Frankfurt University of Applied Sciences

– Fachbereich 2: Informatik und Ingenieurwissenschaften–

Nebenläufige Algorithmen im maschinellen Lernen: Analyse, Implementierung und vergleichende Untersuchungen zur Parallelisierung einer Bibliothek für künstliche neuronale Netze

Abschlussarbeit zur Erlangung des akademischen Grades
Bachelor of Science (B.Sc.)

vorgelegt am 21. Mai 2023 von

Luca Andrea John Vinciguerra

Matrikelnummer: 1296334

Referent	:	Prof. Dr. Thomas Gabel
Korreferent	:	Prof. Dr. Christian Baun

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Frankfurt, 21. Mai 2023

Luca Andrea John Vinciguerra

Inhaltsverzeichnis

Kurzfassung

Kapitel 1

Einleitung

Die Informatik schöpft oft aus der Natur, sei es durch die Nachahmung tierischer Bewegungen bei Robotern, die Organisation von Multiagentensystemen in vogelähnlichen Schwärmen oder die Anwendung evolutionärer Algorithmen zur Simulation natürlicher Prozesse. Doch eines der faszinierendsten Phänomene der Natur ist das menschliche Gehirn und seine Fähigkeit, aus Erfahrung zu lernen. Dieses komplexe Organ beschäftigt Wissenschaftler seit langem, und die Suche nach Möglichkeiten, seine Lernfähigkeit zu simulieren, hat zu bedeutenden Fortschritten geführt.

Ein zentrales Element im Gehirn ist das Neuron, das als Grundbaustein für die Informationsverarbeitung dient. Ein Neuron besteht aus einem Zellkörper, mehreren Dendriten und einem Axon. Die Dendriten empfangen elektrische Signale von vorgeschalteten Neuronen und fungieren somit als Eingangsebene für Informationen. Diese eingehenden Signale werden zum Zellkörper weitergeleitet, wo sie aufsummiert werden. Wenn ein bestimmter Schwellenwert überschritten wird, leitet das Neuron das elektrische Signal über das Axon weiter. Das Axon wiederum steht über den synaptischen Spalt mit den Dendriten nachgeschalteter Neuronen in Verbindung. Am synaptischen Spalt werden die elektrischen Signale in chemische Botenstoffe umgewandelt und ausgeschüttet. Je nach Menge der ausgeschütteten Botenstoffe entstehen in den Dendriten der nachgeschalteten Neuronen elektrische Signale, wodurch Informationen zwischen Neuronen ausgetauscht werden. Das Axon agiert somit als Ausgangsebene für Informationen.

Inspiriert von diesem biologischen Vorbild entwickelte Frank Rosenblatt 1958 das mathematische Modell des einfachen Perzeptrons – ein künstliches Neuron, das auch heute noch als Grundlage für die Entwicklung künstlicher neuronaler Netzwerke dient . Diese Netzwerke können mithilfe von Lernalgorithmen trainiert werden, um vielfältige Klassifikations- und Regressionsprobleme zu lösen. Heutzutage finden künstliche neuronale Netzwerke in verschiedenen Bereichen Anwendung, darunter die Bioinformatik, Bild- und Spracherkennung sowie die Finanzprognose , um nur einige zu nennen.

Parallel zur Weiterentwicklung von neuronalen Netzwerken ist die Frage der Skalierbarkeit und Effizienz von entscheidender Bedeutung. Insbesondere die Verarbeitung großer Datenmengen erfordert effiziente Algorithmen und Techniken zur Parallelisierung. In dieser Arbeit werden wir uns daher auch mit der Parallelisierung von neuronalen Net-

zen befassen und untersuchen, wie diese Techniken die Leistung und Effizienz unserer Implementierungen beeinflussen.

1.1 Aufgabenstellung

In Anbetracht der weitverbreiteten Verwendung von Mehrkernprozessoren in modernen Computersystemen und der dadurch ermöglichten Parallelisierung zur Steigerung der Leistungsfähigkeit, ist es von besonderem Interesse, die Leistung der bestehenden N++ Bibliothek für maschinelles Lernen durch Parallelisierung zu verbessern. Die N++ Bibliothek ist in C++ implementiert, weshalb die Parallelisierung mithilfe von Threads realisiert werden soll. Eine zentrale Herausforderung besteht darin, geeignete Stellen im Algorithmus zu identifizieren, die von der Parallelisierung profitieren könnten. Hierbei können bereits vorhandene Bachelorarbeiten und Implementierungen als Vergleich herangezogen werden.

Die vorliegende Arbeit zielt darauf ab, die erzielte Leistungsverbesserung durch die Parallelisierung zu untersuchen und zu quantifizieren. Durch die Implementierung der Parallelisierung und die anschließende Ausführung von ausgewählten Algorithmen des maschinellen Lernens auf geeigneten Datensätzen, können wir den Effekt der Parallelisierung auf die Leistung der N++ Bibliothek evaluieren. Zudem werden wir die Auswirkungen verschiedener Parameter und Konfigurationen im Zusammenhang mit der Parallelisierung analysieren, um ein umfassendes Verständnis der Leistungsverbesserung durch Parallelisierung zu erlangen.

Insgesamt strebt diese Arbeit danach, nicht nur die technische Umsetzung der Parallelisierung zu präsentieren, sondern auch deren Auswirkungen auf die Leistungsfähigkeit der N++ Bibliothek für maschinelles Lernen zu untersuchen und zu bewerten.

1.2 Gliederung

Kapitel 2

Grundlagen

Das folgende Kapitel legt die Grundlagen für künstliche neuronale Netzwerke dar, indem es detailliert auf ihre Struktur und Funktionsweise eingeht, insbesondere für vorwärtsgerichtete Netzwerke. Dabei wird ein umfassender Überblick über die potenziellen Anwendungsbereiche von künstlichen neuronalen Netzwerken gegeben, wobei deren Rolle in verschiedenen Bereichen wie Bilderkennung, Sprachverarbeitung und Mustererkennung hervorgehoben wird.

Des Weiteren wird die Thematik der Parallelisierung sowohl im allgemeinen Kontext als auch speziell im Zusammenhang mit neuronalen Netzwerken erläutert. Es werden die Vor- und Nachteile dieser Technik beleuchtet und die gängigsten Methoden zur Parallelisierung von Berechnungen in neuronalen Netzwerken werden ausführlich diskutiert. Dabei wird besonders auf die Parallelisierung von Berechnungen innerhalb vorwärtsgerichteter Netzwerke eingegangen.

Die Nutzung von Grafikprozessoren (GPUs) für parallele Berechnungen wird dabei angesprochen, jedoch liegt der Fokus auf den grundlegenden Prinzipien der Parallelisierung von neuronalen Netzwerken und deren Implementierung mittels Thread- und Prozessparallelsierung. Durch das fundierte Verständnis der zugrunde liegenden Konzepte können wir die Potenziale und Herausforderungen der Parallelisierung in diesem spezifischen Kontext besser einschätzen und geeignete Ansätze zur Leistungssteigerung identifizieren.

2.1 Grundprinzipien von neuronalen Netzwerken

Neuronale Netzwerke sind ein wesentlicher Bestandteil des maschinellen Lernens und der künstlichen Intelligenz. Sie sind inspiriert von der Funktionsweise des menschlichen Gehirns und bestehen aus einer Ansammlung miteinander verbundener Knoten, die als Neuronen bezeichnet werden. Diese Netzwerke können eine Vielzahl von Aufgaben ausführen, von der Bilderkennung bis hin zur Sprachverarbeitung.

Die Funktionsweise eines neuronalen Netzwerks lässt sich grob in zwei Hauptphasen unterteilen: Vorwärtspropagierung und Rückwärtspropagierung. Während der Vorwärtspropagierung fließen die Daten durch das Netzwerk, beginnend mit den Ein-

gangsneuronen, die die Rohdaten empfangen, und endend mit den Ausgangsneuronen, die die Vorhersagen oder Klassifikationen des Netzwerks liefern. Jedes Neuron in einem neuronalen Netzwerk ist mit anderen Neuronen verbunden, und diese Verbindungen sind mit Gewichten versehen, die die Stärke der Verbindung zwischen den Neuronen darstellen.

Während der Vorwärtspropagierung durchläuft jede Eingabe eine Reihe von Schichten im Netzwerk, wobei jede Schicht aus einer bestimmten Anzahl von Neuronen besteht. Jedes Neuron in einer Schicht erhält Inputs von den Neuronen der vorherigen Schicht, multipliziert diese Inputs mit den entsprechenden Gewichten und summiert sie. Anschließend wird eine Aktivierungsfunktion auf die gewichtete Summe angewendet, um die Ausgabe des Neurons zu berechnen, die dann an die Neuronen der nächsten Schicht weitergeleitet wird.

Die Rückwärtspropagierung ist der Prozess, bei dem das Netzwerk lernt, indem es seine Gewichte entsprechend der Fehler zwischen den tatsächlichen und den vorhergesagten Ausgaben anpasst. Dies geschieht durch die Berechnung von Gradienten mit Hilfe des Backpropagation-Algorithmus und die Anpassung der Gewichte mithilfe eines Optimierungsalgorithmus wie dem Gradientenabstiegsverfahren.

Insgesamt ermöglicht die Funktionsweise von neuronalen Netzwerken die Modellierung komplexer Zusammenhänge in Daten und die Durchführung verschiedenster Aufgaben des maschinellen Lernens und der künstlichen Intelligenz.

2.2 Aufbau eines neuronalen Netzwerks

2.2.1 Neuronen und deren Verbindungen

2.2.2 Schichten und ihre Funktionen

2.2.3 Vorwärtsgerichtete und rückwärtsgerichtete Netzwerke

2.3 Anwendungsfälle von neuronalen Netzwerken

2.4 Einführung in die Parallelisierung

2.4.1 Definition und Herangehensweise

2.4.2 Vor- und Nachteile von Parallelisierung

2.5 Parallelisierung in vorwärtsgerichteten Netzwerken

2.5.1 Thread- und Prozessparallelsierung

2.5.2 Implementierung von Parallelisierungstechniken

2.5.3 Auswirkungen auf die Leistungsfähigkeit

Kapitel 3

Implementierung der Parallelisierung in N++

3.1 Erläuterung des N++ Simulatorkerns

3.2 Voraussetzungen

3.2.1 Entfernung von Data Races und Deadlocks

3.2.2 Verlagerung der zu parallelisierenden Routine

3.3 Vorstellung der Implementierung

3.3.1 Verwendung von threadsicheren Funktionen

3.3.2 Entfernung globaler Variablen

Kapitel 4

Experimentelle Untersuchungen

4.1 Verfahren A: ein neues Verfahren zur Steuerung von ...

Kapitel 5

Ergebnisse

Kapitel 6

Fazit und Ausblick

Anhang A

Programmcode