



Neuroevolution mit MPI

Analyse und Optimierung von NEAT für ein
verteiltes System

Masterthesis

zur Erlangung des akademischen Grades
Master of Science (M.Sc.)
im Studiengang Angewandte Informatik
an der Hochschule Flensburg

Simon Hauck

Matrikelnummer: 660158

Erstprüfer: Prof. Dr. rer. nat. Tim Aschmoneit
Zweitprüfer: Prof. Dr. rer. nat. Torben Wallbaum

3. Juli 2020

Neuroevolutionäre Algorithmen sind ein mögliches Optimierungsverfahren für neuronale Netze. Abhängig von dem verwendeten Algorithmus können die Gewichte der Verbindungen im Netz und die Struktur entwickelt und optimiert werden.

Der Optimierungsprozess ist, unabhängig vom Verfahren, sehr aufwändig und dementsprechend zeit- und rechenintensiv. Für eine schnellere Durchführung des Trainingsprozesses bieten sich Algorithmen an, die gut parallelisierbar sind. Die benötigte Ausführungszeit dieser kann durch Hinzufügen weiterer Rechenknoten mit geringem Aufwand maßgeblich reduziert werden.

Neuroevolutionäre Algorithmen bieten sich aufgrund der Verfahrensweise und der vielen unabhängigen neuronalen Netzen für eine parallele Ausführung an.

In dieser Arbeit wird, stellvertretend für neuroevolutionäre Algorithmen, der NeuroEvolution of Augmenting Topologies (NEAT) Algorithmus betrachtet. Dieser wurde im Jahr 2002 veröffentlicht und ist im Vergleich zu den damals bekannten Algorithmen besonders effizient. Zudem dient der Algorithmus als Grundlage für viele Erweiterungen. Die erhaltenen Ergebnisse dieser Arbeit lassen sich somit gut auf ebendiese Erweiterungen übertragen.

Im ersten Schritt dieser Arbeit wird die Laufzeit des NEAT Algorithmus mit verschiedenen Optimierungsaufgaben analysiert. Mit den erhaltenen Ergebnissen wird eine parallelisierte Implementierung erstellt. Diese führt mit unterschiedlich vielen Rechenknoten dieselben Optimierungsaufgaben durch. Am Ende dieser Arbeit werden die Ergebnisse von beiden Implementierungen verglichen.

Inhaltsverzeichnis

1	Motivation	1
1.1	Problemstellung	1
1.2	Ziel der Arbeit	1
1.3	Struktur der Arbeit	1
2	Grundlagen	2
2.1	Neuronale Netze	2
2.1.1	Biologische neuronale Netze	2
2.1.2	Künstliche neuronale Netze	5
2.1.3	Das Neuron	6
2.1.4	Netzstrukturen	7
2.1.5	Optimierungsverfahren	7
2.2	Evolutionäre Algorithmen	8
2.3	NEAT	8
2.4	MPI	8
3	Analyse	9
3.1	Anforderungen	9
3.2	Softwarearchitektur und Implementierung	9
3.3	Testsetup	9
3.4	Evaluation	9
4	Software Architektur und Implementierung	10
5	Evaluation	11
5.1	Testsetup	11
5.2	Ergebnisse	11
6	Zusammenfassung und Ausblick	12
	Quellenverzeichnis	13
	Eidesstattliche Erklärung	15

Abbildungsverzeichnis

2.1	Schematische Abbildung einer Nervenzelle, Quelle [1].	4
-----	---	---

Akronymverzeichnis

T	Test
roAAAA	Very much A's
NEAT	NeuroEvolution of Augmenting Topologies
KNN	Künstliche neuronale Netze
PNS	Periphere Nervensystem
ZNS	Zentrale Nervensystem

1 Motivation

1.1 Problemstellung

1.2 Ziel der Arbeit

1.3 Struktur der Arbeit

2 Grundlagen

2.1 Neuronale Netze

Klassische Algorithmen in der Informatik beschreiben mit welchen Schritten ein spezielles Problem gelöst werden kann. In vielen Anwendungsfällen, wie zum Beispiel beim Sortieren einer Liste, verwenden Computersysteme diese und lösen das gegebene Problem schneller und effizienter als es Menschen möglich ist.

Dennoch gibt es Aufgaben, die von Menschen ohne Aufwand gelöst werden, aber Computersysteme vor große Herausforderungen stellen. Hierzu zählt unter anderem die Klassifizierung von Bildern. Ein Mensch kann Bilder von Hunden und Katzen unabhängig von Blickwinkel und Bildqualität unterscheiden beziehungsweise richtig zuordnen. Trotzdem lassen sich für solche Probleme keine klassischen Algorithmen finden, da die Lösung von vielen subtilen Faktoren abhängt [1].

In vielen dieser Aufgabenfelder werden Künstliche neuronale Netze (KNN) eingesetzt, welche von den biologischen neuronalen Netzen inspiriert sind und zum Forschungsgebiet des maschinellen Lernens gehören. Die Grundlage für die KNN bildet die Arbeit von McCulloch und Pitts, in der sie 1943 ein einfaches neuronales Netz mit Schwellwerten entwickelt haben. Dies ermöglicht die Berechnung von logischen und arithmetischen Funktionen [2]. In den folgenden Jahrzehnten wird die Funktionsweise der neuronalen Netze weiterentwickelt und der Einsatz in verschiedensten Aufgabenfeldern ermöglicht. Hierzu zählen neben der Klassifizierung von Bildern [3] unter anderem das Erkennen und die Interpretation von Sprache [4], [5] sowie das selbständige Lösen von Computer- und Gesellschaftsspielen [6], [7].

In diesem Kapitel wird zuerst ...

2.1.1 Biologische neuronale Netze

Wie bereits beschrieben orientiert sich das Fachgebiet der KNN an den erfolgreichen biologischen neuronalen Netzen, wie zum Beispiel dem menschlichen Gehirn [1]. In diesem Abschnitt werden die Eigenschaften betrachtet, die das Vorbild erfolgreich machen und für die KNN übernommen werden sollen. Im Zuge dessen wird ein grober Überblick über die Struktur und Funktionsweise des menschlichen Gehirns gegeben.

Jede Sekunde erfassen die Rezeptoren des menschlichen Körpers unzählige Reize, wie zum Beispiel Licht, Druck, Temperatur und Töne. Die Reize werden anschließend elektrisch oder chemisch kodiert und über Nervenbahnen an das Gehirn geleitet, welches die Aufgabe hat diese zu filtern, zu verarbeiten und entsprechend zu reagieren. Als Reaktion

können zum Beispiel Signale an entsprechende Muskeln oder Drüsen gesendet werden [8].

Bevor im nächsten Kapitel die Funktionsweise des Gehirns näher betrachtet wird, sollen drei Eigenschaften genannt werden, die klassische Algorithmen nicht besitzen beziehungsweise nur schwer umsetzen können, aber für biologische neuronale Netze keine Herausforderung sind. Ziel ist es, diese für die KNN zu übernehmen [1].

1. Fähigkeit zu Lernen

Das menschliche Gehirn ist nicht wie ein klassischer Algorithmus für seine Aufgaben programmiert. Stattdessen besitzt es die Fähigkeit anhand von gegebenen Beispielen und oder einfachem Ausprobieren zu lernen [1]. Hierbei wird das gewünschte Ergebnis mit dem tatsächlich erhaltenen Ergebnis verglichen und das Verhalten entsprechend angepasst. Dies ermöglicht es Menschen verschiedenste Aufgabengebiete erfolgreich zu lösen und sich ändernden Anforderungen anzupassen.

2. Fähigkeit zur Generalisierung

Allerdings kann nicht jedes mögliche Szenario für ein Aufgabenfeld durch Ausprobieren oder Beobachtung gelernt werden. Trotzdem trifft das Gehirn in den meisten Situationen plausible Lösungen, da es die Fähigkeit zur Generalisierung besitzt [1]. Das bedeutet, dass viele Situationen bereits bekannten Problemen zugeordnet werden können, mithilfe derer eine passende Verhaltensstrategie ausgewählt wird.

3. Toleranz gegenüber Fehlern

Die Fähigkeit zu Generalisieren erlaubt auch eine hohe Fehlertoleranz gegenüber verrauschten Daten. Bei dem oben genannten Beispiel der Klassifizierung von Bildern kann ein Teil des Bildes fehlen oder unscharf sein und trotzdem kann das abgebildete Motiv richtig zugeordnet werden.

Struktur des menschlichen Gehirns

Die Forschungsgebiet der Neurowissenschaften befasst sich unter anderem mit dem menschlichen Gehirn, dessen Funktionsweise noch nicht vollständig nachvollzogen werden kann. Trotzdem ist schon seit 1861 durch die Arbeit von Paul Broca bekannt, dass es im menschlichen Gehirn verschiedene Regionen mit unterschiedlichen Aufgaben gibt [9]. Zum Beispiel wird das sogenannte Kleinhirn (Cerebellum) für einen Großteil der motorischen Koordination verwendet während an das Großhirn (Telencephalon) unter anderem visuelle Reize geleitet werden [1]. Trotz der unterschiedlichen Aufgaben haben alle Bereiche des Gehirns einen gemeinsamen Grundbaustein, die sogenannten Neuronen [9]. Im folgenden wird der Aufbau und die Funktionsweise von diesen oberflächlich im Bezug zu den später vorgestellten künstlichen Neuronen betrachtet. Für einen vollständigen Überblick und eine genaue Beschreibung der Vorgänge wird auf entsprechende Fachliteratur verwiesen.

Das menschliche Gehirn besitzt ungefähr 10^{11} einzelne Neuronen, deren schematischer Aufbau in Abbildung 2.1 dargestellt ist. Jedes Neuron besitzt einen Zellkern, der sich im

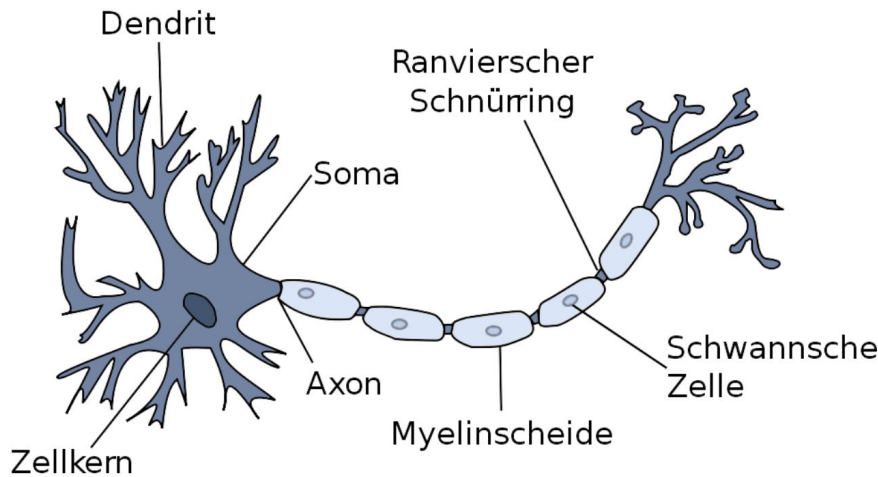


Abbildung 2.1: Schematische Abbildung einer Nervenzelle, Quelle [1].

Zellkörper (Soma) befindet. Von dem Zellkörper gehen mehrere Fasern aus, die Dendriten genannt werden [9]. An diesen befinden sich Synapsen, welche als Übertragungsstelle fungieren und elektrische oder chemische Signale von Rezeptoren oder anderen Neuronen empfangen [1]. Typischerweise empfängt ein Neuron Signale von 2000 und 10.000 anderen Nervenzellen [10].

Synapsen, die elektrische Signale empfangen, haben eine starke, direkte, nicht regulierbare Verbindung vom Sender zum Empfänger. Diese sind für hart kodierte Verhaltensmechanismen nützlich wie zum Beispiel den Fluchtreflex. Die chemische Synapse hingegen ist nicht direkt mit dem Sender verbunden, sondern durch den synaptischen Spalt getrennt [1]. Zur Übertragung eines elektrischen Signals wird dieses auf der präsynaptischen Seite in ein chemisches Signal kodiert, indem Neurotransmitter freigesetzt werden. Diese können über den synaptischen Spalt übertragen und anschließend auf der postsynaptischen Seite wieder in ein elektrisches Signal kodiert werden. Ein großer Vorteil dieser Übertragungsart ist die Regulierbarkeit [1]. Verschiedene Neurotransmitter können unterschiedliche Effekte auf das Neuron haben, beispielsweise anregend (exzitatorisch) oder hemmend (inhibitorisch) [11]. Zusätzlich kann die Menge der freigesetzten Neurotransmitter die Stärke des Signals beeinflussen [1]. Auf lange Zeit gesehen können auch neue Verbindungen entstehen oder alte aufgelöst werden. Es wird angenommen, dass dies die Grundlage des Lernens im menschlichen Gehirn ist [9].

Sowohl die erregenden als auch hemmenden Signale werden über die Dendriten an den Axonhügel weitergeleitet, welcher sich zwischen dem Soma und dem Axon befindet. Dort werden die Signale akkumuliert. Wird bei diesem Vorgang ein gewisser Schwellwert überschritten, wird ein elektrischer Impuls erzeugt der über das Axon weitergeleitet wird [11]. Das Axon ist typischerweise 1cm in Ausnahmen sogar bis zu einem 1m lang und von der Myelinscheide umgeben, die unter anderem Schutz vor mechanischer Überanspruchung bietet [9]. Zusammen mit den Ranvierschen Schnürringen ermöglicht diese zudem eine schnellere Weiterleitung des Aktionspotenzials [11]. Das Axon endet mit dem sogenann-

ten Endknopf oder auch Axonterminal genannt. Dieses ist mit den Synapsen von anderen Neuronen verbunden und kann beim Eintreffen eines Signals die Neurotransmitter freisetzen und somit das Signal übertragen [11]. Typischerweise gibt ein einzelnes Neuron sein Signal an 1000 bis 10.000 anderen Neuronen weiter, in Ausnahmefällen sogar an bis zu 150.000 andere Neuronen [10], die alle parallel arbeiten. So entsteht ein sehr großes und leistungsfähiges neuronales Netz.

2.1.2 Künstliche neuronale Netze

KNN sind ein mathematisches Modell, dass im Vergleich zum biologischen Vorbild stark vereinfacht und idealisiert. Trotzdem können unterschiedlichste mathematische Funktionen abgebildet werden. In diesem Kapitel wird die grundsätzliche Funktionsweise sowie die einzelnen Komponenten der KNN vorgestellt.

Betrachtet man ein KNN als Blackbox (TODO REFERENZ BILD), gibt es eine Menge von Eingabewerten, die in einem Eingabevektor kodiert sind und eine Menge an Ausgaben, die in einem Ausgabevektor kodiert sind [12]. Die Eingaben werden im Falle der KNN nicht durch Rezeptoren erfasst sondern durch ein Optimierungsproblem gegeben. Der Ausgabevektor soll das gewünschte Ergebnis enthalten. Die Interpretation von diesem variiert je nach Optimierungsproblem und Netzarchitektur.

Betrachtet man die Struktur der KNN sind einige Ähnlichkeiten zum biologischen Vorbild erkennbar. Diese werden im folgenden genauer betrachtet [10]:

1. Neuronen

Ähnlich zu den biologischen neuronalen Netzen, besteht auch das KNN aus vielen Neuronen [10]. Dies sind einfache Recheneinheiten, die primitive Funktionen bestimmen können [12] und deren genaue Funktionsweise in Kapitel 2.1.3 erläutert wird. Vorweggenommen sei, dass ein Neuron mehrere Eingabewerte besitzt, welche gewichtet sind und akkumuliert werden. Hierbei entsteht ein skalarer Ausgabewert, der den Aktivierungsgrad des Neurons repräsentiert und von anderen Neuronen als Eingabe verwendet werden kann [1].

2. Gerichtete gewichtete Verbindungen

Wie im vorherigen Punkt angedeutet, sind Neuronen über gerichtete Verbindungen miteinander vernetzt. Der Aktivierungszustand eines Neurons wird entsprechend der Verbindungen an die Zielneuronen weitergegeben, welche diesen Wert als Eingabe verarbeiten. Wie bei den biologischen neuronalen Netzen auch, können Eingaben unterschiedlich stark anregend und hemmend wirken. Dies wird bei den KNN über Gewichtete in den Verbindungen realisiert [10].

3. Struktur und Gewichte

Der Ausgabevektor eines KNN ist abhängig von der Struktur des Netzwerkes und der Gewichte in den einzelnen Verbindungen. Für das erfolgreiche Lösen eines Optimierungsproblems muss ein KNN die richtige Kombination von Neuronen,

Netzwerkstruktur und gewichteten Verbindungen besitzen. Diese müssen durch Lernverfahren bestimmt werden, auf die in Kapitel 2.1.5 näher eingegangen wird.

Trotz der vorgestellten Ähnlichkeiten, gibt es sehr viele Unterschiede zwischen den biologischen neuronalen Netzen und den KNN. Beispiel hierfür ist der Größenunterschied. Das menschliche Gehirn mit seinen 10^{11} Neuronen besitzt pro Neuron ungefähr 10^4 Verbindungen, während die meisten KNN nur 10^2 bis 10^4 Neuronen mit insgesamt 10^5 Verbindungen besitzen. Auch werden keine chemischen Effekte die auf benachbarte Neuronen wirken sowie zeitliche und räumliche Lokalisierungsprinzipien beachtet [10]. Aus diesen Gründen sind die KNN keine Nachbildung der biologischen neuronalen Netzen sondern verwenden diese nur als Inspiration.

2.1.3 Das Neuron

In diesem Kapitel wird die genaue Funktionsweise der einzelnen Neuronen betrachtet. Hierfür werden drei Phasen vorgestellt, in denen die Ausgabe eines einzelnen Neurons berechnet wird. Betrachtet man ein KNN führen typischerweise mehrere Verbindungen zu einem Neuron j , welche von den Neuronen i_1, i_2, \dots, i_n ausgehen [1]. Dieses ist schematisch in Abbildung (TODO ABBILDUNG EINFÜGEN) dargestellt.

Propagierungsfunktion

Die Ausgabewerte $o_{i_1}, o_{i_2}, \dots, o_{i_n}$ der Neuronen i_1, i_2, \dots, i_n werden als Eingabewerte für das Neuron j verwendet. Für jeden Eingabewert existiert ein entsprechendes Gewicht w_1, w_2, \dots, w_n [1]. Somit repräsentiert w_{ij} das Gewicht für die Verbindung von Neuron i zu Neuron j [10]. Die Propagierungsfunktion f_{prop} berechnet die Netzeingabe net_j , welche in der nächsten Phase weiterverwendet wird [1].

$$net_j = f_{prop}(o_1, o_2, \dots, o_n, i_1, i_2, \dots, i_n)$$

Die meist verwendete Propagierungsfunktion ist die gewichtete Summe. Hierbei werden, entsprechend der Formel, die Werte o_i mit dem entsprechenden Gewicht w_i multipliziert und aufsummiert [1]:

$$net_j = \sum_i (o_i \cdot w_{i,j})$$

Aktivierungszustand

Der Aktivierungszustand $a_j(t)$ gibt den Grad der Aktivierung von Neuron j zum Zeitpunkt t an [10]. Ein neuer Aktivierungszustand zum Zeitpunkt $t + 1$ wird mit der Aktivierungsfunktion f_{act} berechnet. Diese berücksichtigt nicht nur die Netzeingabe $net_j(t)$ sondern auch den vorherigen Aktivierungszustand $a_j(t)$ und den Schwellwert Θ der Aktivierungsfunktion [10]. Ein Schwellwert Θ_j ist einem Neuron j zugeordnet und gibt die Stelle an, an welcher die Aktivierungsfunktion die größte Steigung hat [1]. Somit kann

die Berechnung der Aktivierung $a_j(t+1)$ durch folgende Formel ausgedrückt werden:

TOlleformel

2.1.4 Netzstrukturen

2.1.5 Optimierungsverfahren

Das Gebiet der Künstlichen Neuronalen Netze wird bereits seit 1943 erforscht
Test (T) and [14] with Stanley und Miikkulainen

asfkajsfö

2.2 Evolutionäre Algorithmen

2.3 NEAT

2.4 MPI

3 Analyse

3.1 Anforderungen

3.2 Softwarearchitektur und Implementierung

3.3 Testsetup

3.4 Evaluation

4 Software Architektur und Implementierung

5 Evaluation

5.1 Testsetup

5.2 Ergebnisse

6 Zusammenfassung und Ausblick

Quellenverzeichnis

- [1] David Kriesel. 2008. Ein kleiner überblick über neuronale netze. *Download unter <http://www.dkriesel.com/index.php>*.
- [2] Warren S McCulloch und Walter Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5, 4, 115–133.
- [3] Alex Krizhevsky, Ilya Sutskever und Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.
- [4] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath u. a. 2012. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal processing magazine*, 29, 6, 82–97.
- [5] Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov und Michael Collins. 2016. Globally normalized transition-based neural networks. *arXiv preprint arXiv:1603.06042*.
- [6] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra und Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- [7] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot u. a. 2016. Mastering the game of go with deep neural networks and tree search. *nature*, 529, 7587, 484.
- [8] Werner Kinnebrock. 2018. *Neuronale Netze: Grundlagen, Anwendungen, Beispiele*. Walter de Gruyter GmbH & Co KG.
- [9] Stuart Russell und Peter Norvig. 2013. Künstliche intelligenz. ein moderner ansatz, 3. ak. aufl. (2013).
- [10] Andreas Zell. 2003. Simulation neuronaler netze. 4., unveränderte auflage. (2003).
- [11] Clemens Kirschbaum. 2008. *Biopsychologie von A bis Z*. Springer-Verlag.
- [12] Andreas Scherer. 2013. *Neuronale Netze: Grundlagen und Anwendungen*. Springer-Verlag.
- [13] Raul Rojas. 1996. *Neural Networks: A Systematic Introduction*. Springer Science & Business Media.

-
- [14] Kenneth O Stanley und Risto Miikkulainen. 2002. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10, 2, 99–127.

Eidesstattliche Erklärung

This is the beginning