



# 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

27. Juni 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

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

# **Abbildungsverzeichnis**

# Akronymverzeichnis

<b>T</b>	Test
<b>roAAAA</b>	Very much A's
<b>NEAT</b>	NeuroEvolution of Augmenting Topologies
<b>KNN</b>	Künstliche neuronale Netze
<b>PNS</b>	Periphere Nervensystem
<b>ZNS</b>	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**

Das menschliche Gehirn kann die genannten Eigenschaften durch seine besondere Struktur und Informationsverarbeitung erzielen. Diese wird im Folgenden oberflächlich erläutert, da dies die Grundlage für der KNN ist. Für ein tieferes biologisches Verständnis wird auf entsprechende Fachliteratur verwiesen. Das menschliche Gehirn kann in mehrere Regionen unterteilt werden. Das Großhirn Das menschliche Gehirn besteht aus  $10^{11}$  einzelnen Neuronen, welche unabhängig voneinander operieren. Jedes Neuron kann zu

#### **2.1.2 Das Neuron**

#### **2.1.3 Netzstrukturen**

#### **2.1.4 Optimierungsverfahren**

Das Gebiet der Künstlichen Neuronen Netze wird bereits seit 1943 erforscht  
Test (T) and [9] 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] 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