 (2021), S. 1328–1347. DOI: 10.1109/TKDE.2019.2946162 (zitiert auf S. 9).

[RML14] P. Rebentrost, M. Mohseni, S. Lloyd. „Quantum Support Vector Machine for Big Data Classification“. In: *Phys. Rev. Lett.* 113 (13 Sep. 2014). DOI: 10.1103/PhysRevLett.113.130503 (zitiert auf S. 9).

[RT19] R. Raz, A. Tal. „Oracle Separation of BQP and PH“. In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. Association for Computing Machinery, 2019, S. 13–23. ISBN: 9781450367059. DOI: 10.1145/3313276.3316315 (zitiert auf S. 9, 11).

[SBG+19] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, N. Killoran. „Evaluating analytic gradients on quantum hardware“. In: *Phys. Rev. A* 99 (3 März 2019). DOI: 10.1103/PhysRevA.99.032331 (zitiert auf S. 30, 32).

[SCH+22] K. Sharma, M. Cerezo, Z. Holmes, L. Cincio, A. Sornborger, P. Coles. „Reformulation of the No-Free-Lunch Theorem for Entangled Datasets“. In: *Physical Review Letters* 128 (Feb. 2022). DOI: 10.1103/PhysRevLett.128.070501 (zitiert auf S. 21).

[SES+20] N. Stamatopoulos, D. J. Egger, Y. Sun, C. Zoufal, R. Iten, N. Shen, S. Woerner. „Option Pricing using Quantum Computers“. In: *Quantum* 4 (Juli 2020), S. 291. DOI: 10.22331/q-2020-07-06-291 (zitiert auf S. 9).

[SF14] L. Susskind, A. Friedman. *Quantum mechanics: the theoretical minimum*. 2014. ISBN: 9780141977812. DOI: 10.1119/1.4890980 (zitiert auf S. 22).

[SHM+16] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot et al. „Mastering the game of Go with deep neural networks and tree search“. In: *nature* 529.7587 (2016), S. 484–489. DOI: 10.1038/nature16961 (zitiert auf S. 9, 17).

[Sho94] P. Shor. „Algorithms for quantum computation: discrete logarithms and factoring“. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, S. 124–134. DOI: 10.1109/SFCS.1994.365700 (zitiert auf S. 11).

[SIKC20] J. Stokes, J. Izaac, N. Killoran, G. Carleo. „Quantum Natural Gradient“. In: *Quantum* 4 (Mai 2020), S. 269. DOI: 10.22331/q-2020-05-25-269 (zitiert auf S. 56).

[SK13] H.-R. Schwarz, N. Köckler. *Numerische mathematik*. Springer-Verlag, 2013. ISBN: 9783834815514. DOI: 10.1007/978-3-8348-8166-3 (zitiert auf S. 27).

[Smi85] G. D. Smith. *Numerical solution of partial differential equations: finite difference methods*. Oxford university press, 1985. ISBN: 0198596413. DOI: 10.2307/3616228 (zitiert auf S. 27).

[SMM+22] S. H. Sack, R. A. Medina, A. A. Michailidis, R. Kueng, M. Serbyn. „Avoiding barren plateaus using classical shadows“. In: *arXiv preprint arXiv:2201.08194* (2022). DOI: 10.48550/ARXIV.2201.08194 (zitiert auf S. 56).

[Spa87] J. C. Spall. „A stochastic approximation technique for generating maximum likelihood parameter estimates“. In: *1987 American control conference*. IEEE. 1987, S. 1161–1167. DOI: 10.23919/ACC.1987.4789489 (zitiert auf S. 29).

[Spa98] J. C. Spall. „Implementation of the simultaneous perturbation algorithm for stochastic optimization“. In: *IEEE Transactions on Aerospace and Electronic Systems* 34.3 (1998), S. 817–823. DOI: 10.1109/7.705889 (zitiert auf S. 30).

[SSSI11] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel. „The German traffic sign recognition benchmark: a multi-class classification competition“. In: *The 2011 international joint conference on neural networks*. IEEE. 2011, S. 1453–1460. DOI: 10.1109/IJCNN.2011.6033395 (zitiert auf S. 17).

[SWM+20] R. Sweke, F. Wilde, J. Meyer, M. Schuld, P. Fährmann, B. Meynard-Piganeau, J. Eisert. „Stochastic gradient descent for hybrid quantum-classical optimization“. In: *Quantum* 4 (Aug. 2020), S. 314. DOI: 10.22331/q-2020-08-31-314 (zitiert auf S. 19, 26).

[TB06] A. G. Taube, R. J. Bartlett. „New perspectives on unitary coupled-cluster theory“. In: *International Journal of Quantum Chemistry* 106.15 (2006), S. 3393–3401. DOI: https://doi.org/10.1002/qua.21198 (zitiert auf S. 56).

[TBB+22] F. Truger, M. Beisel, J. Barzen, F. Leymann, V. Yussupov. „Selection and Optimization of Hyperparameters in Warm-Started Quantum Optimization for the MaxCut Problem“. In: *Electronics* 11.7 (März 2022). DOI: 10.3390/electronics11071033. URL: https://www.mdpi.com/2079-9292/11/7/1033 (zitiert auf S. 56).

[TMGB19] F. Tacchino, C. Macchiavello, D. Gerace, D. Bajoni. „An artificial neuron implemented on an actual quantum processor“. In: *npj Quantum Information* 5.1 (2019), S. 1–8. DOI: 10.1038/s41534-019-0140-4 (zitiert auf S. 21).

[VMS04] J. J. Vartiainen, M. Möttönen, M. M. Salomaa. „Efficient Decomposition of Quantum Gates“. In: *Phys. Rev. Lett.* 92 (17 Apr. 2004). DOI: 10.1103/PhysRevLett.92.177902 (zitiert auf S. 53).

[WIWL21] D. Wierichs, J. Izaac, C. Wang, C. Y.-Y. Lin. „General parameter-shift rules for quantum gradients“. In: *arXiv preprint arXiv:2107.12390* (2021) (zitiert auf S. 33, 35, 56).

[ZG21] C. Zhao, X.-S. Gao. „QDNN: deep neural networks with quantum layers“. In: *Quantum Machine Intelligence* 3 (Juni 2021). DOI: 10.1007/s42484-021-00046-w (zitiert auf S. 21).

[ZPWV17] L. Zhou, S. Pan, J. Wang, A. V. Vasilakos. „Machine learning on big data: Opportunities and challenges“. In: *Neurocomputing* 237 (2017), S. 350–361. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2017.01.026 (zitiert auf S. 9).

[ZVSW03] J. Zhang, J. Vala, S. Sastry, K. B. Whaley. „Geometric theory of nonlocal two-qubit operations“. In: *Phys. Rev. A* 67 (4 Apr. 2003). DOI: 10.1103/PhysRevA.67.042313 (zitiert auf S. 31).

Alle URLs wurden zuletzt am 02. 06. 2022 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift