



hochschule mannheim

Entwicklung eines Efficiently Updatable Neural Network (NNUE) zur Evaluation von Schach Positionen

Marvin Karhan

Bachelor-Thesis

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

Studiengang Informatik

Fakultät für Informatik

Hochschule Mannheim

31.08.2022

Betreuer

Prof. Jörn Fischer, Hochschule Mannheim

Prof. Thomas Ihme, Hochschule Mannheim

Karhan, Marvin:

Entwicklung eines NNUE zur Evaluation von Schach Positionen / Marvin Karhan. –
Bachelor-Thesis, Mannheim: Hochschule Mannheim, 2022. 8 Seiten.

Karhan, Marvin:

Development of an NNUE for the Evaluation of Chess Positions / Marvin Karhan. –
Bachelor Thesis, Mannheim: University of Applied Sciences Mannheim, 2022. 8 pages.

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ich bin damit einverstanden, dass meine Arbeit veröffentlicht wird, d. h. dass die Arbeit elektronisch gespeichert, in andere Formate konvertiert, auf den Servern der Hochschule Mannheim öffentlich zugänglich gemacht und über das Internet verbreitet werden darf.

Mannheim, 31.08.2022

A handwritten signature in blue ink, appearing to read 'M. Karhan', written in a cursive style.

Marvin Karhan

Abstract

Entwicklung eines NNUE zur Evaluation von Schach Positionen

Jemand musste Josef K. verleumdet haben, denn ohne dass er etwas Böses getan hätte, wurde er eines Morgens verhaftet. Wie ein Hund! sagte er, es war, als sollte die Scham ihn überleben. Als Gregor Samsa eines Morgens aus unruhigen Träumen erwachte, fand er sich in seinem Bett zu einem ungeheueren Ungeziefer verwandelt. Und es war ihnen wie eine Bestätigung ihrer neuen Träume und guten Absichten, als am Ziele ihrer Fahrt die Tochter als erste sich erhob und ihren jungen Körper dehnte. Es ist ein eigentümlicher Apparat, sagte der Offizier zu dem Forschungsreisenden und überblickte mit einem gewissermaßen bewundernden Blick den ihm doch wohl bekannten Apparat. Sie hätten noch ins Boot springen können, aber der Reisende hob ein schweres, geknotetes Tau vom Boden, drohte ihnen damit und hielt sie dadurch von dem Sprunge ab. In den letzten Jahrzehnten ist das Interesse an Künstlern sehr zurückgegangen. Aber sie überwandten sich, umdrängten den Käfig und wollten sich gar nicht fortrühren.

Development of an NNUE for the Evaluation of Chess Positions

The European languages are members of the same family. Their separate existence is a myth. For science, music, sport, etc, Europe uses the same vocabulary. The languages only differ in their grammar, their pronunciation and their most common words. Everyone realizes why a new common language would be desirable: one could refuse to pay expensive translators. To achieve this, it would be necessary to have uniform grammar, pronunciation and more common words. If several languages coalesce, the grammar of the resulting language is more simple and regular than that of the individual languages. The new common language will be more simple and regular than the existing European languages. It will be as simple as Occidental; in fact, it will be Occidental. To an English person, it will seem like simplified English, as a skeptical Cambridge friend of mine told me what Occidental is.

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	2
2.1	Hand-crafted Evaluation	2
2.2	Neuronale Netze	2
2.2.1	Das Neuron	3
2.2.2	Backpropagation	3
2.2.3	Convolutional Neural Networks	3
2.2.4	Loss functions	3
2.2.5	Quantization	3
2.3	SIMD	3
2.3.1	Memory Alignment	3
3	Verwandte Arbeiten	4
4	NNUE	5
4.1	Architektur	5
4.1.1	Feature Set	5
4.1.2	Eingabeschicht	5
4.1.3	Versteckte Schicht	6
4.1.4	Ausgabeschicht	6
4.2	Training	6
4.2.1	Eingabedaten	6
4.3	Integration in einen Schachcomputer	6
4.3.1	Eingabeschicht	6
4.3.2	Versteckte Schicht	6
5	Ergebnisse	7
5.1	Verbesserungen	7
6	Fazit und Ausblick	8
	Abkürzungsverzeichnis	v
	Tabellenverzeichnis	vi

Abbildungsverzeichnis	vii
Literatur	viii

Kapitel 1

Einleitung

Computer Schach ist ein viel betrachtetes Thema. Schon Alan Turing und Claude Shannon hab sich damit befasst [1], [2].

Ein Schachcomputer kann

Kapitel 2

Grundlagen

In diesem Kapitel wird das Wissen vermittelt, welches benötigt wird um zu Verstehen, wie NNUEs im Rahmen von Schachcomputern funktionieren. Zuerst wird die Evaluation, wie sie in herkömmlichen Schachcomputern funktioniert erklärt, auch hand-crafted evaluation (HCE) genannt.

2.1 Hand-crafted Evaluation

Die Evaluation einer Schachposition ist eine heuristische Methode der Position einen numerischen Wert zuzuordnen. Hätten wir unendliche Ressourcen könnten wir aus jeder Position alle mögliche Zugfolgen sehen und der Positionen einer der drei Werte: -1 (Verlust), 0 (remis), 1 (Gewinn) geben. In der Realität ist es nicht möglich den exakten Wert der Stellung zu kennen, deshalb wird in der HCE versucht anhand von Menschen festgelegten Kriterien einen Wert der Position zuzuordnen. Vor der Verbreitung von neuronale Netze (NNs), war HCE die einzige Form der Positions-Evaluation.

2.2 Neuronale Netze

künstliche neuronale Netze (KNNs) oder einfach NNs genannt, sind Computer Systeme, die dem biologischen Vorbild des Gehirns nachempfunden sind. In Abbildung 2.1 ist ein einfaches neuronales Netz zu sehen. Es besteht aus NN bestehen aus

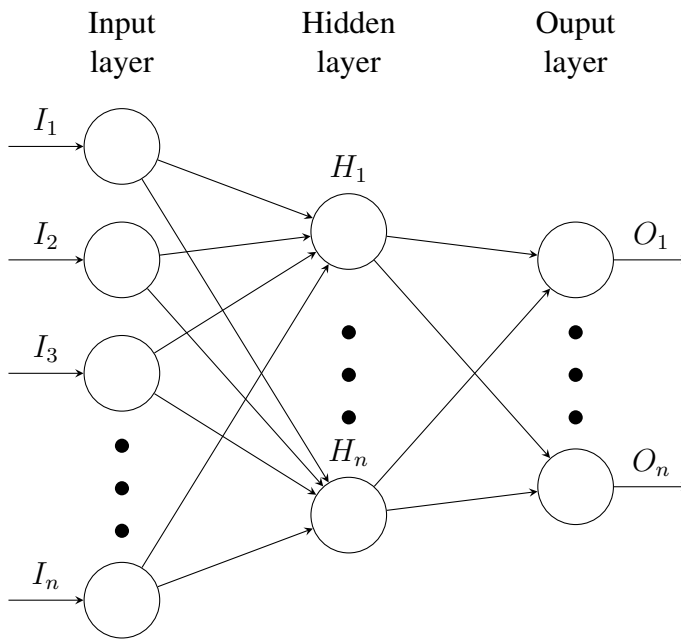


Abbildung 2.1: Ein exemplarisches neuronales Netz

2.2.1 Das Neuron

Das Neuron ist der elementare Bestandteil eines NNs. Um Neuronen eines NNs zu verstehen, schauen wir

2.2.2 Backpropagation

2.2.3 Convolutional Neural Networks

2.2.4 Loss functions

2.2.5 Quantization

2.3 SIMD

Single instruction, multiple data (SIMD)

2.3.1 Memory Alignment

Kapitel 3

Verwandte Arbeiten

Kapitel 4

NNUE

Ziel dies Kapitels ist es,

Kapitel Abschnitt 2.1 zeigt wie die herkömmliche Art und Weise der Positions-Evaluation funktioniert. Nach kurzer Überlegung wird aber klar, dass die HCE nur so gut sein kann wie die Schachspieler die sie Entwickeln. Natürlich können die darin verwendeten Parameter durch Optimierungsalgorithmen wie genetische Algorithmen oder Simulated Annealing maximiert werden, letztendlich bleibt der limitierende Faktor das Spielverständnis der Entwickler.

4.1 Architektur

4.1.1 Feature Set

4.1.2 Eingabeschicht

- sparse inputs -> little changes - max 32 inputs setboolean - input either 0 or 1

4.1.3 Versteckte Schicht

4.1.4 Ausgabeschicht

4.2 Training

4.2.1 Eingabedaten

Die Erzeugung der Eingabedaten ist nicht teil dieser Arbeit. Jedoch ist es wichtig zu wissen wie die Eingabedaten generiert werden und wie sie in den Trainer geladen werden, um zu verstehen, wie das neuronale Netz lernt. Im Training für diese Arbeit wurden drei verschiedene generierte Datensätze verwendet. Diese Datensätze wurden von Stockfish für das Training der neusten Variante ihres NNUEs verwendet [3]. Sie

4.3 Integration in einen Schachcomputer

4.3.1 Eingabeschicht

4.3.2 Versteckte Schicht

- simple -> quantization

Kapitel 5

Ergebnisse

5.1 Verbesserungen

Kapitel 6

Fazit und Ausblick

Wie in Kapitel 5 festgestellt, hat das NNUE hat nicht die erwarteten Ergebnisse geliefert. Dafür kann es mehrere Gründe geben:

Die Implementierung des Netzes in dem Schachcomputer kann Fehler enthalten. Die rekursive Art und Weise, der Zugsuche macht es nicht einfach mögliche Fehlerquellen zu identifizieren. In einer zeitlich limitierten Anstrengung Fehler in der Implementierung zu finden, wurde das Einlesen der Gewichte und Bias, sowie das Verhältnis zwischen updates und refreshes des Akkumulators überprüft. Es konnten keine Unregelmäßigkeiten festgestellt werden. Fehler an anderen Stellen der Implementierung sind wahrscheinlich und können ausschlaggebend für die vorzufindenden Ergebnisse sein. Leider waren weitere Tests im Rahmen dieser Arbeit nicht möglich.

Ein weiterer Grund für eine nicht vorhandene Steigerung der Spielstärke kann außerhalb der Evaluation liegen. Die zugrundeliegende Suche von des im Modul Künstliche Intelligenz für autonome Systeme (KIS) entwickelten Schachcomputers ist nur minimal optimiert. Sie besteht aus einem Negamax Depth-First Suchalgorithmus [4] mit Move Ordering, Transposition Tables, Quiescence Search und Iterative Deepening. Es fehlen viele weit verbreitete pruning Techniken wie unter anderem: razoring [5, S. 123-128], .

Abkürzungsverzeichnis

NNUE	Efficiently Updatable Neural Network
SIMD	Single instruction, multiple data
HCE	hand-crafted evaluation
KNN	künstliches neuronales Netz
NN	neuronales Netz
KIS	Künstliche Intelligenz für autonome Systeme

Tabellenverzeichnis

Abbildungsverzeichnis

2.1	Ein exemplarisches neuronales Netz	3
-----	--	---

Literatur

- [1] Alan Turing, *Faster Than Thought - A Symposium on Digital Computing Machines*. London: Sir Isaac Pitman & Sons, 1953, 1953.
- [2] Claude E. Shannon, „XXII. Programming a computer for playing chess“, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, Jg. 41, Nr. 314, S. 256–275, März 1950. DOI: 10.1080/14786445008521796.
- [3] Joost VandeVondele. „Update default net to nn-3c0054ea9860.nnu“. (2013), Adresse: <https://github.com/official-stockfish/Stockfish/pull/4100> (besucht am 16.08.2022).
- [4] Murray S. Campbell und T.A. Marsland, „A comparison of minimax tree search algorithms“, *Artificial Intelligence*, Jg. 20, Nr. 4, S. 347–367, Juli 1983. DOI: 10.1016/0004-3702(83)90001-2.
- [5] David Levy, Hrsg., *Computer Chess Compendium*. Springer New York, 1988. DOI: 10.1007/978-1-4757-1968-0.