Neue Einsatzmöglichkeit von Hardwarebeschleunigern für nachhaltigere KI-Modelle: Entwicklung und Evaluation der Boltzmann Maschinen auf einem physikinspirierten Hardwarebeschleuniger

Bachelorarbeit

submitted on February 21, 2024

Fakultät Wirtschaft und Gesundheit

Studiengang Wirtschaftsinformatik

Kurs WWI2021F

von

Simon Spitzer

Betreuer in der Ausbildungsstätte:

⟨ Hewlett Packard GmbH ⟩
⟨ Dr. Fabian Böhm ⟩
⟨ Research Scientist at Hewlett Packard Labs⟩

DHBW Stuttgart:

⟨ Prof. Dr., Kai Holzweißig ⟩
⟨ der/des wissenschaftlichen Betreuerin/Prüferin ⟩

Unterschrift der Betreuerin/des Betreuers

Contents

List of abbreviations

Ein Abkürzungsverzeichnis ist optional. Das Paket `acronym` kann weit mehr, als hier gezeigt.[1] Beachten Sie allerdings, dass Sie die Einträge selbst in sortierter Reihenfolge angeben müssen.

BM Boltzmann Maschine

RBM Restriced Boltzmann Maschine

DNN Deep Neural Network

EBM Energy Based Model

Ergänzende Bemerkung: Eine im Text verwendete Abkürzung sollte bei ihrer ersten Verwendung erklärt werden. Falls Sie sich nicht selbst darum kümmern möchten, kann das das Paket `acronym` übernehmen und auch automatisch Links zum Abkürzungsverzeichnis hinzufügen. Dazu ist an allen Stellen, an denen die Abkürzung vorkommt, `\ac{ITIL}` zu schreiben.

Das Ergebnis sieht wie folgt aus:

- erstmalige Verwendung von `\ac{ITIL}` ergibt: **ITIL!** (**ITIL!**),
- weitere Verwendung von `\ac{ITIL}` ergibt: **ITIL!**

Wo benötigt, kann man mit dem Befehl `\acl{ITIL}` wieder die Langfassung ausgeben lassen: **ITIL!**.

Falls man die Abkürzungen durchgängig so handhabt, kann man durch Paket-Optionen (in `_dhbw_praeambel.tex`) erreichen, dass im Abkürzungsverzeichnis nur die tatsächlich verwendeten Quellen aufgeführt werden (Option: `printonlyused`) und zu jedem Eintrag die Seite der ersten Verwendung angegeben wird (Option: `withpage`).

[1] siehe `http://ctan.org/pkg/acronym`

List of Figures

List of Tables

1 Einleitung

1.1 Motivation

1.2 Problemstellung

In der Forschung und Entwicklung von Generativen KI-Modellen rückt die Rechengeschwindigkeit und Energieeffizienz zunehmend in den Fokus[2] Die Autor*innen von Open AI bestätigen, dass die Wachstumsrate von Machine-Learning-Modellen die Effizienzrate von Computerchips schon längst übertroffen hat. So verdoppeln sich jede 3-4 Monate der Rechenbedarf dieser Modelle jedoch verdoppeln sich nach Moore's Law die Leistung der Computerchips nur jede 2 Jahre.[3] Angesichts der Probleme des steigenden Energieverbrauchs von Rechenzentren und den damit verbundenen Treibhausgasemissionen dieser, ist die Suche nach effizienteren Lösungen essenziell für die Zukunft. Weltweit steigern Datenzentren ihren Energieverbrauch jährlich um 20-40%, wodurch sie 2022 etwa 1,3% des globalen Energieverbrauchs und 1% der energiebedingten globalen Treibhausgasemissionen verursacht haben.[4] Jedoch ist hier nicht zu erkennen, wie groß dabei der KI-Anteil zur Grundgesamtheit beiträgt.

Ein bereits bekannter Ansatz ist die Benutzung von KI-Beschleunigern basierend auf ASICs (Application-specific Integrated Circuits) - also Schaltungen, die anwendungsspezifisch verwendet werden, wie zum Beispiel Google TPUs (Tensor Processing Unit).[5] Dies ist auch sinnvoll, da die Verwendung von Mehrzweckmodellen für diskriminierende Aufgaben im Vergleich zu aufgabenspezifischen Modellen energieintensiver ist.[6] Ein alternatives vielversprechendes Konzept in der Forschung ist die Verwendung von physikinspirierten Hardwarebeschleunigern, die primär bei Optimierungsalgorithmen eingesetzt werden aufgrund ihrer Fähigkeit Probleme schneller und effizienter als GPUs lösen zu können.[7] Ein skalierbarer physikinspirierter Hardwarebeschleuniger (auch Ising-Maschine genannt), der die Leistung bestehender Standard-Digitalrechner übertrifft, könnte einen großen Einfluss auf praktische Anwendungen für eine Vielzahl von Optimierungsproblemen haben.[8]

Solche physikinspirierten Hardwarebeschleuniger bieten durch ihre besondere Berechnungsweise Potenzial für eine effizientere Verarbeitung von rechenintensiven Aufgaben. Konkret wird die Beschleunigung, anders als es bei digitalen Computern der Fall ist, durch die Berechnung rechenintensiver Aufgaben mit analogen Signalen erreicht. Die Implementierung auf dedizierter Hard-

[2] Vgl. Luccioni/Jernite/Strubell 2023, p. 1
[3] Vgl. Dario Amodei/Danny Hernandez 2024, p. 1
[4] Vgl. Hintemann/Hinterholzer 2022, p. 1
[5] Vgl. Wittpahl 2019, p. 39
[6] Vgl. Luccioni/Jernite/Strubell 2023, p. 5
[7] Vgl. Mohseni/McMahon/Byrnes 2022, p. 1
[8] Vgl. Mohseni/McMahon/Byrnes 2022, p. 1

ware bietet darüber hinaus die Möglichkeit, die Parallelisierung von digitalen Hardwarebeschleunigern und analogem Rechnen auszunutzen.[9]

Interessanterweise zeigen die Energiefunktionen von Hardwarebeschleunigern, die in Ising-Maschinen verwendet werden, große Parallelen zu denen in Boltzmann Maschinen, trotz ihrer unterschiedlichen Anwendungen, daher liegt es nahe, dass Ising Maschinen auch für KI gut funktionieren.[10] Ising-Maschinen zielen darauf ab, ihre Energie zu minimieren, wobei sie Energie als eine paarweise Interaktion von binären Variablen „Spins“ definieren.[11] BoltzmannMaschinen hingegen sind energiebasierte neuronale Netzwerke, die Klassifizierungen durchführen, indem sie jeder Konfiguration der Variablen eine skalare Energie zuordnen. Die Netzwerkenergie zu minimieren ist hierbei vergleichbar mit der Lösung des Optimierungsproblems.[12] Aktuelle Probleme mit Boltzmann-Maschinen umfassen die hohe Komplexität und Anforderungen an die All-to-All-Kommunikation zwischen Verarbeitungseinheiten, was ihre Implementierung auf herkömmlichen digitalen Computern ineffizient macht, sowie eine inhärent langsame Konvergenz in bestimmten Prozessen wie Simulated Annealing.[13] Diese Herausforderungen erschweren das Training und die Anwendung von Boltzmann-Maschinen insbesondere für große Datenmengen und komplexe Optimierungsaufgaben.[14] Nichtsdestotrotz impliziert die Ähnlichkeit der beiden, dass Ising-Maschinen in der Lage sein könnten, dieses spezielle KI-Modell, energieeffizienter und mit höherer Rechengeschwindigkeit auszuführen. Aktuell existieren nur wenige Konzepte eine Implementierung von Boltzmann Maschinen auf Ising-Maschinen zu erreichen. Das Paper der Autoren Mahdi Nazm BojnordiEngin und Engin Ipek ist hier ein vielversprechender Ansatz, jedoch konnte nicht gezeigt werden, wie es auf einem richtigen Beschleunigerchip funktionieren würde.

Vor diesem Hintergrund ergeben sich folgende zentrale Forschungsfragen:

1. Können Boltzmann Maschinen auf physikinspiriertenHardwarebeschleunigern durch analoge Rauschinjektion effizient implementiert werden?

    - Wie ist die Genauigkeit des KI-Modells im Hardwarebeschleuniger? Metrik: Prediction Accuracy

    - ergleichen mit anderen Hardwarebeschleuniger, FPGA, GPU oderCPU aus der Literatur (gute und schlechte) in Bezug auf Energieeffizienz und Rechengeschwindigkeit – Metriken: Troughput(Samples/Sec), Energieverbrauch (Energy/Operation)

Daher gilt es zu testen, ob dieses generative KI-Modell mit Ising Maschinen kompatibel ist und ob diese Lösung effizient ist oder nicht.

[9] Vgl. Mohseni/McMahon/Byrnes 2022, p. 4
[10] Vgl. Cai et al. 2019, p. 10
[11] Vgl. Wang/Roychowdhury 2017, p. 1
[12] Vgl. Nazm Bojnordi/Ipek 2016, p. 2
[13] Vgl. Nazm Bojnordi/Ipek 2016, p. 1
[14] Vgl. Nazm Bojnordi/Ipek 2016, p. 2

1.3 Zielsetzung(ohne gneaue Metriken nennen, generell halten)

Das primäre Ziel dieser Bachelorarbeit ist die Erforschung und Erweiterung eines bestehenden physikinspirierten Hardwarebeschleunigers (ISING Maschine) zur Implementierung und Evaluation von Boltzmann Maschinen, einem energiebasierten KI-Modell. Dabei sollen die aufgestellten Forschungsfragen beantwortet werden.

Hierzu ist es zu Beginn nötig eine Simulator Pipeline zu konstruieren mit der Boltzmann Maschinen auf dem Hardwarebeschleuniger übersetzt werden. Die Simulator Pipeline besteht dabei aus einer bestehender KI-Bibliothek und bestehenden Hardwarebeschleuniger, die miteinander verbunden werden. Mit der Simulator Pipeline soll gezeigt werden, dass der Hardwaresimulator die Boltzmann Maschinen umsetzen kann. Aus der Simulator Pipeline heraus werden die Aktivierungswahrscheinlichkeiten der einzelnen Neuronen auf der simulierten Hardware gemessen und bei Erfolg bis zu einem vollständigen Neuronalen Netzwerk erweitert. Finaler Schritt ist, dass der Hardwarebeschleuniger für Training und Interferenz genutzt werden kann und dabei vergleichbar mit herkömmlichen MLLibraries ist. Diese Phase umfasst die sorgfältige Anpassung und möglicherweise Erweiterung des bestehenden Beschleunigers, um die spezifischen Anforderungen der Boltzmann Maschinen zu erfüllen.

Wenn die Simulator Pipeline validiert werden kann, wird ein Workload auf ein Standarddatenset zur Handschrifterkennung getestet. Dabei werden die Prediction Accuracy, Troughput (Samples/Sec) und der Energieverbrauch (Energy/Operation) der Boltzmann Maschinen auf dem ISING Hardwarebeschleuniger untersucht und dadurch die aufgestellten Forschungsfragen beantwortet.

1.4 Forschungsmethodik

Design Science Research

1. **Problemorientierung:** DSR fokussiert auf die Lösung praktischer Probleme, wie die Forschung zur Steigerung der Effizienz und Rechengeschwindigkeit in KI-Modellen.
2. **Artefakt Entwicklung:** Zentral in DSR ist die Entwicklung innovativer Artefakte. Die Arbeit zielt darauf ab, ein solches Artefakt in Form des physikinspirierten Hardwarebeschleunigers weiterzuentwickeln und für KI-Modelle einzusetzen.
3. **Iterative Evaluation:** Durch die iterative Vorgehensweise in DSR kann die Ausarbeitung der Lösung fortlaufend verbessert und angepasst werden, was für die Entwicklung und Optimierung von KI-Systemen entscheidend ist (ebenfalls das Konzept).
4. **Beitrag zur Wissensbasis und Praxisrelevanz:** DSR unterstützt die Generierung neuer Erkenntnisse und stellt sicher, dass Forschungsergebnisse sowohl theoretisch fundiert

als auch praktisch anwendbar sind, was mit den Zielen Ihres Projekts im Einklang steht. Untermethodik könnte hierbei eine Simulation sein. Variabel, je nach Verlauf der Forschung.

1.5 Aufbau der Arbeit

2 Aktueller Stand der Forschung und Praxis (generell auch wiedergeben von aktuell existierenden Lösungsmustern)

2.1 Ressourcenverbrauch bei KI-Modellen

2.1.1 Ressourcenverbrauch bei KI-Modellen

Nachhaltigkeit

Stromverbrauch

Rechenleistung begrenzt, KI-Modelle wachsen schneller als verfügbare Leistung

2.2 Deep Neural Network - Boltzmann Maschinen (Erstmal DNN erklären generell)

(Erklären von Deep Neurol Network und Neurol Network) -> Anwendungsbereiche Spracherkennung, Image recognition Solche Deep Neural Networks, sind sehr ressourceneffizient und möglicher Forschungsbereich für Nutzung des Hardwarebeschleunigers Idee dabei, die repräsentationspower von energybased model höher als bei LLMs, mit weniger Neuronen besser als LLMs

Some regression tasks within computer vision in Deep Neural Network (DNN) include object detection, medical image registration, head- and body-pose estimation, age estimation and visual tracking.[15]

Energy Based Models -> Hinton Paper Gleichung 2, Warscheinlichkeitsbasierte Modelle Durch Gleichung 3 kann erklärt werden wieso es nicht berechenbar ist.

-> Dann auf BM kommen und erklären mit Training

->RBMs einführen und sagen warum Training vereinfacht ist

—

Warscheinlichkeiten und Energie sind gekoppelt, Modifiziere Energielandachaft oder Warscheinlichkeiten

Hintons Paper benutzen und ableiten —

[15] Vgl. Gustafsson et al. 2020, pp. 325–326

2.2.1 Energy-based models

An Energy Based Model (EBM) is a type of statistical model where the likelihood of a particular state is determined by an energy function.[16] Since 1982, those models have been continuously emerging in the machine learning field when J.J. Hopfield introduced the Hopfield Network.[17] Current developments include their use in reinforcement learning, potential replacements for discriminators in generative adversarial networks and for quantum EBMs.[18] The underlying idea behind EBMs is to establish a probabilistic physical system that is able to learn and memorize patterns but most importantly generalize it.[19] Specifically it involes learning an energy function $E_\theta(x) \in \mathbb{R}$ and assigning the low energy to the observed data x_i and high energy to other values$x$.[20]

EINFÜGEN VON EINER ENERGIELANDSCHAFT BILD

In this figure a simplified energy landscape is shown where the local minima correspond to states that encode an MNIST digit.[21] It is visible that observed data settles in the local minimum of the energy landscape, in this case a clear 0. On the other hand close to the local maxima of the energy landscape the 0 is only barely recognizable and therefore got a higher energy value assigned to it. The assumption of the underlying distribution function $P(x)$ is equal to the solution of the optimization problem:

$$P(x) = \frac{1}{Z} \exp\left(-\frac{E(x)}{T}\right), \tag{2.1}$$

where Z is given by the partition function to ensure that the density function normalizes to a total probability of 1 and T is interpreted as the temperature .[22] As a result the behavior of a EBM is determined by 2.2. The aim of the training is to match the real data P_{data} as closely as possible with the internal model P_{model}. A practical method to achieve this goal is to use the KL divergence. KL divergence is a mathematical equation that helps to meassure how close the predictions are by comparing the model's learned distribution to the true distribution of the data:

$$G = \sum_x P^+(x) \ln\left(\frac{P^+(x)}{P^-(x)}\right) \tag{2.2}$$

Here, $P^+(x)$ is the probability when the states are determined by a data input from the environm-net, while $P^-(x)$ represents the internal network running freely, also referred to as "dreaming".[23] To optimise the KL divergence, in this case G, the energy is adjusted, whereby data is assigned to low energy states (according to 2.1) and the training data receives high energy and therefore

[16] Vgl. Huembeli et al. 2022, p. 2
[17] Vgl. Hopfield 1982
[18] Vgl.Verdon et al. 2019, p. 1; Vgl.Du/Lin/Mordatch 2021, p. 1
[19] Vgl. Huembeli et al. 2022, p. 2
[20] Vgl. Gustafsson et al. 2020, p. 330
[21] Vgl. Huembeli et al. 2022, p. 6
[22] Vgl. Huembeli et al. 2022, pp. 2–3
[23] Vgl. Ackley/Hinton, G. E./Sejnowski, T. J. 1985, pp. 154–155

high probabiliies.[24] To complete the section the "partition function", Z, used in 2.1 is given by summing over all possible pairs of visible and hidden vectors:

$$Z = \sum_{x} \exp\left(-\frac{E(x)}{T}\right) \tag{2.3}$$

As a side note that is worth mentioning is, that using the maximum likelihood estimator for Z is impractical due to the requirement of summing over all possible states, which leads to an exponential increase in the number of states for larger systems.[25]

2.2.2 concept Boltzmann Maschine and usage of the model

A Boltzmann Maschine (BM) is a specific symmetrical EBM consisting of binary neurons $\{0, 1\}$.[26] The neurons of the network can be split into two functional groups, a set of visible neurons and a set of hidden neurons.[27] Therefore, the BM is a two-layer model with a visible layer ("v") and a hidden layer ("h").[28] The visible layer is the interface between the network and the environment. It receives data inputs during training and sets the state of a neuron to either $\{0, 1\}$ which represents activated or not activated. On the other hand, the hidden units are not connected to the environment and can be used to "explain" underlying constraints in the internal model of input vectors and they cannot be represented by pairwise constraints.[29] The connection between the individual neurons is referred to as bidirectional, as each neuron communicates with each other in both directions.[30] In the following figure 1, a general BM is depicted, where the upper layer embodies a vector of stochastic binary 'hidden' features, while the lower layer embodies a vector of stochastic binary 'visible' variables.[31]

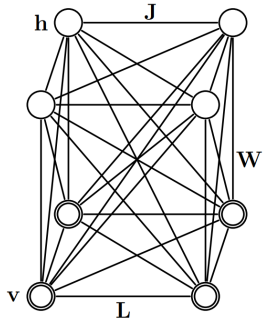

Abb. 1: figure of a general Boltzmann Machine

[24] Vgl. Zhai et al. 2016, pp. 2–3
[25] Vgl. Zhai et al. 2016, pp. 2–3
[26] Vgl. Amari/Kurata/Nagaoka 1992, p. 260
[27] Vgl. Ackley/Hinton, G. E./Sejnowski, T. J. 1985, p. 154
[28] Vgl. Salakhutdinov/Hinton, G. 2009, p. 448
[29] Vgl. Ackley/Hinton, G. E./Sejnowski, T. J. 1985, p. 154
[30] Vgl. Ackley/Hinton, G. E./Sejnowski, T. J. 1985, p. 149
[31] Vgl. Salakhutdinov/Hinton, G. 2009, p. 449

It contains a set of visible units $v \in \{0,1\}$, and a set of hidden units $h \in \{0,1\}$ (see Fig. 1). The energy function of the BM with the states $\{v,h\}$ is defined as:

$$E(v,h;\theta) = -\frac{1}{2}v^T Lv - \frac{1}{2}h^T Jh - v^T Wh, \tag{2.4}$$

where $\theta = \{W,L,J\}$ are the model parameters.[32] W,L,J represent visible-to-hidden, visible-to-visible and hidden-to-hidden weights. The individual neurons can be made to try to minimize the global energy by setting the right assumptions.[33] Entering a particular input to the machine, the system will find the minimum energy configuration that can illustrate the input.[34] A simple method to find a local energy minimum is to switch into wichever of the two states of a neuron hold the lower energy given the current state of the other neurons.[35] The exact reason for this is the following: "If all the connection strengths are symmetrical, which is typically the case for constraint satisfaction problems, each unit can compute its effect on the total energy from information that is locally available."[36] By inserting the function 2.4 into the earlier introduced KL-divergence 2.2 and doing gradient descend the following learning rule to update the weights and biases results[37]:

$$\Delta w_{ij} = \epsilon(\langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}}) \tag{2.5}$$

— Deep Boltzmnn Machines, Kann nicht trainiert werden da exponentiell

—

The network can now update the weights "W" that exist between the neurons through the training rule based on the observations that served as input.[38]

Performing exact maximum likelihood learning in this model is intractable because exact computation of the data predictions and the model predictions takes a time that is exponential in the number of hidden units.[39]

As early as 1985, one of the founding fathers of artificial intelligence, "Geoffrey Hinton", was aware that an BM is able to learn its underlying features by looking at data from a domain and developing a generative internal model.[40] In the next step, it is possible to generate examples with the same probability distribution as the examples shown.

[32] Vgl. Salakhutdinov/Hinton, G. 2009, p. 448
[33] Vgl. Ackley/Hinton, G. E./Sejnowski, T. J. 1985, p. 150
[34] Vgl. Ackley/Hinton, G. E./Sejnowski, T. J. 1985, p. 150
[35] Vgl. Fahlman/Hinton, G./Sejnowski, T. 1983, p. 110
[36] Fahlman/Hinton, G./Sejnowski, T. 1983, p. 110
[37] Vgl. Hinton, G. E. 2012b, p. 5
[38] Vgl. Barra et al. 2012, pp. 1–2
[39] Vgl. Salakhutdinov/Hinton, G. 2009, p. 449
[40] Vgl. Ackley/Hinton, G. E./Sejnowski, T. J. 1985, p. 148

2.2.3 Energiefunktion, Training von BMs

If the diagonal elements L and J of the general BM introduced earlier, are set to 0 the known model of a RBM establishes shown in fig.2.[41]

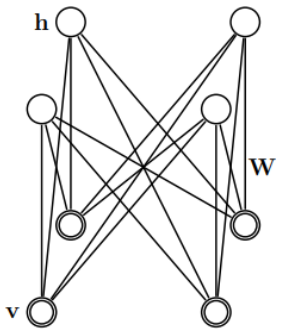

Abb. 2: Figure of a RBM

As a result no more visible-to-visible and hidden-to-hidden connections can be found in the network. The configuration of the visible and hidden units (v, h) has an energy (Hopfield, 1982) given by:

$$E(v,h) = -\sum_{i \in \text{visible}} a_i v_i - \sum_{j \in \text{hidden}} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \tag{2.6}$$

where v_i, h_j are the binary states of visible unit i and hidden unit j, a_i, b_j are their biases and w_{ij} is the weight between them.[42]

2.2.4 Restriced Boltzmann Machine

Markov-Chain-Monte-Carlo-Verfahren

Metropolis Hastings, Conrtrastive Divergence

2.2.5 Aktuelle Probleme mit RBM/BM

Exact maximum likelihood learning in the Boltzmann machine is infeasible due to the exponentially increasing computation time with the number of hidden units. Hinton and Sejnowski's 1983 algorithm approximates this via Gibbs sampling, but it is limited by the significant time needed to reach the stationary distribution in a complex, multimodal energy landscape.

[41] Vgl. Salakhutdinov/Hinton, G. 2009, p. 449
[42] Vgl. Hinton, G. E. 2012a, pp. 3–4

2.3 Hardwarebeschleuniger

2.3.1 Aktuelle Ansätze im Bereich KI und weitere Lösungen

Asics

Quantencomputing

2.3.2 ISING Maschine/ Physikinspirierter Hardwarebeschleuniger

Konzept (mit Energiefunktion), Probleme der Digitalrechner bzw. Unterschied zu Digitalrechner

Aktuelle Anwendung

Potentielle Einsatzgebiete für KI-Modelle

Parallelen Energiefunktion BM und ISING Maschine

2.4 Memristor Hopfield Network

2.4.1 Memristor

2.4.2 Hopfield Network

2.4.3 Crossbar

2.4.4 Output Hopfield Networtk

2.4.5 Noisy HNN

3 Zielspezifikation und Darlegung der Forschungsmethodik

3.1 Zielspezifikation (genauer als in Einleitung, Metriken erwähnen, Erfolg meiner Methode bewerten, Welcher Teil der Forschungsfrage wird beantwortet?)

3.2 Design Science Research

3.3 Zielsetzung(ohne gneaue Metriken nennen, generell halten)

3.4 Laborexperiment für die Umsetzung

4 Implementierung/Laborexperiment der Simulator Pipeline

4.1 Zielsetzung und Forschungsmethodik

4.2 Aufbau der Simulator Pipeline

4.3 KI-Bibliothek Scikit-Learn

5 Evaluation der BM auf dem physikinspiriertem Hardwarebeschleuniger

5.1 Zielsetzung und Forschungsmethodik

5.1.1 Prediction Accuracy

5.1.2 Troughput (Samples/Sec)

5.1.3 Energieverbrauch (Energy/Operation)

5.2 Vergleichen mit anderen Hardwarebeschleuniger, FPGA, GPU oder CPU aus der Literatur

6 Kritische Reflexion und Ausblick

6.1 Evaluation der Erkenntnisse in Bezug auf die Zielsetzung der Arbeit

6.2 Kritische Reflexion der Ergebnisse und Methodik

6.3 Zielsetzung(ohne gneaue Metriken nennen, generell halten)

6.4 Ergebnisextration für Theorie und Praxis (evtl. mit 6.4 Zusammenlegen)

6.5 Ausblick

Appendix

List of appendices

Appendix 1: So funktioniert's

Um den Anforderungen der Zitierrichtlinien nachzukommen, wird das Paket `tocloft` verwendet. Jeder Anhang wird mit dem (neu definierten) Befehl **anhang**{Bezeichnung} begonnen, der insbesondere dafür sorgt, dass ein Eintrag im Anhangsverzeichnis erzeugt wird. Manchmal ist es wünschenswert, auch einen Anhang noch weiter zu unterteilen. Hierfür wurde der Befehl **anhangteil**{Bezeichnung} definiert.

In Anhang 1/1 finden Sie eine bekannte Abbildung und etwas Source Code in **??**.

Anhang 1/1: Wieder mal eine Abbildung

Abb. 3: Mal wieder das DHBW-Logo.

List of references

Ackley, D. H./Hinton, G. E./Sejnowski, T. J. (1985): A Learning Algorithm for Boltzmann Machines. In: *Cognitive Science* 9.1, pp. 147–169. ISSN: 0364-0213. DOI: 10.1016/S0364-0213(85)80012-4. URL: https://www.sciencedirect.com/science/article/pii/S0364021385800124 (retrieval: 02/16/2024).

Amari, S./Kurata, K./Nagaoka, H. (1992): Information Geometry of Boltzmann Machines. In: *IEEE Transactions on Neural Networks* 3.2, pp. 260–271. ISSN: 1941-0093. DOI: 10.1109/72.125867. URL: https://ieeexplore.ieee.org/abstract/document/125867 (retrieval: 02/16/2024).

Barra, A./Bernacchia, A./Santucci, E./Contucci, P. (2012): On the Equivalence of Hopfield Networks and Boltzmann Machines. In: *Neural Networks* 34, pp. 1–9. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2012.06.003. URL: https://www.sciencedirect.com/science/article/pii/S0893608012001608 (retrieval: 02/16/2024).

Cai, F./Kumar, S./Van Vaerenbergh, T./Liu, R./Li, C./Yu, S./Xia, Q./Yang, J. J./Beausoleil, R./Lu, W./Strachan, J. P. (2019): Harnessing Intrinsic Noise in Memristor Hopfield Neural Networks for Combinatorial Optimization. DOI: 10.48550/arXiv.1903.11194. arXiv: 1903.11194 [cs]. URL: http://arxiv.org/abs/1903.11194 (retrieval: 02/15/2024). preprint.

Dario Amodei/Danny Hernandez (2024): AI and Compute. URL: https://openai.com/research/ai-and-compute (retrieval: 02/15/2024).

Du, Y./Lin, T./Mordatch, I. (2021): Model Based Planning with Energy Based Models. DOI: 10.48550/arXiv.1909.06878. arXiv: 1909.06878 [cs, stat]. URL: http://arxiv.org/abs/1909.06878 (retrieval: 02/19/2024). preprint.

Fahlman, S./Hinton, G./Sejnowski, T. (1983): Massively Parallel Architectures for AI: NETL, Thistle, and Boltzmann Machines., p. 113. 109 pp.

Gustafsson, F. K./Danelljan, M./Bhat, G./Schön, T. B. (2020): Energy-Based Models for Deep Probabilistic Regression. In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi/Horst Bischof/Thomas Brox/Jan-Michael Frahm. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 325–343. ISBN: 978-3-030-58565-5. DOI: 10.1007/978-3-030-58565-5_20.

Hintemann, R./Hinterholzer, S. (2022): Data Centers 2021: Data Center Boom in Germany Continues - Cloud Computing Drives the Growth of the Data Center Industry and Its Energy Consumption. DOI: 10.13140/RG.2.2.31826.43207.

Hinton, G. E. (2012a): 'A Practical Guide to Training Restricted Boltzmann Machines'. In: *Neural Networks: Tricks of the Trade: Second Edition*. Ed. by Grégoire Montavon/Geneviève B. Orr/Klaus-Robert Müller. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 599–619. ISBN: 978-3-642-35289-8. DOI: 10.1007/978-3-642-35289-8_32. URL: https://doi.org/10.1007/978-3-642-35289-8_32 (retrieval: 02/15/2024).

– **(2012b)**: 'A Practical Guide to Training Restricted Boltzmann Machines'. In: *Neural Networks: Tricks of the Trade*. Ed. by Grégoire Montavon/Geneviève B. Orr/Klaus-Robert Müller.

Vol. 7700. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 599–619. ISBN: 978-3-642-35288-1 978-3-642-35289-8. DOI: 10.1007/978-3-642-35289-8_32. URL: http://link.springer.com/10.1007/978-3-642-35289-8_32 (retrieval: 02/15/2024).

Hopfield, J. J. (1982): Neural Networks and Physical Systems with Emergent Collective Computational Abilities. In: *Proceedings of the National Academy of Sciences* 79.8, pp. 2554–2558. DOI: 10.1073/pnas.79.8.2554. URL: https://www.pnas.org/doi/10.1073/pnas.79.8.2554 (retrieval: 02/19/2024).

Huembeli, P./Arrazola, J. M./Killoran, N./Mohseni, M./Wittek, P. (2022): The Physics of Energy-Based Models. In: *Quantum Machine Intelligence* 4.1, p. 1. ISSN: 2524-4914. DOI: 10.1007/s42484-021-00057-7. URL: https://doi.org/10.1007/s42484-021-00057-7 (retrieval: 02/19/2024).

Luccioni, A. S./Jernite, Y./Strubell, E. (2023): Power Hungry Processing: Watts Driving the Cost of AI Deployment? DOI: 10.48550/arXiv.2311.16863. arXiv: 2311.16863 [cs]. URL: http://arxiv.org/abs/2311.16863 (retrieval: 02/15/2024). preprint.

Mohseni, N./McMahon, P. L./Byrnes, T. (2022): Ising Machines as Hardware Solvers of Combinatorial Optimization Problems. DOI: 10.48550/arXiv.2204.00276. arXiv: 2204.00276 [physics, physics:quant-ph]. URL: http://arxiv.org/abs/2204.00276 (retrieval: 02/15/2024). preprint.

Nazm Bojnordi, M./Ipek, E. (2016): Memristive Boltzmann Machine: A Hardware Accelerator for Combinatorial Optimization and Deep Learning, p. 13. 1 p. DOI: 10.1109/HPCA.2016.7446049.

Salakhutdinov, R./Hinton, G. (2009): Deep Boltzmann Machines. In: *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*. Artificial Intelligence and Statistics. PMLR, pp. 448–455. URL: https://proceedings.mlr.press/v5/salakhutdinov09a.html (retrieval: 02/16/2024).

Verdon, G./Marks, J./Nanda, S./Leichenauer, S./Hidary, J. (2019): Quantum Hamiltonian-Based Models and the Variational Quantum Thermalizer Algorithm. arXiv: 1910.02071 [quant-ph]. URL: http://arxiv.org/abs/1910.02071 (retrieval: 02/19/2024). preprint.

Wang, T./Roychowdhury, J. (2017): Oscillator-Based Ising Machine. DOI: 10.48550/arXiv.1709.08102. arXiv: 1709.08102 [physics]. URL: http://arxiv.org/abs/1709.08102 (retrieval: 02/15/2024). preprint.

Wittpahl, V., ed. **(2019)**: Künstliche Intelligenz: Technologie | Anwendung | Gesellschaft. Berlin, Heidelberg: Springer. ISBN: 978-3-662-58041-7 978-3-662-58042-4. DOI: 10.1007/978-3-662-58042-4. URL: http://link.springer.com/10.1007/978-3-662-58042-4 (retrieval: 02/15/2024).

Zhai, S./Cheng, Y./Lu, W./Zhang, Z. (2016): Deep Structured Energy Based Models for Anomaly Detection. In: *Proceedings of The 33rd International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, pp. 1100–1109. URL: https://proceedings.mlr.press/v48/zhai16.html (retrieval: 02/19/2024).

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit mit dem Thema: *Mein Titel* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

(Ort, Datum) (Unterschrift)