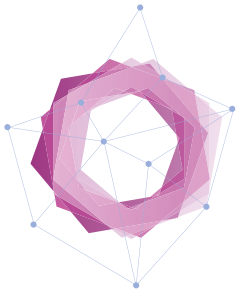
dena
Deutsche Energie-Agentur



Future Energy
Lab

STUDIE

Energieeffiziente künstliche Intelligenz für eine klimafreundliche Zukunft

Neue Erkenntnisse über Energieeinsparpotenziale bei KI-Anwendungen

Impressum

Herausgeber:
Deutsche Energie-Agentur GmbH (dena)
Chausseestraße 128 a
10115 Berlin

Tel.: +49 30 66 777- 0
Fax: +49 30 66 777- 699

E-Mail: info@dena.de
Internet: www.dena.de

Autorinnen und Autoren:
Fabian Seiter, dena
Hendrik Zimmermann, dena
Linda Babilon, dena
Philipp Richard, dena
Karsten Müller, Fraunhofer HHI
Wojciech Samek, Fraunhofer HHI
Benno Stabernack, Fraunhofer HHI
Fritjof Steinert, Fraunhofer HHI

Gestaltung:
The Ad Store GmbH

Stand:
Januar 2024

Alle Rechte sind vorbehalten. Die Nutzung steht unter dem Zustimmungsvorbehalt der dena.

Bitte zitieren als:
Deutsche Energie-Agentur (dena) (Hrsg.) (2023): Studie: Energieeffiziente künstliche Intelligenz für eine klimafreundliche Zukunft. Neue Erkenntnisse über Energieeinsparpotenziale bei KI-Anwendungen.

Bundesministerium für Wirtschaft und Klimaschutz

Die Veröffentlichung dieser Publikation erfolgt im Auftrag des Bundesministeriums für Wirtschaft und Klimaschutz. Die Deutsche Energie-Agentur GmbH (dena) unterstützt die Bundesregierung in verschiedenen Projekten zur Umsetzung der energie- und klimapolitischen Ziele im Rahmen der Energiewende.

Inhaltsverzeichnis

Inhaltsverzeichnis		**3**
Vorwort		**4**
Kurzfassung und Handlungsempfehlungen		**5**
1	**Einleitung**	**8**
1.1	Stromverbrauch durch KI-Anwendungen	9
1.2	Einführung in künstliche neuronale Netze	10
2	**Energieeffiziente Ausführung von KI-Modellen**	**12**
2.1	Alternative Methoden der Integration von Rechenbeschleunigern	17
2.2	Konzeption einer Rechnerarchitektur für energieeffiziente Inferenz	17
2.3	Implementierung der NAA-Rechnerarchitektur für energieeffiziente Inferenz	19
2.4	Spezifikation der Key-Performance-Indikatoren	20
2.5	FPGA-NAA-Testaufbau	20
2.6	Messkonzept für NAA-Beschleuniger	23
2.7	Energiemessung	23
2.8	Webbasierter Demonstrator	25
3	**Energieeffizienz bei der Übertragung von KI-Modellen**	**27**
3.1	Direkte Übertragung neuronaler Netze	28
3.2	NNC (Neural Network Coding)	29
3.2.1	Parameterreduktion	30
3.2.2	Quantisierung	31
3.2.3	Entropiecodierung	32
3.3	Föderiertes Lernen als Anwendungsfall zur systematischen Untersuchung	32
3.4	Untersuchungsergebnisse bei der Übertragung neuronaler Netze	34
3.4.1	Testbeschreibung und Detailanalyse der Energieeinsparung mittels NNC bei einmaliger Übertragung	34
3.4.2	Testergebnisse für das Modell ResNet18 auf dem Datensatz CIFAR10	40
3.4.3	Testergebnisse für das Modell ResNet152 auf dem Datensatz CIFAR10	43
3.5	Webbasierter Demonstrator	45
4	**Weiterführende Energieeffizienzpotenziale**	**47**
4.1	Weiterführende modulare Hardware-Optimierung für alle KI-Prozesse	48
4.2	KI-Modell-Recycling	48
4.3	Transferlernen (als selektives KI-Modell-Recycling)	49
4.4	KI-Modelloptimierung und Energieeffizienz-Klassifizierung	50
5	**Handlungsempfehlungen**	**51**
5.1	Handlungsempfehlungen auf Basis der neuen Hardwarearchitektur für den Einsatz in Rechenzentren:	52
5.2	Handlungsempfehlungen auf Basis der Kompression der Übertragung beim föderierten Lernen	52
6	**Fazit**	**53**
Abbildungsverzeichnis		**55**
Tabellenverzeichnis		**56**
Literaturverzeichnis		**57**
Abkürzungsverzeichnis		**59**

Vorwort

Künstliche Intelligenz (KI) ist weltweit auf dem Vormarsch und kommt sektorenübergreifend zum Einsatz: Einer Umfrage des Ifo-Instituts zufolge beschäftigen sich mehr als 60 Prozent der Unternehmen in Deutschland mit Anwendungsmöglichkeiten im eigenen Betrieb (Schaller, 2023). Spätestens seit der Veröffentlichung von ChatGPT ist KI auch in der allgemeinen Öffentlichkeit mehr als nur eine abstrakte Zukunftstechnologie: So geben in einer repräsentativen Umfrage ein Fünftel der Teilnehmerinnen und Teilnehmer an, die textbasierte KI von OpenAI bereits ausprobiert zu haben und mehr als 80 Prozent haben von ihr gehört (Bitkom, 2023). Auch in der Energiebranche vollzieht sich die Transformation vom „KI-Potenzial“ zur „KI-Wirklichkeit“ und KI kommt vermehrt zum Einsatz, so zum Beispiel bei der Wartung von Anlagen und Maschinen (Predictive Maintenance) oder in der Kraftwerkseinsatzplanung. Dabei wird besonders die Chance hervorgehoben, KI effektiv im Kampf gegen den Klimawandel zu nutzen und damit die Energiewende zu beschleunigen. Konkrete Anwendungen reichen von verbesserten Wetterprognosen bis hin zu KI-gesteuerten Drohnen zur Stromnetzüberwachung. Ein Aspekt, der in der öffentlichen Diskussion allerdings wenig Beachtung findet, ist der Stromverbrauch von KI selbst.

Die Deutsche Energie-Agentur (dena) hat die Aufgabe, die Energiewende in Deutschland als Ideengeber und Umsetzungspartner voranzubringen. Das Future Energy Lab ist die Antwort der dena auf die Notwendigkeit einer digitalisierten Energiewende: Ohne digitale Technologien ist eine vollständige Umstellung der Energieversorgung auf erneuerbare Energien nicht möglich. Im Future Energy Lab entstehen neue Ideen in Zusammenarbeit mit innovativen Start-ups, Stakeholder aus Politik, Wirtschaft und Wissenschaft werden zusammengebracht, um neue digitale Trends zügig für die Energiewende zu verwerten, und es werden zukunftsweisende Pilotprojekte auf den Weg gebracht. Dabei stellt die Nutzbarmachung von KI für die Energiewende einen wesentlichen Tätigkeitsschwerpunkt dar. Der Fokus liegt aber nicht einseitig auf Anwendungsmöglichkeiten von KI und digitalen Technologien im Allgemeinen, sondern ebenso auf der Klimabelastung, die durch die Nutzung selbst entsteht. Denn nur eine ganzheitliche Betrachtung, die die vollständigen hardware- und softwareseitigen Lebenszyklusemissionen berücksichtigt, garantiert einen Mehrwert für die Energiewende und damit Klima- und Menschenschutz.

Das Projekt „Energieeffiziente künstliche Intelligenz“ (EEKI) steht beispielhaft für diesen Ansatz: Das Ziel lautete, hardware- und softwareseitig Energieeinsparpotenziale für KI-Anwendungen zu identifizieren und experimentell zu bestätigen. Mit dem Fraunhofer Heinrich-Hertz-Institut (HHI) wurde ein kompetenter Partner für die Versuchsdurchführung gefunden, die mit ausgesprochen positiven Ergebnissen abgeschlossen werden konnte.

Das Projekt wurde im Auftrag des Bundesministeriums für Wirtschaft und Klimaschutz (BMWK) durchgeführt.

Hendrik Zimmermann

Fabian Seiter

Kurzfassung und Handlungsempfehlungen

Künstliche Intelligenz (KI) verbreitet sich rasant: KI-basierte Systeme kommen in immer mehr Anwendungsfeldern zum Einsatz und unterstützen zunehmend Entscheidungsprozesse in der Energiewirtschaft, zum Beispiel für Bedarfsprognosen, die prädiktive Wartung, die Erstellung von Kundenclustern oder die Abwehr von Cyberangriffen (Kratochwill et al., 2020). Für eine erfolgreiche Anwendung müssen diese komplexen Systeme mit Millionen von Beispieldaten „angelernt" werden. Dieser rechenintensive Prozess wird üblicherweise in großen Rechenzentren unter erheblichem Energieverbrauch durchgeführt. Bereits 2019 haben Strubel et al. (Strubell, Ganesh & McCallum, 2019) gezeigt, dass der CO_2-Fußabdruck eines State-of-the-Art-KI-Modells mit dem CO_2-Ausstoß vergleichbar ist, den eine Person mit über 300 Hin- und Rückflügen zwischen San Francisco und New York verursacht. Die Entwicklung zu immer komplexeren KI-Modellen hat sich zudem in den letzten Jahren insbesondere durch die generativen prädiktiven Transformer-Modelle (GPT – Generative Predictive Transformer) (OpenAI – Models, abgerufen am 11.09.2023) beschleunigt, die beispielsweise in ChatGPT eingesetzt werden. Bei diesen Modellen hat sich die Anzahl trainierbarer Parameter noch einmal vertausendfacht, was mit einer entsprechenden Erhöhung des Energieverbrauchs einhergeht. Beim verteilten Training – einer Technik des maschinellen Lernens (ML) von KI-Modellen – entsteht zusätzlich ein nicht zu vernachlässigender Kommunikationsaufwand, der die Energiebilanz weiter verschlechtert.

Im Sinne der Klimaziele der Bundesregierung ist es neben der Umstellung auf eine auf erneuerbaren Energien basierte Energieversorgung wichtig, den Stromverbrauch digitaler Technologien und Anwendungen zu senken, insbesondere vor dem Hintergrund ihrer wachsenden Größe und zunehmenden weltweiten Anwendung.

Diesem Ziel diente das vom Future Energy Lab der Deutschen Energie-Agentur (dena) durchgeführte und vom Bundesministerium für Wirtschaft und Klimaschutz (BMWK) beauftragte Projekt „Energieeffiziente künstliche Intelligenz".

Konkret sollten auf zwei Wegen Energieeinsparungen erreicht werden:

1. Durch eine verbesserte Modellübertragung beim sogenannten „Föderierten Lernen" (FL – Federated Learning), einem neuartigen Trainingsprozess für KI-Anwendungen. Dabei werden KI-Modelle nicht auf einem zentralen Server trainiert, sondern dezentral auf vielen verschiedenen Geräten. Dies hat den Vorteil, dass die Daten zum Modelltraining nicht übertragen (versandt) werden müssen. Stattdessen werden die Modelle übertragen und später zu einem großen Modell zusammengeführt. Dieser Ansatz ist besonders datenschutzfreundlich. Ziel war, die Energieeffizienz durch neueste Kompressionsmethoden bei der Modellübertragung zu steigern: Dabei werden die KI-Modelle vor der Übertragung komprimiert und nach der Übertragung dekomprimiert, was zu einem geringeren Stromverbrauch führt.

Ergebnis: Durch diesen Ansatz konnten **Energieeinsparungen von bis zu 65 Prozent** gegenüber Referenzverfahren gemessen werden, bei denen keine Komprimierung stattfindet.

2. Durch die Entwicklung einer neuen Hardware-Architektur für die Anwendung austrainierter KI-Modelle. Dabei wurden spezialisierte Rechenbeschleuniger, sogenannte Field Programmable Gate Arrays (FPGAs), so verschaltet, dass für ihre Ansteuerung weniger Rechenleistung benötigt wird. FPGAs sind Computerchips mit der Besonderheit, dass sich ihre elektrischen Schaltkreise im Gegensatz zu herkömmlichen Chips umprogrammieren lassen. Dadurch können sie sehr individuell an die jeweils auszuführende Aufgabe angepasst werden und kommen unter anderem für die Ausführung von KI-Anwendungen in Rechenzentren zum Einsatz.

Ergebnis: Durch diesen Ansatz konnten **Energieeinsparungen von bis zu 31 Prozent** gegenüber herkömmlichen Methoden gemessen werden.

Damit konnte das Future Energy Lab in Zusammenarbeit mit den Projektpartnern vom Fraunhofer Heinrich-Hertz-Institut erstmalig zeigen, dass beim Training und bei der Ausführung der hier untersuchten KI-Anwendungen erhebliche Stromeinsparungen möglich sind. Aus den Ergebnissen lassen sich wesentliche Handlungsempfehlungen ableiten, die in Kapitel 5 hergeleitet werden und im Folgenden zusammengefasst sind:

Handlungsempfehlungen auf Basis der neuen Hardware-Architektur für den Einsatz in Rechenzentren:

- Die öffentliche und private Forschung auf dem Gebiet der Energieeffizienz von Hardware-Architektur sollte weiter vorangetrieben und gefördert werden.
- Die in diesem Projekt unter Laborbedingungen getesteten Methoden sollten in der Praxis eingesetzt werden, um die Ergebnisse im Feld zu bestätigen.
- Rechenzentren werden in Zukunft wesentlich stärker auf bestimmte Rechenoperationen spezialisiert sein bzw. sich spezialisieren müssen, beispielsweise auf die Ausführung von bildverarbeitenden KI-Anwendungen. Vor dem Hintergrund der energieeffizienten Ausführung von Rechenoperationen wird empfohlen, frühzeitig Anreize für diese Spezialisierung zu schaffen.
- Der Anreiz zum Einbau energiesparender Hardware sollte durch weitere, schärfere Effizienzvorgaben bzw. -anreize bei neuen Rechenzentren oder für neu eingebaute Hardware-Komponenten gefördert werden.
- Die Förderung der Forschung zu energieeffizienten KI-Anwendungen ist auch dann sinnvoll, wenn diese

Anwendungen netto für mehr Energieeinsparung als -verbrauch sorgen. Denn der Verbrauch der Anwendung selbst bleibt immer positiv. Je mehr er gesenkt werden kann, desto besser. Selbst wenn in Zukunft die Energieversorgung zu 100 Prozent erneuerbar sein sollte, sind Energieeffizienzsteigerungen nach wie vor sinnvoll.

Handlungsempfehlungen auf Basis der Kompression der Übertragung beim föderierten Lernen:

- Der im Projekt entwickelte Ansatz für eine energieeffizientere Datenübertragung beim föderierten Lernen sollte zügig breitflächig eingesetzt werden, damit er sich idealerweise parallel mit dem föderierten Lernen verbreitet.

Übergreifende Handlungsempfehlungen:

- Die im Projekt ermittelten Ergebnisse können Ausgangspunkt für die Entwicklung eines Energieeffizienzstandards für KI-Modelle sein – hardware- wie softwareseitig. Entsprechende Standards, beispielsweise analog zu den Effizienzlabels für Elektrogeräte, bieten das Potenzial, eine wesentlich klimafreundlichere KI-Entwicklung und -Ausführung zu schaffen.

Im Folgenden wird in das Thema „Energieeffiziente KI“ sowie in die Untersuchungsmethoden eingeführt. Anschließend werden für beide Methoden Versuchsaufbau und -durchführung sowie die Ergebnisse im Detail vorgestellt. Im letzten Teil des Berichts sind weiterführende Forschungs- und Entwicklungsthemen dargestellt, die auf Basis der hier erhaltenen Ergebnisse zu weiteren Energieeinsparungen führen können, und es werden detaillierte Empfehlungen gegeben, in welche Richtung auf Basis der Ergebnisse weiter geforscht werden kann.

1. Einleitung

Künstliche Intelligenz (KI) bleibt weiterhin unangefochten eines der Topthemen der Digitalisierung und bietet großes Potenzial für die Energiewende (Deutsche Energie-Agentur (dena), 2023). Gleichzeitig verbraucht sie bereits heute große Mengen Strom, die in Anbetracht zunehmender Nutzung und immer komplexerer Modelle signifikant steigen werden. Dies stellt einen Zielkonflikt dar. Selbst wenn der Nettonutzen von KI-Anwendungen überwiegt, also wenn die erreichten Einspareffekte größer sind als der durch die Verwendung entstehende Verbrauch, wird ein Nettoverbrauch übrig bleiben. Diesen so weit wie möglich zu reduzieren, ist im Sinne der Energiewende und einer klimaneutralen Digitalisierung.

Es ist relativ komplex, den Energieverbrauch von KI-Anwendungen und im weiteren Sinne die Klima- und Umweltbelastung, was den Ressourcenverbrauch, die Entsorgung von Hardware etc. mit einbezieht, zu quantifizieren. Das Projekt „Energieeffiziente künstliche Intelligenz“ (EEKI) und die durchgeführten Untersuchungen beziehen sich ausschließlich auf den Stromverbrauch während des Trainings und der Ausführung bestimmter KI-Modelle. Eine ganzheitliche Betrachtung, die die gesamte Klima- und Umweltbelastung beispielsweise durch die eingesetzte Hardware mit berücksichtigt, wird nicht vorgenommen, ist aber für eine abschließende Beurteilung der Klimabelastung durch KI-Anwendungen natürlich notwendig. Der Stromverbrauch von KI-Anwendungen und die Bewertung der Klimabelastung durch den Verbrauch hängen von verschiedenen Faktoren ab, wie im Folgenden dargestellt.

1.1 Stromverbrauch durch KI-Anwendungen

Anzahl der Nutzerinnen und Nutzer einer KI-Anwendung

Eine Studie der University of Massachusetts Amherst (Strubell, Ganesh & McCallum, 2019) kam zu dem Ergebnis, dass das Training eines einzigen neuronalen Netzes einen CO_2-Ausstoß von fünf Autos während ihrer gesamten Lebenszeit inklusive des dabei verbrauchten Kraftstoffs verursacht. Zur Bewertung der Klimabelastung muss neben dem absoluten Verbrauch unter anderem die Anzahl der Nutzerinnen und Nutzer des KI-Modells berücksichtigt werden. Im Falle der Fahrzeuge verteilt sich der CO_2-Ausstoß auf möglicherweise 5 bis 20 Personen, im Falle der KI unter Umständen auf mehrere Hunderttausend. Dieses Beispiel verdeutlicht, dass die Bewertung der Klimabelastung durch KI-Anwendungen komplex ist.

Einsatzweck der KI-Anwendung

Neben dem absoluten Energieverbrauch und dem Energieverbrauch pro Nutzerin und Nutzer muss für ein vollständiges Bild der Nutzen der KI-Anwendung in die Betrachtung einbezogen werden. Thorsten Staake, Professor für Wirtschaftsinformatik an der Universität Bamberg und Spezialist für energieeffiziente Systeme, gibt hier eine Einordnung: „Wenn, als Beispiel, für das Training eines neuronalen Netzes für einen Spurhalte-Assistenten für eine ganze Fahrzeugflotte eines großen Herstellers Energie in der Höhe des Verbrauchs eines Pkw aufgewendet wird und damit auch nur ein Unfall verhindert wird, wäre das energetisch schon kompensiert – vom zusätzlichen Wert für die Unfallbeteiligten ganz abgesehen“ (Lobe, 2019). Dies ist ein Beispiel dafür, wie eine KI-Anwendung den Energieverbrauch, der mit Training und Ausführung assoziiert ist, überkompensiert, indem durch ihren Einsatz Energie und indirekt CO_2 gespart wird.

KI-bedingter Strombedarf in Rechenzentren

Der Strombedarf für Rechenzentren betrug bereits im Jahr 2022 mit 17,9 TWh (Borderstep, abgerufen am 15.09.2023) 3,74 Prozent des gesamten Stromverbrauchs der Bundesrepublik Deutschland (insgesamt 484 TWh laut Bundesnetzagentur). Zu diesem Verbrauch in Rechenzentren trugen im Jahr 2022 weit über 2 Millionen Server mit 7,8 TWh bei. Dies entspricht bei der Nutzung des deutschen Strommix für 2022 einem Ausstoß von 3,4 Millionen Tonnen CO_2 (Bitkom, abgerufen am 15.09.2023). Mit 17,9 TWh Stromverbrauch in Rechenzentren im Jahr 2022 wurde damit bereits mehr Strom verbraucht, als noch vor einigen Jahren für das Jahr 2025 mit 16,4 TWh prognostiziert wurde (Hintemann & Clausen, 2016). Wie erwähnt, waren KI-Anwendungen schon in den letzten Jahren mit ihren Millionen benötigten Einzelberechnungen sehr rechenintensiv, wobei durch die kürzlich entwickelten Modelle wie GPT (OpenAI – Models, abgerufen am 11.09.2023) die Anzahl an Berechnungen exponentiell auf Milliarden Rechenoperationen angestiegen ist. Die wachsende Bedeutung von KI-Anwendungen sowie deren zunehmende Komplexität bzw. immer höhere benötigte Rechenleistung führen zu einem erhöhten Stromverbrauch und infolgedessen zu einem Anstieg der CO_2-Emissionen. Dieser Verbrauch findet in erster Linie in Rechenzentren statt, in denen KI-Anwendungen in der Regel trainiert und ausgeführt werden. Vor dem Hintergrund der Zielsetzung der Bundesrepublik Deutschland, die CO_2-Emissionen bis 2030 um 60 Prozent gegenüber 1990 zu senken und bis 2045 komplett treibhausgasneutral zu werden, stellt die energieeffiziente Ausgestaltung digitaler Technologien eine Grundvoraussetzung für das Erreichen dieser Zielmarken dar. In anderen Staaten und Regionen kann die CO_2-Intensität des Stromverbrauchs von Rechenzentren außerdem noch höher liegen.

Komplexität des Use Case und des KI-Modells

Der Trend geht zu immer komplexer werdenden KI-Ansätzen und einem enormen Anstieg an zu verarbeitenden Daten. Dies führt zu einem zusätzlichen Rechenleistungs- und damit einem erhöhten Energiebedarf. Wenngleich KI ein großes Potenzial zur Energieeffizienzsteigerung bietet, muss daher verhindert werden, dass es bei ihrer Anwendung zu Rebound-Effekten kommt, also die Nutzeneffekte durch den erhöhten Energieverbrauch wieder aufgehoben werden. Insofern ist eine wirksame Lösungsstrategie zur energieeffizienten Ausgestaltung der Technologie zwingend

erforderlich. Dabei muss die Effizienz entlang der gesamten Wertschöpfungskette sowohl hardware- als auch softwareseitig betrachtet werden. Dies wurde im Projekt EEKI beispielhaft für die Hardware-Optimierung von KI-Modellen bei der Anwendung sowie für die Software-Lösung bei der energieeffizienten Übertragung von KI-Modellen durchgeführt.

1.2 Einführung in künstliche neuronale Netze

Die Entwicklung künstlicher Intelligenz verfolgt das Ziel, Maschinen in die Lage zu versetzen, Probleme und Aufgaben selbstständig zu lösen. Das KI-Teilgebiet des maschinellen Lernens befasst sich mit dem Trainieren von Modellen für das Erkennen und Vorhersagen statistischer Muster und Eigenschaften in Daten. Angelehnt an das biologische Vorbild eines natürlichen neuronalen Netzes zur Informationsverarbeitung werden beim Deep Learning, einem Teilgebiet des maschinellen Lernens, künstliche neuronale Netze (KNN) als Modelle verwendet. Die KI-Anwendungen, deren Energieeffizienz im Projekt untersucht wurde, basieren auf künstlichen neuronalen Netzen. Deren Aufbau und Funktion werden im Folgenden erläutert.

KNN sind üblicherweise in sequenziell angeordneten Schichten aufgebaut, wobei jede Schicht den Output der vorherigen Schicht transformiert bzw. filtert und wiederum als Input an die nächste Schicht weiterreicht (siehe Abbildung 1). Jede Schicht besteht aus einer Reihe von Knotenpunkten, an denen Informationssignale zusammenlaufen und für die nächste Schicht weiterverarbeitet werden. Ein Knotenpunkt wird in Anlehnung an das menschliche Gehirn auch als Neuron bezeichnet.

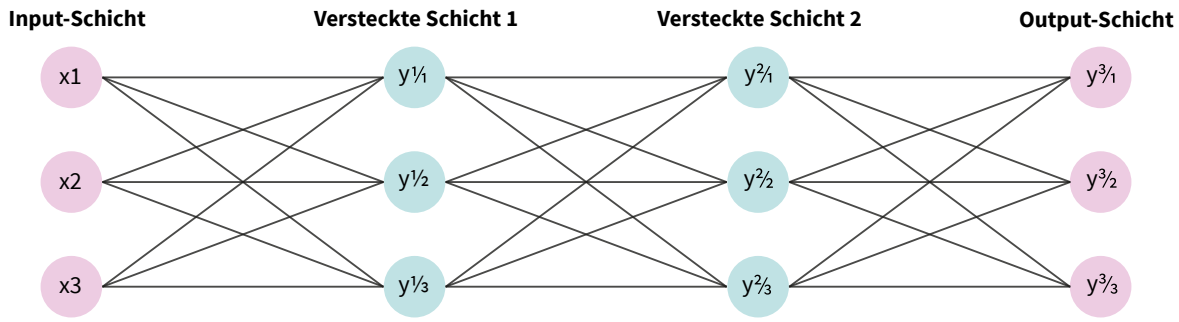


Abbildung 1: Schematische Darstellung eines KNN mit vier Schichten

Durch diese Eigenschaften sind KNN in der Lage, natürliche neuronale Systeme nachzubilden. Sie werden beispielsweise in der Bildklassifikation eingesetzt. Hier sorgen die verschiedenen Schichten zumeist für die Detektion geometrischer Objekte in unterschiedlichen Abstraktionslevels: Während der Input-Schicht nachgelagerte Schichten versuchen, einfache Strukturen wie etwa Kanten im Bild zu erkennen, setzen spätere Schichten diese einfachen Strukturen zu immer komplexeren Gefügen zusammen, wodurch schließlich ganze Szenen im Bild klassifiziert werden können.

Realisiert wird eine Schicht eines KNN durch eine Umwandlung (üblicherweise in Form einer linearen Transformation) der Aktivierungen der Input-Neuronen (Input-Schicht in Abbildung 1) mit anschließender nicht linearer Aktivierung der Output-Neuronen. Die Umwandlung der Aktivierung der Input-Neuronen kann als Matrix-Vektor-Multiplikation implementiert werden und stellt analog zur Synapse als biologische Verknüpfung zweier Nervenzellen die Verbindung zwischen den Input- und Output-Neuronen einer Schicht her.

Die Aktivierungsfunktion, das bedeutet die Parameter, die dazu führen, dass ein Neuron aktiviert wird, kann beispielsweise dafür sorgen, dass Neuronen, deren Aktivierung einen gewissen Schwellenwert nicht überschreitet, gar nicht erst aktiviert werden und sie somit keinen Einfluss auf nachfolgende Schichten haben. Gerade nicht lineare Aktivierungsfunktionen, wie etwa die häufig verwendeten Funktionen ReLU (Rectified Linear Unit) und GELU (Gaußian Error Linear Unit), ermöglichen die Verwendung beliebig komplexer Transformationsschichten, was wiederum KNN ihre vielseitige Einsetzbarkeit verleiht. Wichtig ist auch, dass es keine allgemeingültige Konvention gibt, ab welcher Anzahl von Schichten ein KNN als tiefes neuronales Netz bezeichnet wird.

Die Verbindungen zwischen den Input- und den Output-Neuronen einer KNN-Schicht stellen also Gewichte oder Gewichtungsfaktoren in einer Transformationsmatrix dar. Diese Gewichte werden innerhalb eines Trainings beispielsweise auf einem vorklassifizierten Datensatz iterativ mittels Optimierungsverfahren wie dem stochastischen Gradientenabstiegsverfahren so lange angepasst, bis sich eine gewünschte Klassifikationsgüte einstellt. Vereinfacht gesagt, gibt ein Gewicht den Grad an,

wie stark ein Input-Neuron pro Schicht verändert wird. Alle Gewichte bzw. Parameter eines KNN zusammen bilden ein (KI-) Modell. Auch der Trainingsprozess selbst bildet ein biologisches neuronales Netz nach, indem sich mit jedem Trainingsschritt die Stärke der Verbindungen verändert, also die Gewichtswerte für das KNN.

Die Trainingsphase zur Einstellung der Gewichte ist sehr rechenintensiv. Daher konnten erst in den letzten Jahrzehnten über die stetig steigende Verfügbarkeit von Rechenleistung und Verarbeitbarkeit großer Datenmengen infolge der Digitalisierung immer komplexere KNN konzipiert werden, die die bis dahin verwendeten Technologien bei Weitem übertrafen. Gerade die Verwendung tiefer neuronaler Netze kann Lösungen für komplexe Probleme in Anwendungsbereichen wie Bildanalyse, Sprachverarbeitung, Autonomes Fahren oder Medikamentenforschung und in vielen weiteren liefern. Durch diese Erfolge sind neuronale Netze inzwischen ein essenzieller Bestandteil vieler Anwendungen und aktuell geht die Entwicklung hin zu immer komplexeren und zuverlässigeren neuronalen Netzen. Entsprechend wird auch deren Topologie immer komplexer, also die Anzahl ihrer Schichten, Verknüpfungen und Parameter. Dies hat sich in den letzten Jahren noch einmal durch die generativen prädiktiven Transformer-Modelle (GPT) (OpenAI – Models, abgerufen am 11.09.2023) beschleunigt, die zum Beispiel in ChatGPT eingesetzt werden.

Ein typischer Vertreter eines solchen KI-Modells für Bilderkennung ist VGG16 (Visual Graph Group mit 16 Parameter-Schichten), ein neuronales Netz mit 21 Schichten (davon 16 Parameter-Schichten und fünf spezielle „MaxPool"-Schichten ohne Parameter) und über 138 Millionen trainierbaren Parametern. Im Vergleich dazu hat ein einfaches lineares Modell, das den Input lediglich linear transformiert, lediglich rund 50.000 trainierbare Parameter. Durch die generativen prädiktiven Transformer-Modelle, die in den letzten Jahren entwickelt wurden, hat sich die Anzahl an Parametern noch einmal exponentiell erhöht. So hat beispielsweise das GPT-3-Modell ca. 175 Milliarden Parameter (Brown, 2020).

Die zunehmende Komplexität von KI-Modellen scheint einerseits eine zwingende Voraussetzung für die Schlüsselstellung zu sein, die KI mittlerweile einnimmt, führt aber andererseits auch zu einem immer höheren Stromverbrauch für KI-Anwendungen. Neben der bereits angesprochenen rechenintensiven Trainingsphase betrifft die Problematik des höheren Energieverbrauchs insbesondere auch (1) die **Ausführung** (sogenannte Inferenz) dieser komplexen Modelle in KI-Anwendungen und aufgrund der zunehmenden Popularität dezentraler KI-Anwendungen, wie des Internet of Things (IoT) oder des Federated Learning (Föderiertes Lernen), auch (2) die **Übertragung** von KI-Modellen von einem Sender zu einem Empfänger. Diese Phasen waren Gegenstand des EEKI-Projekts und sind in diesem Bericht erläutert.

2. Energieeffiziente Ausführung von KI-Modellen

Wie bereits in der Einleitung dargestellt, hat sich der Stromverbrauch in Rechenzentren weiter erhöht, stärker sogar, als noch vor einigen Jahren prognostiziert. Ein wesentlicher Grund dafür besteht in Anwendungen des maschinellen Lernens (ML), die in den letzten Jahren drastisch zugenommen haben. Diese Anwendungen werden nun neben den klassischen Verfahren zur Signalverarbeitung und Computergrafik auf den vorhandenen prozessorbasierten Rechnersystemen und Servern, aber auch auf den klassischen Arbeitsplatzrechnern ausgeführt. Da diese Hardware ursprünglich nicht für KI-Prozesse konzipiert wurde, werden diese Anwendungen ineffizient ausgeführt und benötigen sehr viel Energie. Insbesondere haben prozessorbasierte Rechnersysteme eine geringe spezifische Rechenleistung pro Rechenknoten mit weniger Floating Point Operations (FLOPs) pro Sekunde. (Die Einheit wird als Vergleichsmaßstab für Rechenoperationen verwendet.)

Um hier für eine höhere spezifische Rechenleistung zu sorgen, wurden spezielle Rechenbeschleuniger entwickelt. So haben sich im Bereich der Grafikverarbeitung eigens darauf ausgerichtete Grafikkarten (GPU – Graphics Processing Unit) etabliert, während für allgemeinere Anwendungen der Signalverarbeitung überwiegend FPGA-basierte Systeme (FPGA – Field Programmable Gate Array) entwickelt und eingesetzt wurden. Auch in neueren Anwendungen, wie dem Edge Computing in Mobilfunkanwendungen, kommen FPGAs zum Einsatz. Bezüglich der neueren Anwendungen zum maschinellen Lernen werden sowohl GPUs als auch FPGAs eingesetzt (Falsafi et al., 2017). Beide Systeme werden etwa von kommerziellen Anbietern von Cloud-Computing-Ressourcen angeboten. Als Beispiel sei hier Amazon Cloud AWS genannt, das Nutzern sowohl FPGA-Systeme (Amazon-a, abgerufen am 16.09.2023) als auch GPU-basierte Rechnerinstanzen (Amazon-b, abgerufen am 16.09.2023) anbietet. Auch der Cloud-Anbieter Microsoft Azure setzt teilweise FPGA-basierte Systeme zur Beschleunigung seiner KI-basierten Dienste ein, wie zum Beispiel für die Suchmaschine Microsoft Bing (Putnam, Caulfield, Chung & Burger, 2015).

Neben diesen großen Anbietern von Public-Cloud-Systemen setzen zunehmend auch nationale Anbieter in großem Maße Rechenbeschleuniger ein, um den drastisch steigenden Bedarf an Rechenleistung für KI-Anwendungen decken zu können. Dies gilt in besonderem Maße auch für Deutschland, da Datenschutzbehörden aus rechtlichen und sicherheitsrelevanten Erwägungen von der Nutzung globaler Anbieter wie Amazon, Google und Microsoft teilweise abraten.

Abbildung 2 stellt eine typische Konfiguration einer Private-Cloud-Installation dar, wie sie bei Bedarf von entsprechenden Dienstleistern angemietet werden kann. Private-Cloud-Dienste sind dadurch gekennzeichnet, dass alle Hard- und Software-Ressourcen einem einzigen Kunden zugeordnet sind. Große Unternehmen, aber auch staatliche Institutionen nutzen Private-Cloud-Dienste aus Effizienz- und Sicherheitsgründen. Die in der Abbildung dargestellte Konfiguration beinhaltet spezielle Rechenbeschleuniger (Compute Nodes), wie sie für KI-Anwendungen eingesetzt werden. Neben einer Hochgeschwindigkeits-Verbindungsstruktur besitzt sie ein Speicher- und ein Managementsystem, über das die Lastverteilung der Rechner vorgenommen werden kann.

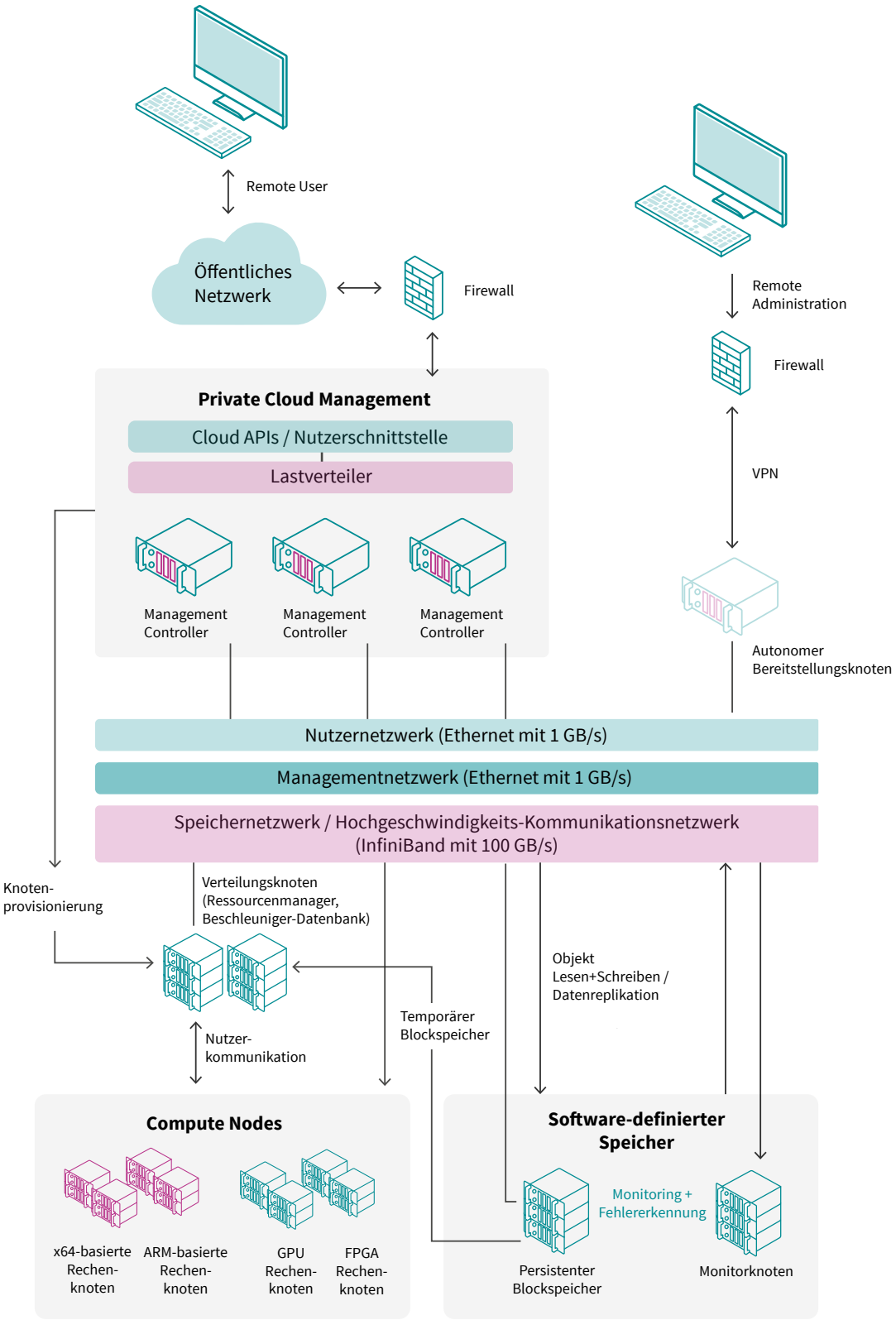


Abbildung 2: Typische Private-Cloud-Architektur mit GPU- und FPGA-Compute Nodes für KI-Anwendungen

Die im Projekt entwickelte Methode zur energieeffizienten Anwendung von KI-Modellen besteht in einer neuartigen Art und Weise, wie KI-spezifische Rechenbeschleuniger in Rechnersysteme integriert werden können (siehe auch Abbildung 2). Sie zeichnet sich dadurch aus, dass auf einen großen Teil der normalerweise benötigten Serverrechner verzichtet werden kann. Dies wird durch eine starke Reduzierung der Verlustleistung (Differenz zwischen benötigtem Strom für Rechenleistung und tatsächlich verbrauchtem Strom) erreicht, die in herkömmlichen Systemen anfällt. Entsprechend kann die Energieeffizienz erheblich gesteigert und damit der KI-spezifische CO_2-Ausstoß wesentlich verringert werden.

Die Verarbeitungsarchitekturen für KI sind auf unterschiedliche Weise implementiert: für GPU- oder prozessorbasierte Systeme meist als Software und für FPGA-basierte Systeme als Hardware-Komponenten. Die Hauptrechenlast und damit der größte Anteil des Energiebedarfs werden dabei durch die Kernalgorithmen der KI-Anwendungen (wie Matrix- und Tensor-Multiplikationen) verursacht. Zur Reduktion dieses Energiebedarfs gibt es unterschiedliche Ansätze. So wird im Wesentlichen an Lösungen gearbeitet, die FPGAs für die Beschleunigung von KI-Algorithmen verwenden oder eigene Chips (ASIC – Application-Specific Integrated Circuit) entwickeln. ASICs wiederum stellen die optimale Lösung für eine spezifische Anwendung dar, insbesondere bezüglich des Energieverbrauchs pro Rechenoperation, sind jedoch durch ihre spezielle Entwicklung sehr teuer und können nicht für andere Anwendungen verwendet werden (Sze, Chen, Yang & Emer, 2017). Bei aktuellen Halbleiterprozessen belaufen sich allein die reinen Einrichtungskosten für die Fertigung auf mehrere Millionen Euro.

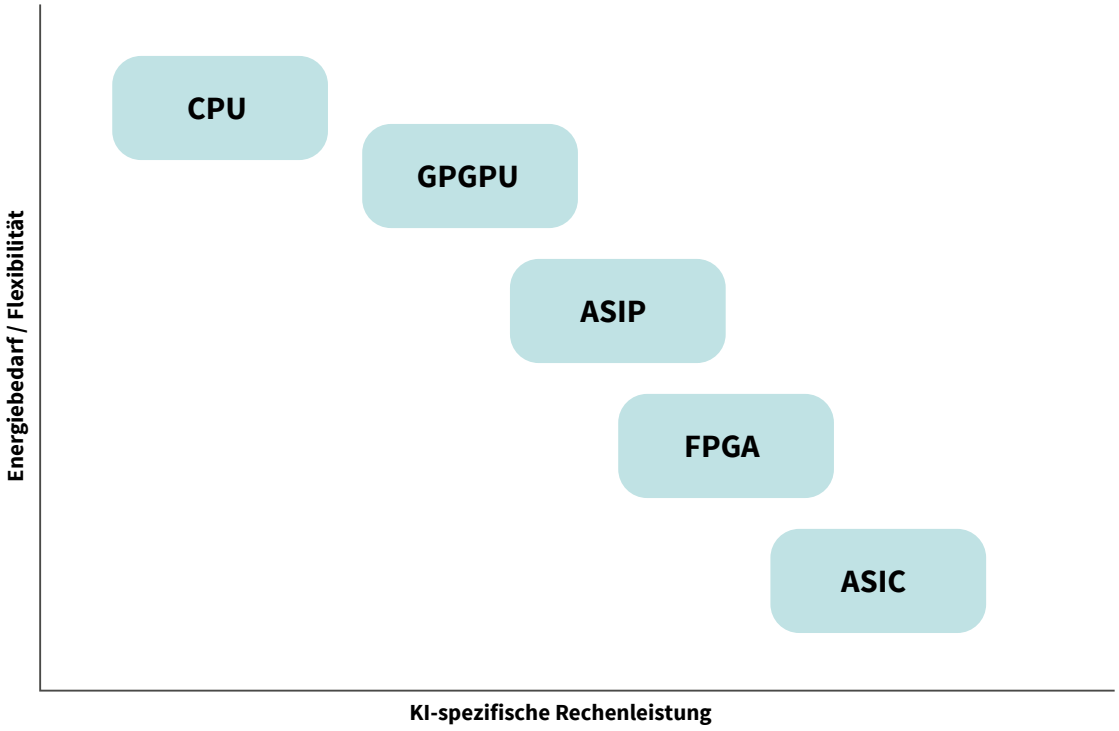


Abbildung 3: Rechnersysteme und ihre KI-spezifische Leistungsfähigkeit sowie Energiebedarf/Flexibilität

In Abbildung 3 sind unterschiedliche Rechnersysteme bezüglich ihrer Rechenleistung für KI-Anwendungen und ihres Energiebedarfs dargestellt. Eine Steigerung der Energieeffizienz wird in der Regel durch einen höheren Spezialisierungsgrad der jeweiligen Ausführungsplattform erzielt. Verfügt ein Prozessor zum Beispiel über spezielle Instruktionssatzerweiterungen[1], um Matrixmultiplikationen bearbeiten zu können, ist er entsprechend besser für Anwendungen des maschinellen Lernens geeignet, da er die Kernoperationen optimaler, das heißt schneller in Bezug auf die Anzahl benötigter Takte einer gegebenen Taktfrequenz, ausführen kann (Nurvitadhi et al., 2016).

Als Erweiterung der GPUs wurden sogenannte General Purpose GPUs (GPGPU) entwickelt, die über sehr viele kleine, spezielle Verarbeitungseinheiten verfügen und besonders effizient bei Problemstellungen mit hoher Datenparallelität sind (Datenparallelität bedeutet, dass Daten unabhängig voneinander und damit gleichzeitig bearbeitet werden können), da sie nach dem SIMD[2]-Prinzip arbeiten. Zusätzlich profitieren GPGPUs vom lokalen Speicher der Grafikkarte und den damit verbundenen kürzeren Zugriffszeiten gegenüber weiter entfernten Speichern wie dem Hauptspeicher eines Computers sowie von einem breitbandigen Speicher-Interface mit hoher Datenrate zum externen Speicher.

Eine weitere Effizienzsteigerung stellen applikationsspezifische Prozessoren, sogenannte ASIPs (Application-Specific Instruction-Set Processors) dar. Bei ihnen wird die grundsätzliche Programmierbarkeit eines speziellen Prozessors zugunsten der effektiveren Abarbeitung von KI-Algorithmen eingeschränkt. Ein typischer ASIP-Vertreter ist die Tensor Processing Unit (TPU) von Google (Google, abgerufen am 16.09.2023), die speziell auf Anwendungen des maschinellen Lernens optimiert wurde und nur für ausgewählte ML-Verfahren programmierbar ist. Auf den bisher beschriebenen Plattformen werden in der Regel Algorithmen unter Einsatz einer spezialisierten Programmiersprache implementiert, die dann unter Anwendung der entsprechenden Entwicklungssoftware in einen auf dem jeweiligen Prozessor ausführbaren Code überführt werden.

Eine weitere Steigerung der Energieeffizienz lässt sich über die Entwicklung spezieller Hardware-Beschleuniger für bestimmte Halbleitertechnologien erreichen. Hier nehmen FPGAs eine Sonderrolle ein, da sie zum einen den Vorteil besitzen, spezielle Hardware-Implementierungen abzubilden, und zum anderen ähnlich wie Central Processing Units (CPUs) oder GPUs während des Betriebs rekonfigurierbar sind, womit eine eigene Art der Programmierbarkeit gegeben ist. Durch ihre spezielle Hardware-Implementierung können FPGAs auch irreguläre Verarbeitungsstrukturen gut abbilden. Gleichzeitig enthalten sie Tausende Hardware-Multiplizierer, die für neuronale Netze eine große Bedeutung haben. In Xia et al. (2019) wird ein FPGA-basiertes System beschrieben, das der umfangreichen parallelen Verarbeitung von tiefen neuronalen Netzen dienen soll.

Damit stellen FPGAs auch in Abbildung 3 ein Optimum zwischen KI-spezifischer Rechenleistung, Flexibilität und Energiebedarf dar und wurden auch als Basis für die Rechnerarchitektur zur energieeffizienten KI-Modellanwendung im Rahmen des Projekts verwendet. Dabei lag der Fokus auf der Flexibilität im Hinblick auf unterschiedliche Netzarchitekturen, wobei eine Kopplung der FPGAs über ein Peripheral Component Interconnect Express (PCIe) Interface[3] vorgenommen wird. Bei den untersuchten Netzen zeigt sich eine leicht schnellere Verarbeitung der Daten bei gesteigerter Energieeffizienz. In Wang et al. (2020) wurde die massive Parallelisierung des Trainings von Deep Convolutional Neural Networks (DCNN) mithilfe von FPGAs gezeigt und eine Energieeinsparung um den Faktor vier beschrieben. Zudem verdeutlicht das dargestellte Verfahren zwar die Bedeutung der schnellen Netzwerkkommunikation, bietet aber keine Integration in Rechenzentren. Im Projekt Brainwave (Microsoft, abgerufen am 16.09.2023) verwendet Microsoft einen mittels PCIe eng an der Server gekoppelten FPGA für maschinelles Lernen der Suchmaschine Bing. In dem Bericht dazu werden insbesondere die geringe Latenz bei zugleich hohem Durchsatz sowie die geringen Kosten von 21 US-Cent pro 1 Million Bildern hervorgehoben. Intel (Nurvitadhi et al., 2019) zeigt die Bedeutung von eng an Server gekoppelten FPGA-Beschleunigern für komplexe neuronale Netze.

Die höchste Energieeffizienz (siehe Abbildung 3) ergibt sich durch ASICs, die jedoch nur für eine spezielle Aufgabe entwickelt wurden und in ihrer Fertigung, wie bereits beschrieben, sehr teuer sind.

Aus diesen Betrachtungen ergab sich die Vorgabe für FPGA-basierte Rechenbeschleuniger. Die Ergebnisse dieser Entwicklung sind in den folgenden Abschnitten dargestellt, insbesondere zur Integration des Rechenbeschleunigers in aktuelle Rechenzentren, zu den notwendigen Schnittstellen und zur erforderlichen Datenübertragung. Des Weiteren wird die exemplarische Verlustleistungsbetrachtung als Grundlage für die Beurteilung unterschiedlicher Verarbeitungsarchitekturen beschrieben.

1 Erweiterung des Befehlssatzes einer CPU für spezielle Aufgaben, beispielsweise für Matrixmultiplikationen.
2 Single Instruction Multiple Data (SIMD) beschreibt die parallele Verarbeitung von Datenwörtern, die sich in einem Prozessorregister befinden, jedoch eine getrennte arithmetische Verarbeitungseinheit verwenden.
3 Standard für eine Erweiterungsschnittstelle eines PC, über die man zum Beispiel Grafikkarten anbinden kann.

2.1 Alternative Methoden der Integration von Rechenbeschleunigern

Neben den oben erwähnten PCIe-Schnittstellen bieten herkömmliche Rechnersysteme eine Vielzahl weiterer Schnittstellen, deren wichtigster Vertreter die standardisierte Ethernet-Schnittstelle ist, eine Hochgeschwindigkeits-Verbindungsstruktur (> 10 GBit/s). Ethernet-Schnittstellen werden in Rechenzentren neben Infiniband[4] für die Vernetzung von Rechnersystemen verwendet. Insofern bietet sich Ethernet als Schnittstelle für die Kommunikation mit Hardware-Beschleunigern an, da die Schnittstellen weit verbreitet sind.

Die ideale Lösung wäre eine Konfiguration, in der die Schnittstelle direkt auf dem Beschleuniger untergebracht ist und kein zusätzlicher Hostrechner für den Datentransfer benötigt wird. Dafür ist der Einsatz eines komplett hardwarebasierten Netzwerkprotokollstapels erforderlich, der es erlaubt, beliebige Rechenbeschleuniger über standardisierte Netzwerkschnittstellen anzusprechen und zu betreiben. Die Schnittstelle ist dann, je nach verwendeter Technologie, mit bis zu 100 Gbit/s betreibbar und liefert damit ähnlich hohe Datenraten wie herkömmliche Bussysteme.[5]

Diese ideale Lösung als reines Hardware-System hat weitere Vorteile, wie zum Beispiel niedrige Latenz, kurze Reaktionszeiten und eine Optimierung der Ausfallsicherheit, da auf jegliche Art von Software verzichtet werden kann.

Zusammenfassend lässt sich hier feststellen: Die Umsetzung hardwarebasierter Netzwerkprotokolle erlaubt es, Hardware-Beschleuniger direkt mit einer Hochgeschwindigkeits-Netzwerkschnittstelle auszustatten (beispielsweise Ethernet), wie sie für die Integration in Rechenzentren benötigt wird, ohne dabei ein Rechnersystem einsetzen zu müssen.

2.2 Konzeption einer Rechnerarchitektur für energieeffiziente Inferenz

Projektziel war, die Energieeffizienz während der Ausführung (Inferenz) über eine neuartige FPGA-basierte Rechnerarchitektur zu steigern. FPGA-Beschleuniger stellen, wie weiter oben dargelegt, eine ideale Mischung aus Flexibilität, Performance und Energieeffizienz dar und bieten sich deshalb als Basis für eine energieeffiziente Rechnerarchitektur an. In bisherigen Systemen sind die FPGA-Beschleuniger meist eng über das interne Bussystem PCIe mit einem Trägerrechner gekoppelt, wie in Abbildung 4 deutlich wird.

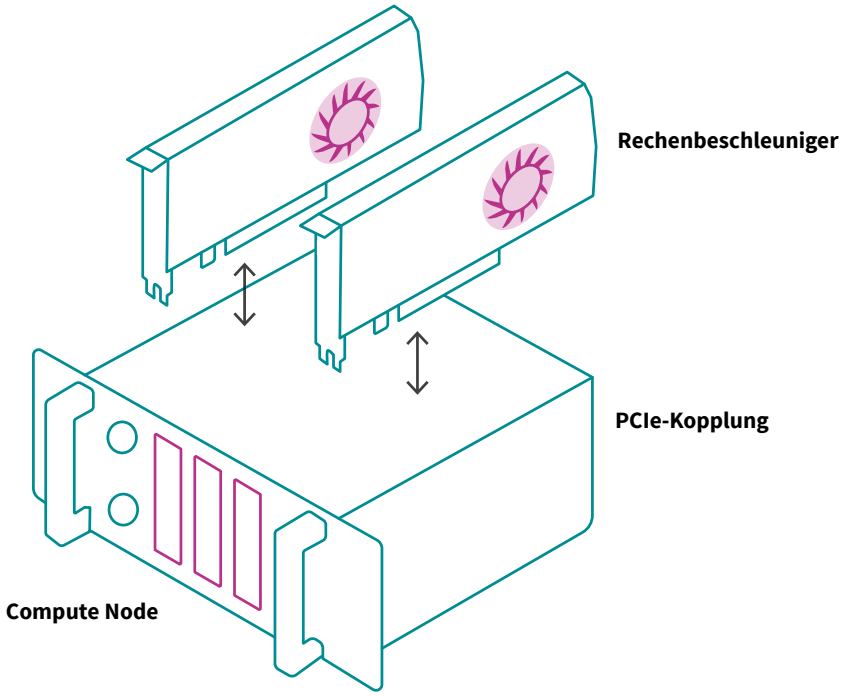


Abbildung 4: PCIe-basierte Anbindung an FPGA- bzw. GPGPU-basierte Rechenbeschleuniger

4 Hochgeschwindigkeitsnetzwerk zur Nutzung in Rechenzentren mit geringer Latenz.
5 Bussysteme dienen als Kommunikationsstruktur in Computern. Damit wird beispielsweise die Speicherhardware an den Prozessor angebunden.

Diese enge Bindung limitiert die Anzahl der Systeme pro Trägerplattform auf oftmals zwei Beschleuniger pro Rechenknoten.

Zur Umgehung dieser Limitierung werden die FPGA-Beschleuniger mittels Netzwerkschnittstellen an das Hochgeschwindigkeits-Kommunikationsnetzwerk des Rechenzentrums angeschlossen. Sämtliche Kommunikation mit dem Beschleuniger erfolgt über das Netzwerk. Auf Basis dieser Kerntechnologie kann zudem auf einen großen Teil der normalerweise benötigten Serverrechner durch die Netzwerkkopplung verzichtet werden. Durch die so zu erreichende drastische Reduzierung der Gesamtverlustleistung kann die Energieeffizienz erheblich gesteigert und damit der KI-spezifische CO_2-Ausstoß wesentlich verringert werden. Das Konzept der netzwerkgekoppelten FPGA-Beschleuniger wird auch NAA (Network-Attached Accelerator) genannt.

Die Anbindung der FPGA-Beschleuniger an einen Steuerrechner über das Netzwerk ist in Abbildung 5 schematisch dargestellt.

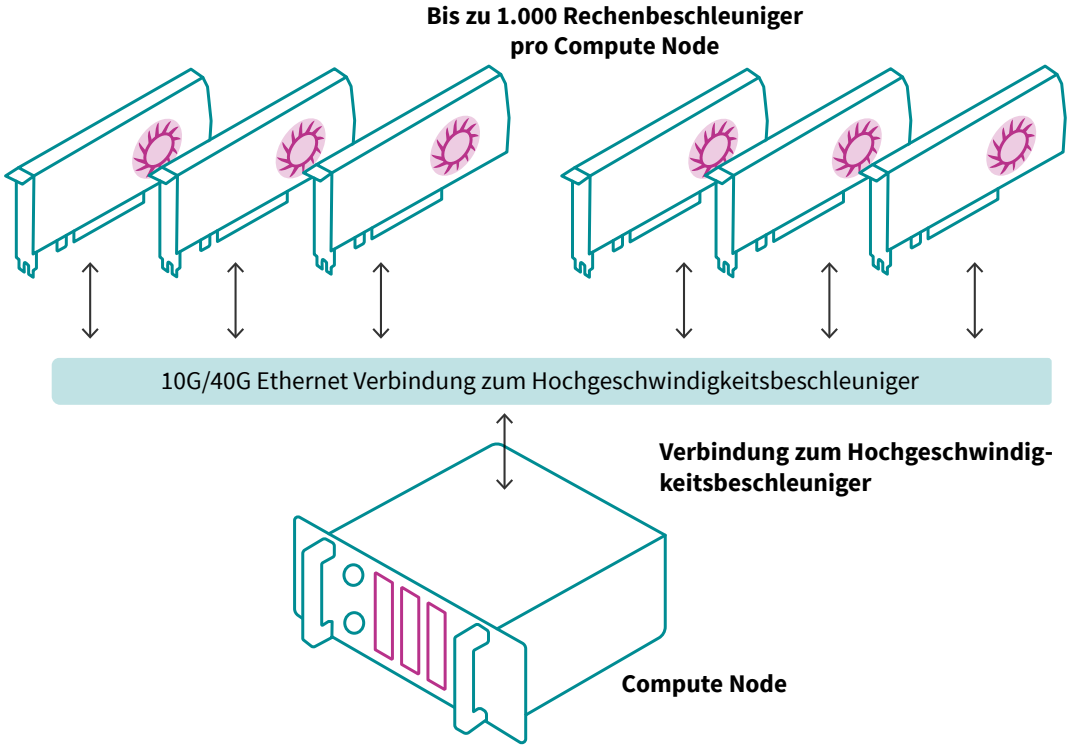


Abbildung 5: Anbindung der netzwerkbasierten FPGA-Beschleuniger an einen Steuerserver (Compute Node)

2.3 Implementierung der NAA-Rechnerarchitektur für energieeffiziente Inferenz

Da die energieeffiziente Ausführung (Inferenz) eines neuronalen Netzes (DCNN) auf Basis einer neuen Rechnerarchitektur mit allein stehenden, netzwerkgekoppelten Rechenbeschleunigern (hier FPGA) unter Verzicht auf einen eng gekoppelten Trägerserver erfolgen sollte, stellt die Netzwerkkommunikation einen wesentlicher Baustein dar.

Die Kommunikation zu und zwischen den NAAs erfolgt über das Ethernet-basierte Hochgeschwindigkeitsnetzwerk mit der Bezeichnung RoCEv2 (RDMA over Converged Ethernet Version 2). Dies bietet eine sehr gute Interoperabilität und eine zuverlässige, verbindungsorientierte Kommunikation, geringe Latenz und RDMA-Transfers (Remote Direct Memory Access), was die CPU-Last der Steuerrechner verringert.

Um die Schnittstellen eines FPGA nutzen zu können, wurde ein Hardware-Framework (auch Accelerator Framework), wie in Abbildung 6 dargestellt, konzeptioniert und implementiert. Das Accelerator Framework bietet Zugriff auf den externen DDR3-Speicher und dient der Kommunikation zwischen verschiedenen FPGA-internen Funktionsmodulen. Zudem sind im Framework die Netzwerkschnittstelle sowie das RDMA-Protokoll RoCEv2 für den externen Datenaustausch via Netzwerk implementiert.

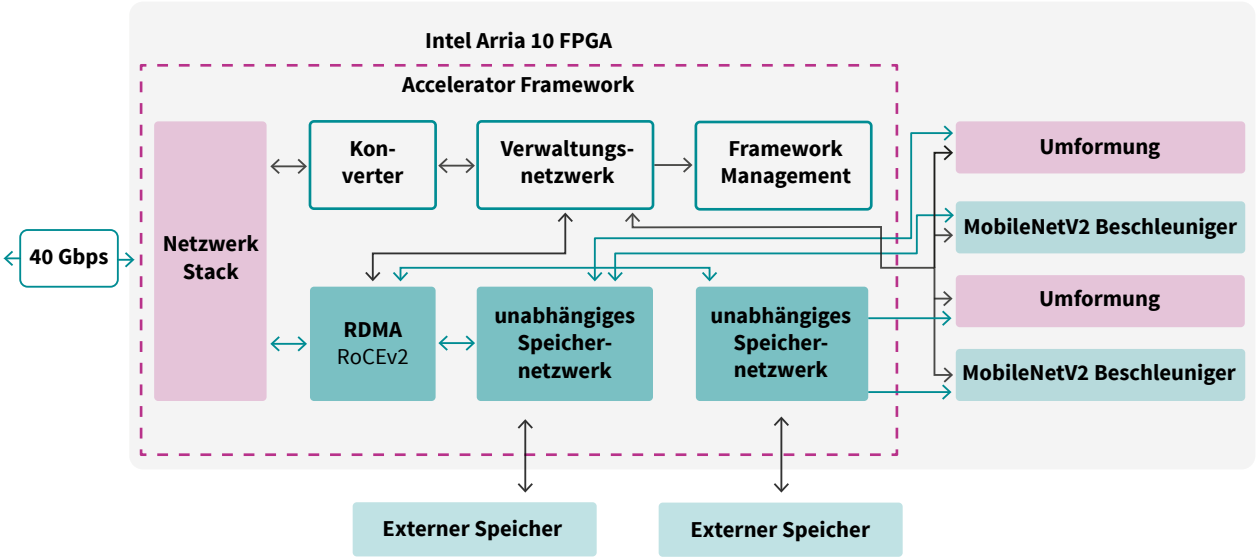


Abbildung 6: NAA-Hardware-Framework

Im Laufe des Projekts wurde der RoCEv2-Funktionsblock, das heißt die Einheit im FPGA, die die Funktion RoCEv2 ausführt, in das Framework integriert. Kommunikationstests mit Testdaten wurden erfolgreich durchgeführt, womit die Integration der ML-Beschleuniger, also die Versorgung der Beschleuniger mit neuen Eingangsdaten sowie das Auslesen der erkannten Objektklassen, sichergestellt werden konnte. Durch die zuverlässige, verbindungsorientierte Kommunikation von RoCEv2 ist ein reibungsloser Betrieb auch in störungsbehafteten Netzwerkumgebungen garantiert. Dies wird durch robuste Fehlererkennungsmechanismen und Korrekturverfahren sichergestellt.

2.4 Spezifikation der Key-Performance-Indikatoren

Um die Effizienzsteigerungen zu messen, wurden Key-Performance-Indikatoren für den Energieverbrauch definiert. Maßgeblich ist dabei der Energieverbrauch einer Klassifizierung pro Bild bei der Inferenz. Dieser Maßstab stellt die eingesetzte Energie in einem Verhältnis zum Durchsatz (also zur Rechenleistung) entsprechend der folgenden Formel dar:

$$EnergieproFrame = \frac{Gesamtenergie\ des\ Messzeitraums}{Anzahl\ der\ Bilder}$$

Bei der Ermittlung des Gesamtenergieverbrauchs müssen nicht nur die eigentlichen Rechenknoten berücksichtigt werden, sondern auch die weiteren Komponenten wie Kontrollrechner und Netzwerkkomponenten (z. B. Netzwerkverteiler). Dadurch wird der Einfluss der Systemarchitektur korrekt erfasst. Für eine detaillierte Erfassung des Energieverbrauchs der einzelnen netzwerkgekoppelten FPGA-Beschleuniger (NAAs) wird er jeweils separat ermittelt. Dafür wurde eine eigene Trägerplatine mit Messsensoren entwickelt. Durch die detaillierte Messung können Rückschlüsse auf Netzteilverluste der NAA-Stromversorgung sowie den Beschleunigerverbrauch in unterschiedlichen Betriebszuständen gezogen werden. Dieses Wissen kann zur weiteren Optimierung genutzt werden.

Ergänzend zum Energieverbrauch sollten exemplarisch die Energiekosten (für die Mittelspannungsebene, da Rechenzentren als Großverbraucher üblicherweise über diese versorgt werden) sowie die CO_2-Belastung des Energieverbrauchs mit deutschen Strommix (Umweltbundesamt, 2023) dargestellt werden. Die Anschaffungskosten der verwendeten Rechenbeschleuniger sowie der weiteren Hauptkomponenten als Kostenmetrik sollen als weiterer Vergleichsindikator dienen.

Die ermittelten Zahlen werden in Relation zu einer nicht beschleunigten Basisvariante bestehend aus CPU-basierten Rechenknoten gestellt.

2.5 FPGA-NAA-Testaufbau

Im Folgenden ist die im Laufe des Projekts implementierte NAA-Architektur dargestellt. Für den Aufbau des Systems wurde eigens ein entsprechendes NAA-Gehäuse angeschafft, in das speziell hierfür erstellte Trägerplatinen untergebracht wurden. Eine einzelne Trägerplatine bietet dabei mechanische Halterung, Stromversorgung sowie Messsensoren für zwei FPGA-basierte NAAs. Für weitere NAAs können modular zusätzliche Platinen verwendet werden, wie in Abbildung 7 mit vier Platinen dargestellt. Dies ermöglicht eine flexible Weiternutzung für andere Gehäuseformen mit weniger oder mehr Trägerplatinen. In der Bildmitte ist das Leistungsverteilungsboard zu sehen, das den Stromfluss des Servernetzteils verlustarm auf bis zu fünf Trägerplatinen aufteilt.

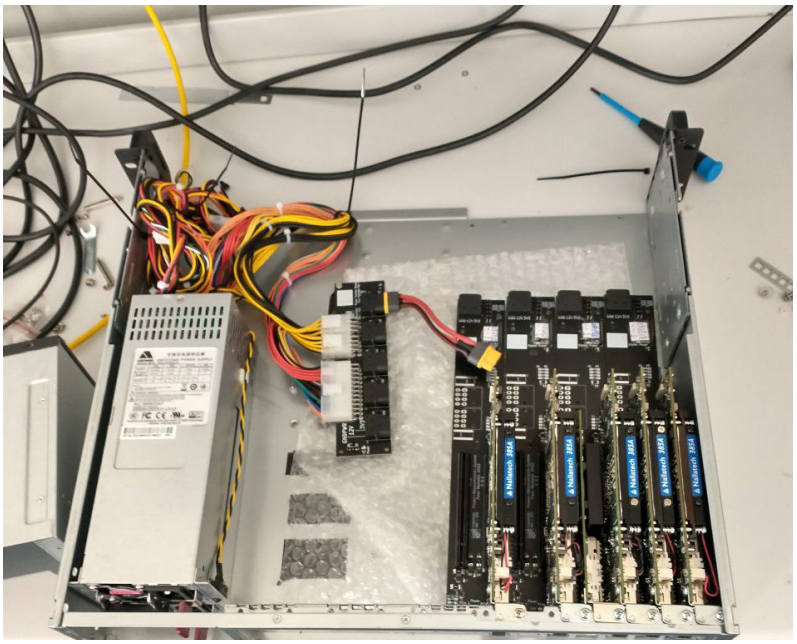


Abbildung 7: NAA-Gehäuse

Die Integration zweier Beschleuniger für spezielle neuronale Netze (im vorliegenden Fall wurde das neuronale Netz MobileNetV2 gewählt) in das FPGA-Framework mit Netzwerkkommunikation wurde dabei in Hardware-Tests anhand eines einzelnen NAA-FPGA erfolgreich verifiziert.

Beide MobileNetV2-Beschleuniger innerhalb eines NAA können gleichzeitig und parallel genutzt werden. Eine Testanwendung kommuniziert dabei mit den NAAs nach dem Kommunikationsmodell 1–n, das heißt, eine Anwendung nutzt eine oder mehrere NAA-Instanzen zur Beschleunigung der Klassifizierung. Die Klassifizierung von Bilddaten nach bestimmten Klassen ist in diesem Fall die Aufgabe, die mithilfe des neuronalen Netzes ausgeführt wird. In Abbildung 8 ist die Übertragung der Klassifizierungsergebnisse zweier NAAs mittels RDMA-Kommunikation dargestellt. Insgesamt können in diesem Setup vier NAAs (entsprechend acht MobileNetV2-Instanzen) parallel verwendet werden, die zusammen einen Gesamtdurchsatz von 5.162 Bildern pro Sekunde erreichen. Dabei beträgt die Klassifizierungsgenauigkeit 71,7 Prozent über das Testdatenset (genauer ILSVRC2012 Validierungsset mit 50.000 Bildern). Dies entspricht auch der erreichten Genauigkeit auf einer CPU, sodass die FPGA-Implementierung äquivalente Ergebnisse erzielt.

```
[RoCE] Connecting to '10.42.42.102'[RoCE] using '10.42.42.1'...
[RoCE] Connected to remote '10.42.42.102' using local '10.42.42.1' successfully.
[RoCE] Reading data with size 262144.
registering addr: 0xd33e90 with length: 262144
init read remote address: 4633133056
[RoCE] read data successfully, disconnecting...
[RoCE] Disconnected.
[RoCE] Freed up.
[RoCE] Connecting to '10.42.42.103'[RoCE] using '10.42.42.1'...
[RoCE] Connected to remote '10.42.42.103' using local '10.42.42.1' successfully.
[RoCE] Reading data with size 163840.
registering addr: 0xcffaa0 with length: 163840
init read remote address: 338165760
[RoCE] read data successfully, disconnecting...
[RoCE] Disconnected.
[RoCE] Freed up.
4.537
2.023
0.001
2.168
0.284
total: 9.014
total: 50000; hits: 35871 -> accuracy: 71.742
time taken: 9.69s -> throughput: 5162.03
```

Abbildung 8: Ausschnitt der Testanwendung zur MobileNetV2-Klassifizierung

Aufbauend auf der kompletten Testanwendung sowie der Integration der MobileNetV2-Beschleuniger in das FPGA-Framework wurden zunächst vier NAAs (mit acht MobileNetV2-Instanzen) und dann acht NAAs (mit 16 Instanzen) parallel betrieben. Die dabei auftretenden Herausforderungen mit der parallelen Netzwerkkommunikation konnten gelöst werden, sodass mit allen Instanzen eine zuverlässige, hochratige und latenzarme Kommunikation auch im Dauerbetrieb möglich ist. Dies ist eine wesentliche Grundlage für den Durchsatz der Inferenz und somit für die Energieeffizienz pro klassifiziertem Bild.

Das laufende Gesamtsystem ist in Abbildung 9 prototypisch dargestellt. Oben im Bild ist das NAA-Gehäuse mit hocheffizientem Servernetzteil und acht eingebauten NAAs zu sehen. Jeder NAA ist über ein passives 40-GBit/s-Kabel an den Switch in der Bildmitte angeschlossen. Der einzelne Kontrollserver im unteren Bildteil dient als Datenquelle bzw. -senke und Steuereinheit. Im NAA-Gehäuse sind zudem Mikrocontroller verbaut, die Leistungsmessdaten an eine Datenbank weiterleiten (siehe Kapitel 2.7 Energiemessung).

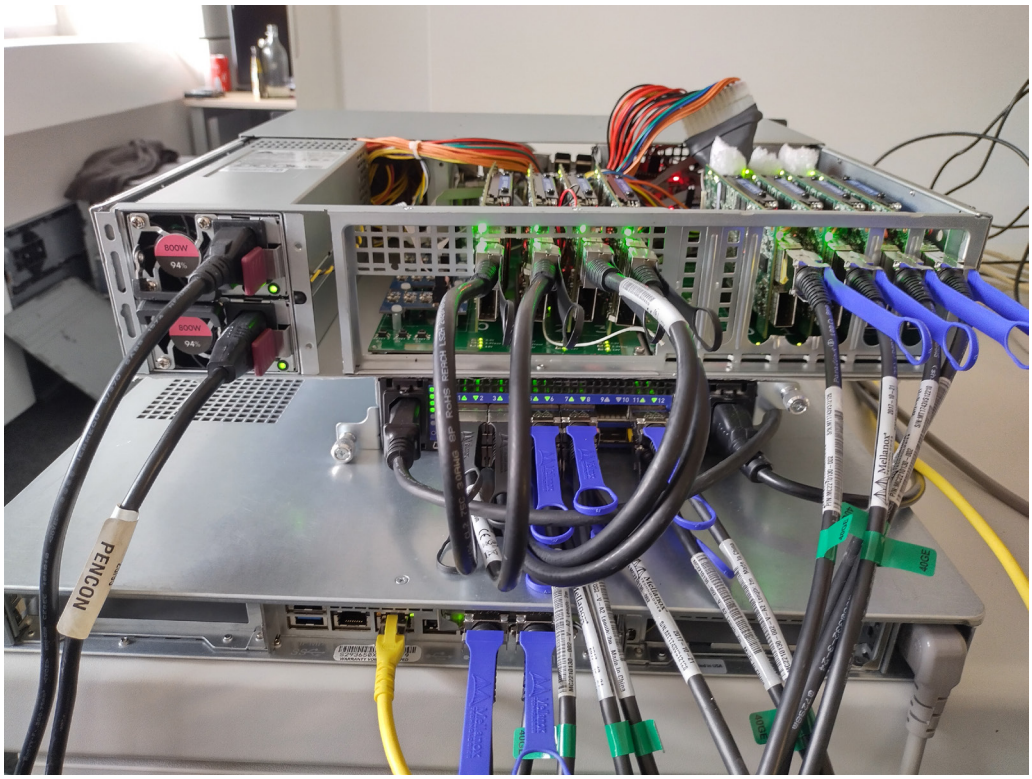


Abbildung 9: Prototypischer Gesamtsystemaufbau mit acht NAAs

2.6 Messkonzept für NAA-Beschleuniger

Das Messkonzept für PCIe-basierte NAA-Beschleuniger ist in Abbildung 10 dargestellt. Für den Betrieb der NAAs wurde eine Trägerplatine mit integrierten Leistungssensoren zur Strommessung entwickelt. Die Trägerplatine stellt über den PCIe-Slot die Stromversorgung sicher und bietet keinerlei Kommunikationsmöglichkeit mit den NAAs. Die Leistungssensoren messen sowohl die Versorgung über den PCIe-Slot als auch die zusätzlichen 12-V-Leitungen, sofern vorhanden. Die Messwerte werden durch einen Mikrocontroller aufbereitet und per Netzwerk an eine zentrale Datenbank gesendet. Zudem erfolgt eine Visualisierung mittels Dashboard.

Zusätzlich erfolgt eine Messung von Steuerservern, Switch und NAA-Gehäuse mittels handelsüblicher Leistungsmessgeräte.

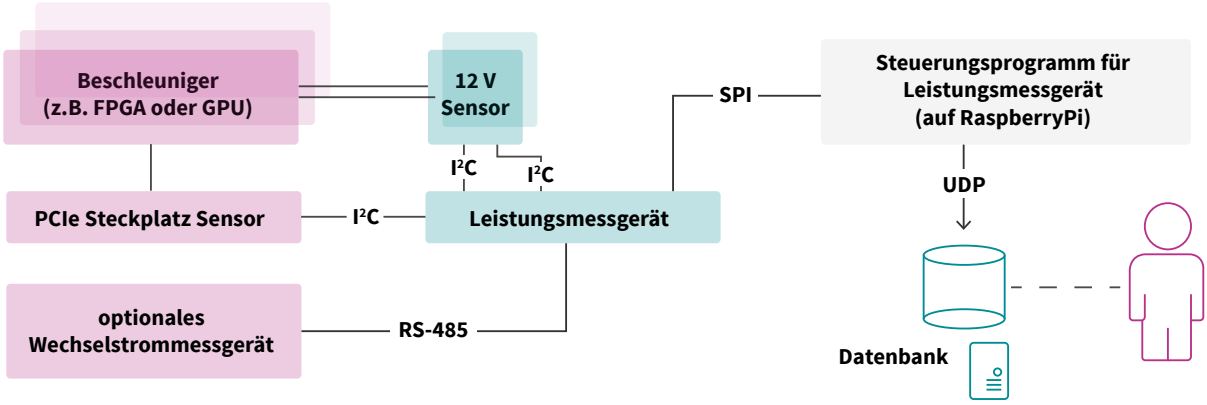


Abbildung 10: Messkonzept für NAA-Beschleunigerkarten

2.7 Energiemessung

Gemäß dem im letzten Abschnitt dargestellten Messkonzept wurde für jeden NAA die benötigte Leistung mittels Mikrocontroller gemessen. Diese Sensormessworte werden an eine zeitbasierte Datenbank (InfluxDB) übertragen und stehen dort für die weitere Auswertung auch nach Abschluss einer Berechnung zur Verfügung. Um die benötigte Rechenlast der Datenbank zu reduzieren, werden die Messwerte in einem Batch zusammengefasst und als Block gespeichert. Durch Zeitstempel bleibt eine eindeutige Zeitzuordnung erhalten.

Die Messwerte können live oder im Nachgang in einem Dashboard visualisiert werden, wie in Abbildung 11 dargestellt. Neben der Gesamtleistung kann auch die Leistung einzelner Messkanäle angezeigt werden.

Abbildung 11: Dashboard zur Energiemessung eines NAA

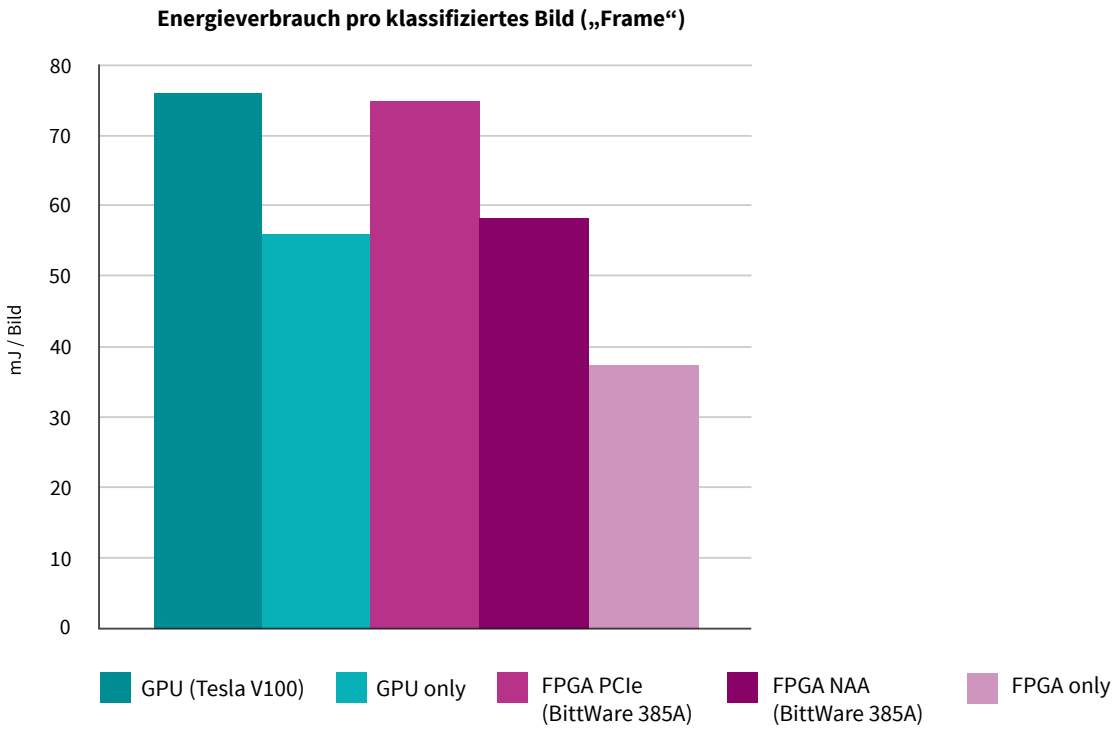


Abbildung 12: Referenzenergiemessung der MobileNetV2-Inferenz

Auf Grundlage des dargestellten Messkonzepts konnten entsprechende vergleichende Energiemessungen vorgenommen werden, die im Folgenden dargestellt sind. Hierfür wurden die Energiemessungen der MobileNetV2-Inferenz mit acht NAAs im vorgestellten NAA-Gehäuse durchgeführt. Das NAA-Gesamtsystem bestehend aus Kontrollserver, Switch und NAA-Gehäuse ist von allen untersuchten Ausführungsplattformen mit 58,2 mJ/Bild die energieeffizienteste Lösung (vgl. unten stehende Tabelle). Die FPGAs benötigen davon 37,5 mJ/Bild, der Server 13,7 mJ/Bild und der Switch 3,8 mJ/Bild.

Das NAA-System ist somit 1,29-mal effizienter als die Inferenz auf PCIe-gekoppelten FPGAs und 1,31-mal effizienter als auf GPUs.

Bei der Bewertung der Ergebnisse ist zu beachten, dass

1. die verwendeten Arria 10 GX FPGAs aus dem Jahr 2013 in einem 20-nm-Fertigungsprozess produziert wurden, was gegenüber den GPUs aus dem Jahr 2017, gefertigt im TSMC 12 nm FFN Prozess, einen Technologienachteil darstellt. Bei der Nutzung von Hardware mit gleichem Baujahr ist zu erwarten, dass die Energieeinsparung mit FPGA noch größer ausfällt.

2. beim NAA-Gesamtsystem Kontrollserver und Switch mit betrachtet werden. In einem Rechenzentrum werden aber auch für eine Inferenz auf einem GPU-Server ein Ressourcenmanagement- und Steuerserver sowie ein Switch für die Netzwerkkommunikation benötigt. Somit fällt der Vergleich zu Ungunsten der NAA-Architektur aus. Unter Berücksichtigung des Kontrollservers und des Switches benötigt das GPU-System 93,5 mJ/Bild.

Somit ist die NAA-Lösung sogar **1,61-mal effizienter.**

Plattform	Energieverbrauch pro Bild
GPU-System (Server mit Tesla V100)	**76 mJ**
Davon allein GPU	56 mJ
FPGA-Gesamtsystem PCIe-gekoppelt (BittWare 385A)	74,8 mJ
FPGA-NAA-Gesamtsystem (BittWare 385A)	58,2 mJ
Davon allein FPGAs	**37,5 mJ**

Eine weitere Steigerung der Energieeffizienz der NAA-Architektur kann durch eine Erhöhung der Packungsdichte erreicht werden, das bedeutet mehr NAAs pro Gehäuse. Das verwendete Gehäuse erlaubt eine Verdoppelung der genutzten NAAs auf 16. Durch die bessere Netzteilauslastung des NAA-Gehäuses wird die Energieeffizienz gesteigert. Da der Kontrollserver kaum CPU-Last aufweist, können diese 16 NAAs voraussichtlich weiterhin von einem Server mit ähnlichem Energieverbrauch betrieben werden. Dies führt gleichfalls zur Steigerung der Energieeffizienz.

Zur Verbesserung der Datenbankleistung wurde das Programm Telegraf zur Speicherung der Messwerte im verwendeten Datenbanksystem (InfluxDB) eingesetzt. Dies führt zu einem einfachen Systemdesign und einer drastischen Reduzierung der CPU-Last des Datenbankservers, was die Skalierbarkeit deutlich verbessert.

2.8 Webbasierter Demonstrator

Die erzielten Energieeinsparungen der NAA-Architektur auf FPGA-Basis werden im Vergleich zu einer GPU- und CPU-Architektur dargestellt und für eine entsprechende Anzahl von Inferenzen, also Anwendungen des KI-Modells, interpoliert. Die im generierten Diagramm dargestellten Werte basieren dabei auf den einzelnen real vorgenommenen Stromverbrauchsmessungen.

Im Folgenden ist das Auswahl-Widget dargestellt, das als Web-Interface auf der Website des Future Energy Lab (https://future-energy-lab.de/projects/energieeffiziente-ki/) zu finden ist. Damit lassen sich die im Projekt
ermittelten Ergebnisse interaktiv ansehen und nachvollziehen.

Die einzelnen Schaltflächen haben dabei folgende Bedeutung und können interaktiv geändert werden:

Server-Standort
Um eine globale Verwendung eines Beschleunigersystems zu simulieren, kann der Standort des Servers ausgewählt werden. Da jedes Land eine andere Kohlenstoffintensität pro kWh Strom hat, ist die schlussendliche CO_2-Ersparnis abhängig von diesen Parametern.

Anzahl der Inferences in Mio.
Hiermit kann ausgewählt werden, wie oft die Inferenz aufgerufen werden soll. Einstellbar ist ein Vielfaches von Tausend, da davon ausgegangen werden muss, dass eine Inferenz sehr häufig ausgeführt wird. Ein Beispiel dafür kann die millionenfache Suche in einer Bilddatenbank sein.

Größenordnung der Inferences
Hiermit kann das Diagramm skaliert werden.

Strompreis in €/kWh
Hiermit kann der anzunehmende Strompreis von 0 Euro bis 0,5 Euro voreingestellt werden.

Basierend auf den Ausführungen und Annahmen im vorhergehenden Abschnitt, können nun über einzelne Schaltflächen die Energiebedarfe der zum Vergleich gemessenen Rechnersysteme in das Diagramm übernommen werden. Folgende Auswahl kann getroffen werden:

FPGA – Energiebedarf des FPGA allein
FPGA NAA – Energiebedarf des FPGA als NAA im Rechenzentrum integriert
FPGA PCI – Energiebedarf des FPGA als Einsteckkarte im PC/Server integriert
GPU – Energiebedarf einer Nvidia Tesla V100 GPU allein
GPU-System – Energiebedarf einer Nvidia Tesla V100 GPU als Einsteckkarte inklusive PC/Server

Energieeffiziente Rechenarchitektur für KI

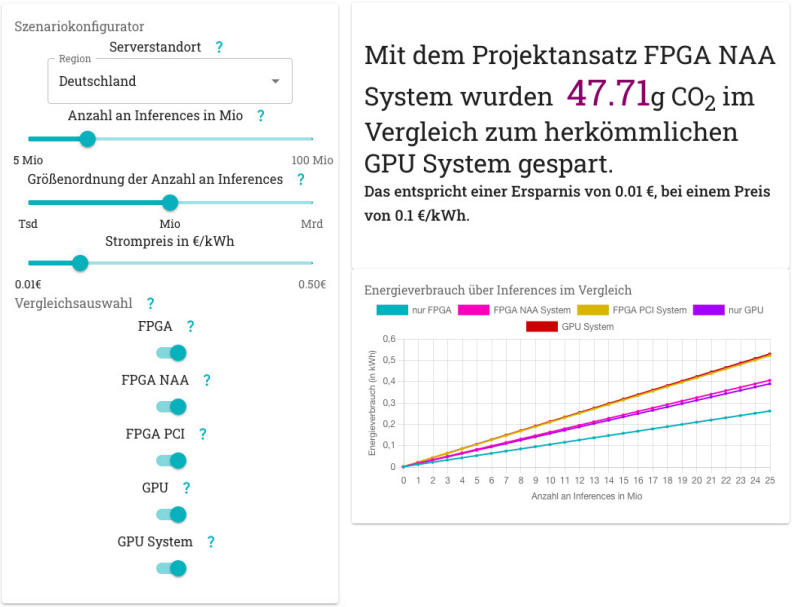


Abbildung 13: Auswahl-Widget

3. Energieeffizienz bei der Übertragung von KI-Modellen

Neben der Ausführung ist die Übertragung neuronaler Netze ein wichtiger Prozessschritt für Energieeinsparungen in KI-Anwendungen. Auch wenn zunächst das Training und die darauffolgende Anwendung von KI-Modellen oftmals im Vordergrund stehen, müssen die trainierten Modelle in vielen KI-Anwendungen an Geräte übertragen, das heißt als Datenpaket gesendet werden. Das ist zum Beispiel der Fall, wenn das Training eines neuronalen Netzes und seine Anwendung (die sogenannte Inferenz) auf unterschiedlichen Geräten erfolgen. Eine Übertragung neuronaler Netze wird auch dadurch notwendig, dass der Trainingsprozess wesentlich rechenintensiver ist als die Anwendung. Daher ist es für viele Geräte aufgrund der Hardware-Einschränkungen nur möglich, trainierte Modelle auszuführen, nicht aber selbst zu trainieren.

Wie bereits in der Einleitung beschrieben, haben sich neuronale Netze hin zu immer komplexeren Strukturen entwickelt. In den letzten Jahren hat sich diese Entwicklung durch die Verbreitung generativer Textmodelle noch einmal stark beschleunigt: Hier hat sich etwa die Anzahl an Parametern (z. B. im GPT3-Modell) im Vergleich zu bisherigen Modellen wie VGG16 vertausendfacht. Entsprechend entsteht bei der Übertragung eines trainierten Modells an ein Zielgerät zur Ausführung eine große Datenmenge, und das bereits bei einmaliger Übertragung. Die Entwicklung von KI-Anwendungen vervielfacht zudem diesen Datentransfer. Da neuronale Netze auch weitertrainiert werden, etwa weil neue relevante Daten für das Training zur Verfügung stehen, findet eine Übertragung neuronaler Netze nicht nur einmal, sondern regelmäßig statt. Eine weitere Dimension kommt durch nutzer- und anwendungsangepasstes Training hinzu, da nun individuelle neuronale Netze an jeden Nutzer und jede Anwendung übertragen werden, die speziell darauf zugeschnitten sind. Beispiele hierfür sind etwa individuelle Vorschläge für Texte, Empfehlungen für Filme oder Produkte oder auch im Bereich autonomes Fahren die Anpassung an lokale Verkehrssituationen (neue Umleitungen, aktuelles Verkehrsaufkommen und Ähnliches). Damit ergibt sich eine weitere Vervielfältigung des übertragenen Datenaufkommens.

Ähnliches gilt auch für die neueren verteilten KI-Anwendungen, wie das föderierte Lernen (FL – Federated Learning), bei dem mehrere Teilnehmer parallel ein globales Modell trainieren. Dafür trainiert jeder Teilnehmer mit seinen Trainingsdaten ein lokales Modell und schickt es sodann an einen zentralen Server, der alle empfangenen Modelle zu einem neuen globalen Modell für die nächste Trainingsiteration zusammenfasst (z. B. durch Mittelwertbildung pro Parameter). Dieser Prozess wird so häufig wiederholt, bis das Modell die gewünschte Güte bzw. Qualität hat.

Aufgrund der erwarteten Verbreitung dieses Trainingsansatzes wurden im Projekt Energieeinsparpotenziale bei Prozessen der Übertragung von KI-Modellen von einem Sender zu einem Empfänger untersucht und analysiert. Wie bereits in der Einleitung beschrieben, kommen als KI-Modelle üblicherweise tiefe neuronale Netzwerke zur Anwendung, die in sequenziell angeordneten Schichten aufgebaut sind. Dabei stellen die Verbindungen zwischen den Input- und den Output-Neuronen einer KNN-Schicht trainierbare Gewichte dar, die üblicherweise als Transformationsmatrix dargestellt werden. Diese trainierbaren Gewichte müssen nun übertragen werden.

Für die Untersuchungen im Projekt wurde davon ausgegangen, dass ein konkretes Modell mit seinen Parametern vorliegt, das möglichst effizient von einem Sender zu einem Empfänger übertragen werden soll. Ein Beispiel hierfür könnte sein, dass für ein autonom fahrendes Fahrzeug ein durch eine verbesserte Datenlage aktualisiertes Modell auf einem Server vorliegt, das an das Fahrzeug zur Optimierung seiner Entscheidungsfindung übermittelt werden soll. Eine weitere konkrete Anwendung, bei der KI-Modelle bzw. ihre Aktualisierungen von einem Sender zu einem Empfänger verschickt werden, ist der beschriebene verteilte Lernansatz. Entsprechend wurden die Untersuchungen im Projekt nach Datenaufkommen für die folgenden Anwendungen gestaffelt:

- Verteilung eines trainierten Modells: einmaliges Senden einen KI-Modells
- Fortlaufende Aktualisierung (durch neue Daten): regelmäßiges Senden von KI-Modellen bzw. ihren Updates
- Nutzerangepasste und verteilte KI-Anwendungen: regelmäßiges Senden von KI-Modellen parallel zwischen vielen Geräten

Bevor die Ergebnisse zur energieeffizienten Übertragung beschrieben werden, folgt zunächst die Darstellung der eingesetzten Verfahren der Datenkompression sowie der Testszenarien.

3.1 Direkte Übertragung neuronaler Netze

Wie bereits im Einleitungskapitel dargestellt, sind KI-Anwendungen immer komplexer geworden, das heißt, ihre Modelle haben inzwischen Millionen von trainierbaren Parametern. Ein Beispiel hierfür aus der Bildanalyse ist das gebräuchliche VGG16-Modell, ein neuronales Netz mit 21 Schichten, von denen 16 Schichten zusammen über 138 Millionen Gewichte bzw. Parameter enthalten. Die restlichen fünf Schichten, die sogenannten Max-Pooling-Layer, haben keine eigenen Parameter, sondern leiten nur jeweils einen maximalen Wert aus vier Aktivierungen der vorangegangenen Schicht weiter. Standardmäßig wird jeder Parameter als Fließkommazahlen (Datentyp FLOAT) dargestellt und belegt damit 4 Byte. Damit kommt das VGG16-Modell auf eine Größe von 138 Millionen Parametern × 4 Byte, also mehr als 550 MB. Damit müssten schon bei einmaliger Übertragung die-

ses Modells 550 MB übertragen werden. Andererseits sind bei der Übertragung der Originaldaten keine Vor- und Nachverarbeitung notwendig, hier insbesondere die Kompression am Sender und die Dekompression am Empfänger, die im Rahmen der Energieeffizienzuntersuchungen bei codierter Übertragung mit einbezogen werden müssen. Während die Übertragung eines originalen VGG16-Modells in einigen Anwendungen, insbesondere ohne Echtzeitanforderung, noch nutzbringend durchführbar wäre, hat sich mit den aktuellen Sprach- und Textgeneriermodellen (wie GPT3) die Anzahl der trainierbaren Parameter mehr als vertausendfacht. Das heißt, hier geht es um Datenmengen der Originalmodelle von 130 bis 200 GB, sodass neben der Energiebilanz überhaupt die Machbarkeit eine wichtige Rolle spielt.

In zeitkritischen Anwendungen wiederum sind bereits bei der Übertragung kleinerer KI-Modelle neben der zu übertragenden Datenmenge auch die Latenzanforderungen und die zur Verfügung stehende Bandbreite mit in die Gesamtbilanz einzubeziehen. Ist die Latenzanforderung hoch, jedoch die Bandbreite gering, wie etwa in kritischen Anwendungen wie dem autonomen Fahren, so müssen in jedem Fall eine Kompression und eine Dekompression erfolgen und damit muss zunächst Energie aufgewendet werden, um das Modell so stark zu komprimieren, dass die Latenzanforderung unter der gegebenen Bandbreite eingehalten werden kann. Daher wird im folgenden Kapitel zunächst die angewendete Codier-Methode beschrieben, die eine entsprechende Reduktion der zu übertragenden Datenmenge ermöglicht.

3.2 NNC (Neural Network Coding)

Das Ziel für eine energieeffiziente Übertragung von KI-Modellen ist es, die zu übertragende Datenmenge durch entsprechende Kompressionsmethoden stark zu reduzieren. Im Ergebnis soll sich bei codierter Übertragung eine geringere Gesamtenergie ergeben als bei uncodierter Übertragung. Die dafür zu berücksichtigenden Verarbeitungsschritte sind in Abbildung 14 dargestellt. Während für die Übertragung des Originalmodells nur der Energieaufwand der Übertragung selbst zu berücksichtigen ist, muss bei der komprimierten Übertragung neben dem Energieaufwand für die eigentliche Übertragung des komprimierten KI-Modells auch der zusätzliche Energieaufwand für die Kompression am Sender und die Dekompression am Empfänger einbezogen werden. Damit besteht das Ziel darin, ein Kompressionsverfahren einzusetzen, das mehr Energie durch die komprimierte KI-Übertragung einspart, als es zusätzlich für Kompression und Dekompression benötigt.

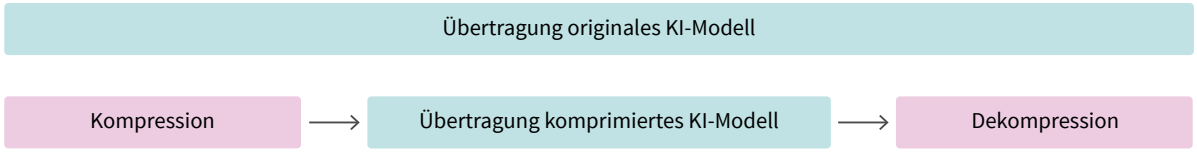


Abbildung 14: Verarbeitungsschritte für die Energiebilanz bei originaler KI-Modell-Übertragung (oben) und komprimierter Übertragung (unten)

Dafür wurde in den letzten Jahren die Grundlage im Rahmen von internationalen Standardisierungsarbeiten geschaffen. Insbesondere wurde von der ISO (International Standardisation Organisation) ein entsprechender Standard entwickelt, an dem auch die nationalen Gremien wie das DIN (Deutsches Institut für Normung) teilnehmen. Dieser sogenannte NNC-Standards (mit Normungsnummer ISO/IEC 15938-17, siehe auch Kirchhoffer et al. (2021) für einen Überblick) hat primär zum Ziel, eine hohe Kompressionsrate für neuronale Netze möglichst ohne Qualitätsverluste zu erzielen. Dabei bezieht sich hier die Qualität auf die Klassifizierungsgenauigkeit. Das heißt, der NNC-Standard hatte lediglich eine hohe Kompressionsfähigkeit zum Ziel, nicht jedoch Gesamtenergiebetrachtungen wie in diesem Projekt vorgenommen. Um das Kompressionsziel zu erreichen, verwendet NNC eine Reihe von Codierungs-Werkzeugen und -Methoden, unter anderem zur Parameterreduktion, Quantisierung und Entropiecodierung. Je nach Bedarf und Anwendungsszenario können diese Methoden individuell miteinander kombiniert werden. Zudem kann der Standard problemlos in gängige Industrieformate für neuronale Netze integriert werden. Wie bereits andere Multimedia-Standards zur Datenkompression spezifiziert auch der NNC-Standard nur den Bitstrom und den Decoder als normative Bestandteile, während Methoden zur Encodierung als nicht normative Beispiele beschrieben sind. Damit kann der Encoder von jeder nutzenden Person oder jedem Hersteller selbst implementiert und gegebenenfalls optimiert werden. Die allgemeine NNC-Struktur mit Encoder und Decoder ist in Abbildung 15 dargestellt.

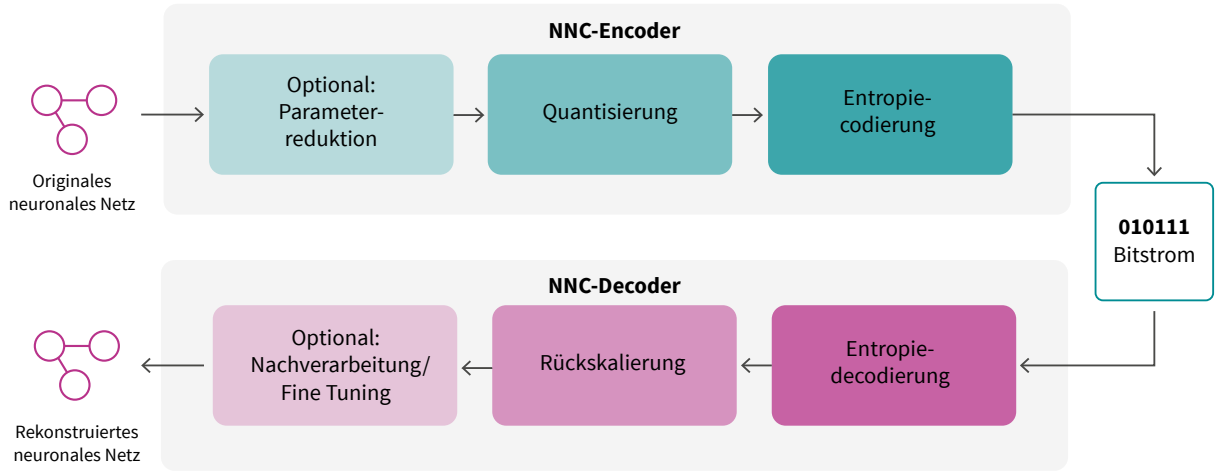


Abbildung 15: Schematische Darstellung der NNC-Struktur mit Encoder und Decoder

Zur Veranschaulichung der Effizienz des neuen NNC-Standards sind in Tabelle 1 Codier-Ergebnisse für unterschiedliche neuronale Netze aufgeführt. Dabei sind die ersten drei Netze mit den Bezeichnungen VGG16, ResNet50 und MobileNetV2 für die Bildklassifizierung und das vierte Netz, DCase, für die Audioklassifizierung trainiert worden.

Modell	Kompression	Top-5-Genauigkeit, rekonstruiertes NN	Top-5-Genauigkeit, originales NN	Originalgröße
VGG16	2,98 %	89,54 %	89,85 %	553,43 MB
ResNet50	6,54 %	91,80 %	92,15 %	102,55 MB
MobileNetV2	12,18 %	90,06 %	90,27 %	14,16 MB
DCase	4,12 %	92,35 %	91,85 %	467,26 KB

Tabelle 1: Codier-Ergebnisse ohne Qualitätsverlust für unterschiedliche neuronale Netze

Für jedes neuronale Netz sind in Tabelle 1 die erreichte Kompression, die Top-5-Genauigkeiten bei der Bild- bzw. Audioklassifikation für das rekonstruierte und decodierte neuronale Netz sowie die Originalgröße in Byte angegeben. VGG16 kann beispielsweise auf 2,98 Prozent seiner Originalgröße komprimiert werden, also von 553 auf ca. 16,5 MB, bei nahezu gleicher Klassifizierungsgenauigkeit von > 89 Prozent für die Top-5-Genauigkeit. Diese Genauigkeit bezieht sich auf den Prozentsatz an gewünschten Ergebnissen (z. B. alle Hunde aus einer Tierbilder-Datenbank), den das neuronale Netz in den ersten 5 Prozent (Top 5) der ähnlichsten Daten findet. In Tabelle 1 ist diese Genauigkeit sowohl für das originale als auch für das rekonstruierte neuronale Netz angegeben, um die gleichbleibende Klassifizierungsqualität bei den jeweiligen Kompressionen zu zeigen. Diese wiederum geben an, auf welchen Prozentsatz seiner Originalgröße das jeweilige neuronale Netz ohne Qualitätsverlust komprimiert werden kann. Als Beispiel sei hier wieder VGG16 herausgegriffen: Das originale Netz hat bei einer Auflösung von 4 Byte pro Parameter eine Gesamtgröße von mehr als 553 MB, was bereits einem Zehntel des durchschnittlichen Mobilfunk-Datenvolumens pro Monat in Deutschland im Jahr 2022 entspricht (Statista, abgerufen am 16.09.2023). Mit einer Kompression auf 2,98 Prozent kann der Bitstrom auf 16,5 MB reduziert werden. Zudem erreicht NNC wesentlich höhere Kompressionen von unter 1 Prozent bei verlustbehafteter Codierung, das heißt, wenn ein Verlust an Klassifizierungsgenauigkeit in Kauf genommen wird.

3.2.1 Parameterreduktion

Standardmodelle, wie zum Beispiel das oben vorgestellte VGG16-Modell, weisen Millionen von trainierbaren Parametern auf und gerade bei kleinen Trainingsdatensätzen führt diese Überparametrisierung häufig zum sogenannten Overfitting, also einer Art Auswendiglernen bzw. Codierung des Trainingsdatensatzes. Aber auch Modelle, die über eine gute Verallgemeinerung auf ungesehenen Daten verfügen, sind meist überparametrisiert; das heißt, es existieren oft kleinere Modelle mit

wesentlich weniger Parametern, die dieselbe Aufgabe mit annähernd gleicher Genauigkeit erledigen. Um dieses Problem zu lösen, werden Methoden zur Parameterreduktion verwendet, die komplexe überparametrisierte KNN in kleinere Modelle überführen. Hierfür wurden unterschiedliche Verfahren entwickelt, die im Folgenden näher vorgestellt werden.

Sparsifikation: Bei diesem Verfahren versucht man, diejenigen Gewichte bzw. Parameter eines KNN zu null zu setzen, die eine geringe Relevanz für das Gesamtverhalten des KNN (z. B. die Klassifikationsgüte) haben. Aufgrund der komplexen Zusammenhänge innerhalb eines tiefen KNN ist es bei diesem Verfahren schwierig, die Relevanz eines jeden einzelnen Parameters oder einer Gruppe von Parametern (z. B. eines konkreten Filters) effizient zu ermitteln, da zunächst kein direkter Zusammenhang zwischen dem absoluten Zahlenwert eines Parameters und seinem Einfluss auf das Gesamtverhalten des KNN besteht. Daher führen insbesondere Methoden, die betragsmäßig kleine Parameter zu null setzen, häufig nicht zu akzeptablen Ergebnissen. Zudem nimmt auch ein zu null gesetzter Wert zunächst noch Speicherplatz in Anspruch und muss schlussendlich für die Übertragung codiert werden.

Pruning: Im Gegensatz zur Sparsifikation werden beim Pruning alle irrelevanten Parameter aus dem KNN gelöscht, sodass sie auch keinen Speicher mehr einnehmen. Allerdings ergibt Pruning nur dann Sinn, wenn es im Ausgangsmodell in regelmäßigen oder strukturierten Teilen erfolgt, zum Beispiel indem ganze Filter oder Neuronen mit ihren Verbindungen zu anderen Neuronen gelöscht werden. Die unstrukturierte Löschung einzelner Gewichte bzw. Parameter ist nicht sinnvoll, da die Position der unstrukturiert gelöschten Parameter wiederum gespeichert und codiert werden muss. Häufig geht das Pruning mit signifikanten Einbußen bei der Güte des KNN einher, insbesondere dann, wenn man nach dem strukturierten Löschen kein Retraining des KNN vornimmt.

Matrixzerlegung: Das Ziel dieser Methode ist es, die Transformationsmatrix einer Schicht durch ein Produkt von zwei Matrizen mit wesentlich weniger Parametern zu approximieren, um die Ausgangsmatrix effizienter übertragen zu können. Wenn also beispielsweise eine $(n \times m)$-dimensionale Ausgangsmatrix in zwei Matrizen mit den Dimensionen $(n \times k)$ und $(k \times m)$ zerlegt werden kann, so hat man anstatt $n \times m$ Parameter nun $k \times (n+m)$ Parameter, was sich gerade bei einem kleinen k bemerkbar macht. Im Allgemeinen ist eine solche Zerlegung gerade für kleine k nur dann möglich, wenn anstelle der Ausgangsmatrix eine Approximation verwendet wird. Bei diesem Verfahren muss üblicherweise ein rechenintensives Optimierungsproblem gelöst werden.

Batchnorm-Faltung: Wie bereits für das VGG16-Modell beschrieben, sind nicht alle Schichten eines Netzwerks mit Parametern behaftet, sondern einige führen lediglich die Maximierung oder Normalisierung von Aktivierungen vorangegangener Schichten durch. Im speziellen Fall der sogenannten Batchnorm-Schicht findet dort eine Normalisierung für die Mittelwerte und Standardabweichungen vorangegangener Aktivierungen statt. Bei der Batchnorm-Faltung wird diese Normalisierung direkt mit den Parametern der davorliegenden Schicht fusioniert. Es werden also zwei Transformationsmatrizen miteinander multipliziert, wodurch die ursprünglichen zwei Schichten für die Codierung zu einer zusammengefasst werden. Allerdings ist zu beachten, dass dieser Schritt nicht reversibel ist.

3.2.2 Quantisierung

Generell wird unter Quantisierung die Approximation einer großen Menge durch eine kleine Menge verstanden. Dementsprechend besteht das Ziel der Quantisierung von KNN darin, die Parameter bzw. Gewichte in niedrigerer Präzision darzustellen und zu codieren. So ist es beispielsweise möglich, die Parameter nicht in der während des Trainings üblicherweise verwendeten 32-Bit- oder 64-Bit-Gleitkommazahlen (Floating Point) darzustellen, sondern in Festkommazahlen oder ganzzahlig mit entsprechender Skalierung. Im Folgenden werden die gebräuchlichsten Formen der Quantisierung näher vorgestellt. Es sei aber bereits hier erwähnt, dass eine Quantisierung der Netzparameter nach dem Training oftmals zu erheblichen Einbußen bei der Güte des KNN führt, was wiederum durch Retraining oder Quantisierungsbewusstes Training (QAT – Quantization Aware Training) ausgeglichen werden kann. Beim QAT wird die vorgesehene Quantisierung des KNN bereits während des Trainings durch sogenannte Fake-Quantisierung (Quantisierung mit sofortiger Dequantisierung) berücksichtigt, sodass sich das KNN bereits in der Trainingsphase darauf einstellen kann.

Gleichmäßige skalare Quantisierung: Bei diesem Verfahren wird der durch Kalibrierung ermittelte Wertebereich der zu quantisierenden Parameter bzw. Aktivierungen durch abstandsgleiche Rekonstruktionswerte repräsentiert, sodass sich die Rekonstruktion jedes quantisierten Ausgangswerts durch einfache Multiplikation des Quantisierungsindex mit der Quantisierungsschrittweite ergibt (sogenannte Dequantisierung). Die Differenz zwischen Ausgangswert und dazugehörigem Rekonstruktionswert entspricht dem Quantisierungsfehler. Diese Methode führt oftmals zu großen Fehlern, da sie die Verteilung über den Wertebereich vernachlässigt. Sie wird aber aufgrund ihrer einfachen Implementierbarkeit dennoch sehr oft eingesetzt.

Codebuch-Quantisierung: Bei dieser Form der Quantisierung werden die Ausgangswerte beliebig über den Wertebereich verteilten Clustern als Rekonstruktionswerte zugewiesen, wodurch man der tatsächlichen Verteilung der Werte über dem Wertebereich besser gerecht werden und somit den Quantisierungsfehler verringern kann. Da bei diesem Verfahren die Lage

der Clusterzentren meist durch Optimierungsmethoden aufwendig ermittelt wird und zudem zusätzlich codiert werden muss, wird in der Praxis oft auf die oben beschriebene einfachere gleichmäßige Quantisierung zurückgegriffen.

Abhängige Quantisierung: Diese spezielle Vektorquantisierungsmethode verbessert die klassische gleichförmige skalare Quantisierung. Dazu werden anstatt nur einem zwei Quantisierer verwendet und die jeweilige Zuordnung zu einem der beiden findet auf Basis der zuvor getroffenen Quantisierungsentscheidungen statt. Damit erreicht dieses Verfahren kleinere Quantisierungsfehler, geht jedoch mit einer Abhängigkeit zwischen zwei aufeinanderfolgenden Quantisierungs- bzw. Rekonstruktionsschritten einher.

3.2.3 Entropiecodierung

Bevor Modelle vom Sender zum Empfänger verschickt werden, durchlaufen sie einen oder mehrere der oben beschriebenen Vorverarbeitungsschritte zur Parameterreduktion und Quantisierung. Anschließend werden die resultierenden Modelle meist noch verlustfrei komprimiert, entweder durch Überführung in ein effizienteres Format (z. B. das Format Compressed Sparse Row/Column (CSR/CSC)) oder durch Verwendung der Entropiecodierung zur Erzeugung eines Bitstroms für die effiziente Modellübertragung. Die Entropiecodierung setzt einen Encoder auf Senderseite mit entsprechendem Decoder auf Empfängerseite voraus und basiert auf der Grundidee, Symbolen, die öfter auftreten, also weniger Informationsgehalt repräsentieren, entsprechend weniger Bits zuzuweisen. Ein sehr bekanntes Verfahren zur Entropiecodierung ist die Huffman-Codierung, die jedem Quellsymbol ein Codewort zuordnet. Als sehr effiziente Erweiterung der Huffman-Codierung hat sich die arithmetische Codierung herausgestellt, bei der mittels Wahrscheinlichkeitsschätzungen für die Quellsymbole die gesamte Nachricht oder ein Teil von ihr in Form einer rationalen Zahl codiert werden kann.

DeepCABAC: Das Basisverfahren für diese Methode bildet das am Fraunhofer HHI entwickelte CABAC (Context-Based Adaptive Binary Arithmetic Coding), das eine Weiterentwicklung der klassischen arithmetischen Codierung darstellt. CABAC ist als effiziente Methode zur verlustfreien Komprimierung Bestandteil weltweiter Videocodierstandards. Das Verfahren wurde für die Kompression neuronaler Netzwerke angepasst (Wiedemann et al., 2020). DeepCABAC erlaubt mit seiner adaptiven, kontextbasierten Ratenmodellierung eine optimale Quantisierung und Codierung der Parametermatrizen eines neuronalen Netzes und somit eine sehr effiziente Kompression ohne Güteverluste.

3.3 Föderiertes Lernen als Anwendungsfall zur systematischen Untersuchung

Föderiertes Lernen (FL – Federated Learning) bezeichnet eine maschinelle Lernumgebung, in der viele Teilnehmer (z. B. mobile IoT-Geräte oder ganze Organisationen) kollaborativ ein globales KI-Modell unter der Orchestrierung eines zentralen Servers trainieren, während die Trainingsdaten dezentralisiert bleiben (Kairouz et al., 2021). Beim föderierten Lernen wird also die komplexe und zeitaufwendige Trainingsphase auf die Teilnehmer verteilt und damit parallelisiert. Dafür vereinbaren die Teilnehmer zunächst in dem unter ihnen gebräuchlichsten FL-Protokoll eine Netzarchitektur für das globale Modell. Jeder Teilnehmer trainiert dann eine lokale Version des Modells (mit einer dem globalen Modell gleichenden Architektur) mit seinen lokalen Trainingsdaten. Da sich die Trainingsdaten der einzelnen Teilnehmer in der Praxis üblicherweise statistisch unterscheiden, werden hierdurch auch leicht unterschiedliche Versionen des neuronalen Netzes erzeugt. Diese lokalen Versionen werden anschließend an das lokale Training von allen Teilnehmern an einen zentralen Server geschickt, der eine neue, einheitliche Version des neuronalen Netzes aus allen erhaltenen Varianten erzeugt, zum Beispiel als parameterweise Mittelwerte aller Versionen. Diese neue Version wird wiederum an alle Beteiligten verteilt, woraufhin eine neue Lern- oder Kommunikationsrunde beginnt. Einen schematischen Überblick über das beschriebene Vorgehen gibt Abbildung 16.

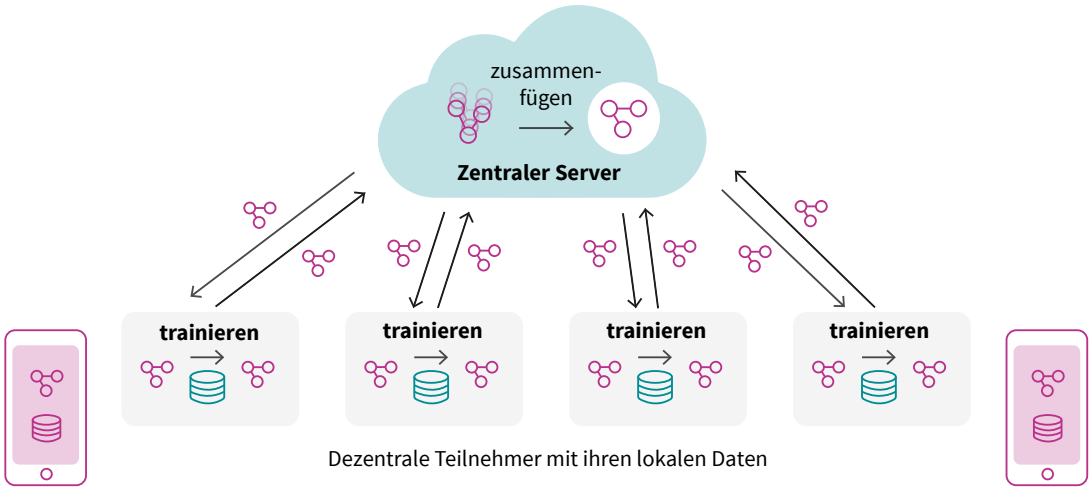


Abbildung 16: Schematische Übersicht des föderierten Lernens

Damit ergeben sich beim FL durch die wiederholte Übertragung neuronaler Netzwerk-Updates Echtzeitanforderungen, die die Anwendung von Methoden zur Komprimierung der übertragenen Modellparameter unerlässlich machen. Dies gilt insbesondere für Anwendungsfälle, in denen Millionen von mobilen Geräten gleichzeitig am Lernverfahren teilnehmen und dabei oftmals nur über eine langsame Verbindung mit dem Server kommunizieren. Diese Anwendungsfälle werden in der Literatur häufig Cross-Device-FL genannt. Der Ansatz des FL mit seinen spezifischen Anforderungen an Latenz und seinen Herausforderungen bezüglich der Bandbreite stellt damit eine ideale Umgebung für die systematische Untersuchung von Energieeinsparpotenzialen bei der Übertragung von KI-Modellen dar.

Erste theoretische Untersuchungen zur Energieeffizienz des föderierten Lernens wurden bereits durch Sattler et al. (2019) durchgeführt und sind in der zu Beginn des Projekts verfassten Kurzstudie der dena „Auf dem Weg zu energieeffizienter künstlicher Intelligenz – Welche Energieeinsparpotenziale bieten KI-Anwendungen?“ (Deutsche Energie-Agentur (dena), 2022) beschrieben. Hier konnte ein hohes Energiesparpotenzial theoretisch aufgezeigt werden. Auf der Basis auch dieser Arbeiten wurden daher systematische Untersuchungen in realistischen Szenarien durchgeführt. Die aufgezeigten hohen Kompressionsraten erfordern jedoch zunächst einen zusätzlichen Energieeinsatz auf der Senderseite, womit sich die folgenden Forschungsfragen für das EEKI-Projekt ergeben haben:

- Welche Energiebilanz weisen die einzelnen Kompressionsmethoden auf und welche Methoden sind daher bei gegebener Latenz und Bandbreite aus Energieeffizienzgründen vorzugswürdig?
- Kann durch Kompression insgesamt Energie eingespart werden?

Die erste Frage ist vergleichbar mit der Frage nach dem effizientesten Transportmittel, um in einer vorgegebenen Zeit von A nach B zu kommen. Die zweite Frage kann mit der Frage verglichen werden, ab wann sich ein Investment (hier die für die Kompression erforderliche Energie) lohnt bzw. (durch Energieeinsparungen bei der Gesamtübertragung) amortisiert.

Einen ersten Schritt zur Beantwortung dieser Fragen hat bereits die Arbeit von Qiu et al. (2021) geleistet. Sie beschreibt erstmalig eine systematische Untersuchung eines CO_2-Fußabdrucks für den dezentralen Lernansatz des föderierten Lernens und vergleicht diesen mit dem des klassischen zentralen Lernansatzes, bei dem alle Daten zentralisiert auf einem Server vorliegen. Die Arbeit zeigt, dass in Abhängigkeit vom verwendeten Energiemix (also dem Verhältnis umweltfreundlicher gegenüber nicht umweltfreundlicher Energieerzeugung) der dezentrale Lernansatz eine bessere Bilanz aufweisen kann als der zentrale Lernansatz. Darüber hinaus zeigt die Studie, dass der Anteil des Energieeinsatzes für die Kommunikation stark von der Größe des KI-Modells, der Größe des Trainingsdatensatzes bei jedem Teilnehmer und der von den Teilnehmern während des Trainings verbrauchten Energie abhängt. Mit diesen Vorarbeiten wurde nun im EEKI-Projekt die Energieeffizienz originaler und codierter Ansätze zur Übertragung von KI-Modellen in verteilten Umgebungen untersucht und je nach Länderauswahl für Sender- und Empfängerstandorte in CO_2-Äquivalente umgerechnet.

3.4 Untersuchungsergebnisse bei der Übertragung neuronaler Netze

In diesem Kapitel werden die Ergebnisse der Energieeffizienzuntersuchungen bei der Übertragung neuronaler Netze beschrieben und dargestellt. Dazu wird im folgenden Unterkapitel beschrieben, wie die Tests durchgeführt und welche Voraussetzungen geschaffen bzw. angenommen wurden. Während zunächst bei der Kompression der KI-Modelle die Kompressionsraten, also das Verhältnis zwischen Originalgröße und komprimierter Bitstrom-Größe, der Modelle direkt gemessen werden können, sind für die Energieberechnung bei der eigentlichen Datenübertragung bestimmte Annahmen zu treffen, die hier dargestellt sind. Gleiches gilt für die Berechnung der eingesparten CO_2-Menge, die sich aus dem Energiemix in den jeweiligen Ländern der Sender- und Empfängerstandorte ergibt, für die die Übertragung simuliert wurde. In den darauffolgenden Unterkapiteln 3.4.2 und 3.4.3 wird dann beispielhaft eine Auswahl der Ergebnisse dargestellt und analysiert. Die gesamten Testdaten sind wiederum Teil des Web-Demonstrators, der zur Veranschaulichung der Ergebnisse auf der Website des Future Energy Lab zu finden ist und der in Kapitel 3.5 beschrieben wird.

3.4.1 Testbeschreibung und Detailanalyse der Energieeinsparung mittels NNC bei einmaliger Übertragung

Testbeschreibung

Zur Untersuchung, ob und wie die Energieeffizienz bei der Übertragung neuronaler Netze gesteigert werden kann, wurden Methoden zum föderierten Lernen (FL) und zur Kompression von KI-Modellen eingesetzt, um die Kommunikation zwischen Sender und Empfänger innerhalb des FL-Systems zu reduzieren. Das konkrete Ziel der Untersuchungen war, eine positive Gesamtbilanz in Bezug auf Energieeinsparungen bei der komprimierten Übertragung von KI-Modellen zu erzielen. Dieser Frage wurde zunächst im einfachsten Fall der einmaligen Übertragung eines neuronalen Netzes von einem Sender zu einem Empfänger nachgegangen. Der Sender komprimiert und codiert die Parameter des neuronalen Netzes und überträgt es zum Empfänger, wodurch Energieaufwand für die Kompression und Codierung sowie für den Upload auf Senderseite und den Download auf Empfängerseite entsteht. Zusätzlich wird der Energieaufwand für die Decodierung des neuronalen Netzes auf Empfängerseite berücksichtigt. Anzumerken ist hierbei, dass der aktuelle NNC-Standard für die meisten KI-Modelle eine verlustbehaftete Komprimierung durchführt, das heißt, die rekonstruierten Gewichte am Decoder unterscheiden sich leicht von den originalen Gewichten vor der Kompression. Diese mathematische Abweichung entsteht durch die gewählte Quantisierung der Netzparameter. Gesteuert wird die Quantisierung durch einen Quantisierungsparameter (QP), der die Schrittweite bei der Quantisierung festlegt, also schlussendlich den Fehler bei der Rundung eines Netzparameters zu einem diskreten Rekonstruktionswert. Der QP bestimmt dabei den Abstand zwischen zwei Rekonstruktionswerten. Je größer dieser Abstand ist, desto größer ist auch der potenzielle Rundungsfehler, jedoch desto geringer ist auch die Zahl der zu übertragenden Bits für die Codierung des Rekonstruktionswertebereichs. Im Allgemeinen wird jedoch die Quantisierung derart gewählt, dass bei Anwendung des rekonstruierten Modells die gleiche oder eine nur geringfügig reduzierte Genauigkeit, zum Beispiel bei der Bilderkennung oder Segmentierung, erreicht wird. Damit wird zwar verlustbehaftet codiert, jedoch bei gleichbleibender Qualität.

In den durchgeführten Experimenten wurden vortrainierte neuronale Netze verwendet und der QP so gewählt, dass das rekonstruierte neuronale Netz auf Empfängerseite annähernd die gleiche Performance (Klassifikationsgüte) wie das Originalnetz auf Senderseite erreicht. Für die Ermittlung des Energieaufwands zur Übertragung von KI-Modellen wurden zunächst durchschnittliche Up- und Download-Geschwindigkeiten im Breitbandnetz Deutschlands wie folgt spezifiziert: 22,51 MBit/s für den Upload und 66,42 MBit/s für den Download. Die verwendete Hardware Nvidia Xavier NX wurde so konfiguriert, dass die mittlere Leerlaufleistung etwa 4 Watt beträgt.

Für das initiale Experiment wurde eine Standardarchitektur, das ResNet18-Modell, gewählt, das insbesondere zur Bildklassifikation verwendet wird. Dieses Modell besitzt etwa 11,5 Millionen trainierbare Parameter und einen Speicherbedarf von etwa 46,8 MB. Im Vergleich ergaben sich folgende Energiebilanzen für die einmalige Übertragung:

- Unkomprimierte Übertragung des ResNet18-Modells: 0,0878 Wh Energieaufwand zur Übertragung von 46,8 MB
- Komprimierte Datenübertragung mit NNC: Durch die Kompression konnte eine Reduktion der Datenrate von 46,8 MB auf 6,86 MB ohne erhebliche Performance-Einbußen erreicht werden. Der Energieaufwand für die komprimierte Übertragung wurde mit nur noch 0,0129 Wh ermittelt. Dem gegenüber wurde ein zusätzlicher Energieverbrauch von Encoder und Decoder von 0,0466 Wh gemessen. Damit ergibt sich für die komprimierte Übertragung eine benötigte Gesamtenergie von 0,0595 Wh.

Diese Werte sind für beide Übertragungsarten in Abbildung 17 dargestellt.

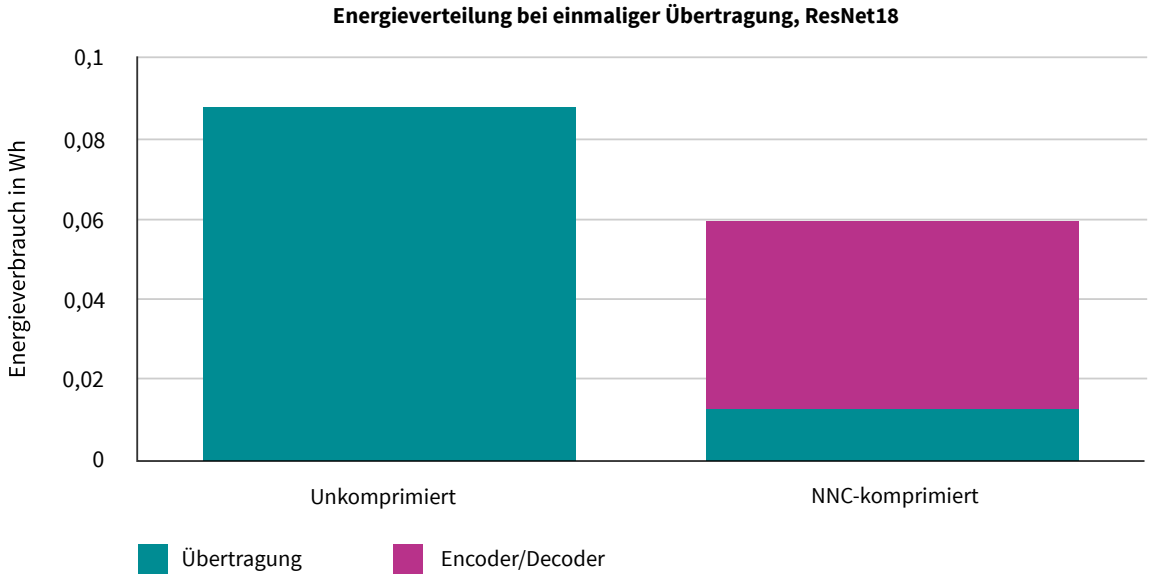


Abbildung 17: Energiebilanz bei einmaliger Datenübertragung: unkomprimierte Übertragung links und komprimierte Übertragung (rechts)

Bilanz: Schon bei einmaliger Übertragung konnte eine Energiereduktion von 0,0878 auf 0,0595 Wh ermittelt werden, wodurch sich eine Energieersparnis bei der Übertragung des komprimierten Netzes von 0,0283 Wh ergibt, also bereits 33 Prozent. Damit konnte bereits hier eine positive Energiebilanz ermittelt werden, die sich in den komplexeren Szenarien mit mehrfachem Senden und parallelem KI-Training mit entsprechendem Modellaustausch noch verstärkt, wie in den folgenden Kapiteln gezeigt.

Mehrmalige Übertragung mittels NNC

Im nächsten Schritt wurden die Experimente auf die mehrfache Übertragung von KI-Modellen ausgedehnt, wie sie für regelmäßige Updates und jedes Server-Client-Paar im verteilten oder föderierten KI-Training auftritt. Beispielhaft ist hier wieder das ResNet18-Modell mit 46,8 MB gezeigt.

Dazu wurden die Experimente und Messungen auf ein Testszenario des föderierten Lernens ausgedehnt. Zunächst wurde angenommen, dass fünf lokale Geräte gemeinsam am Training teilnehmen, das heißt fünf Klienten und ein Server im Bezeichnungsrahmen des föderierten Lernens. Des Weiteren wurde angenommen, dass 50 Runden trainiert wird. Für die Energiebilanzrechnung sind die folgenden Übertragungen zu berücksichtigen:

- Unkomprimierte Übertragung: Pro Trainingsrunde muss zunächst das gemeinsame Servermodell an fünf Klienten übertragen werden. Diese trainieren lokale Varianten und schicken fünf unterschiedliche Modellversionen an den Server. Dieser mittelt alle Varianten und schickt in der nächsten Runde wieder ein gemeinsames Modell an alle fünf Klienten. Damit ergeben sich pro Runde zehn Übertragungen (fünf Server-Klient- und fünf Klient-Server-Übertragungen) und damit für den gesamten Trainingsprozess mit 50 Runden 500 Übertragungen. Das ergibt für das gezeigte ResNet18-Modell mit 46,8 MB eine Gesamtübertragung von 23,4 GB. Zusätzlich wird Energie für das lokale Training bei allen Klienten benötigt.

- Komprimierte Übertragung: Auch für die komprimierte Übertragung ergeben sich 500 Einzelübertragungen, allerdings hier für die komprimierten Bitströme. Überschlagsmäßig kann hier wieder eine komprimierte Rate von 6,86 MB angenommen werden, sodass hier insgesamt nur 3,43 GB übertragen werden. Jede Übertragung erfordert hier zusätzlich einen Kompressionsvorgang beim Sender und einen Dekompressionsvorgang beim Empfänger. Im föderierten Lernszenario gilt nun Folgendes: Pro Trainingsrunde muss der Server einmal komprimieren (da es nur eine Serverversion des KI-Modells gibt) und jeder der fünf Klienten muss dekomprimieren (da es unterschiedliche Geräte sind). Nach dem lokalen Training muss jeder der fünf Klienten sein lokales Modell komprimieren und der Server muss jedes lokale Modell dekomprimieren, also fünfmal (da es unterschiedliche Versionen des KI-Modells sind). Damit ergeben sich pro Trainingsrunde 1 x Kompression + 5 x Dekompression in der Server-Klient-Kommunikation und 5 x Kompression + 5 x Dekompression in der Klient-Server-Kommunikation, also sechs Codier- und zehn Decodiervorgänge. Bei 50 Trainingsrunden wird damit 300-mal komprimiert und 500-mal dekomprimiert. Auch hier wird zusätzlich Energie für das lokale Training bei allen Klienten benötigt.

Die Gesamtenergiebilanz für beide Übertragungsvarianten ist in Abbildung 18 dargestellt.

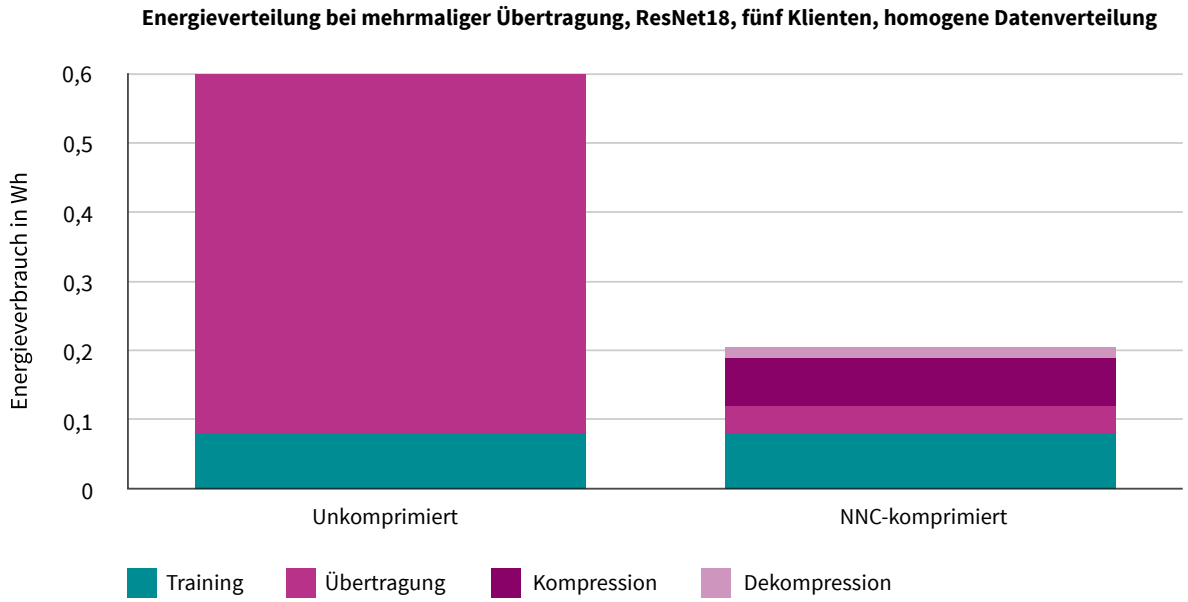


Abbildung 18: Gesamtenergiebilanz beim föderierten Lernen mit fünf Klienten und 50 Trainingsrunden bei unkomprimierter Übertragung (links) und komprimierter Übertragung (rechts)

Zunächst ist zu sehen, dass in beiden Varianten der Energieanteil des Modelltrainings selbst konstant bleibt, da er unabhängig von der Übertragung ist. Des Weiteren ist der Energieaufwand für die Übertragung in Rot dargestellt, der entsprechend bei komprimierter Übertragung wesentlich geringer ist. Für die komprimierte Übertragung (Abbildung 18, rechte Säule) ergeben sich zusätzliche Energieverbräuche durch die Kompressions- und Dekompressionsvorgänge. In der Gesamtbilanz ist zu beobachten, dass der Kompressionsvorgang wesentlich mehr Energie benötigt als der Dekompressionsvorgang, obwohl in der Gesamtbilanz nur 300-mal komprimiert und 500-mal dekomprimiert wird. Dies ist eine Eigenschaft klassischer Multimedia-Codierverfahren, da Encoder (für die Datenkompression) wesentlich komplexer als Decoder (für die Dekompression) sind. Das heißt, Encoder haben wesentlich mehr Rechenschritte, da zunächst die optimale Codiermethode im Encoder ermittelt wird. Dies geschieht durch Ausführen vieler Codiermodi mit vielen unterschiedlichen Kombinationen der Methoden. Der optimale Codiermodus wird dann mit an den Decoder gesendet. Hier wird dann lediglich der eine optimale Codiermodus ausgeführt, das heißt, diese eine spezielle Kombination an Methoden wird zur Datendekompression verwendet. Daher ist es vorteilhaft, möglichst die Anzahl an Kompressionen zu verringern. Dies ist für das föderierte Trainingsverfahren auch gegeben, wie in der Anzahl an benötigten Prozessen weiter oben dargestellt.

Mit allen benötigten Prozessen ergibt sich nun eine Gesamtbilanz von ca. 0,60 Wh für unkomprimierte Übertragung und von 0,21 Wh für komprimierte Übertragung, wie in Abbildung 18 zu sehen ist. Durch die Nutzung des NNC-Codec zur Kompression der Modelle während der Übertragung konnte somit der Gesamtenergieverbrauch um 65 Prozent reduziert werden.

Visualisiert man den Gesamtenergieverbrauch über die Kommunikationsrunden hinweg (Abbildung 19), sieht man zudem, dass der Verbrauch relativ konstant ist, sowohl bei komprimierter als auch bei differenziell-komprimierter Übertragung. Dabei ist auch zu sehen, dass der Energieverbrauch bei differenziell-komprimierter Übertragung geringer ist als bei komprimierter Übertragung.

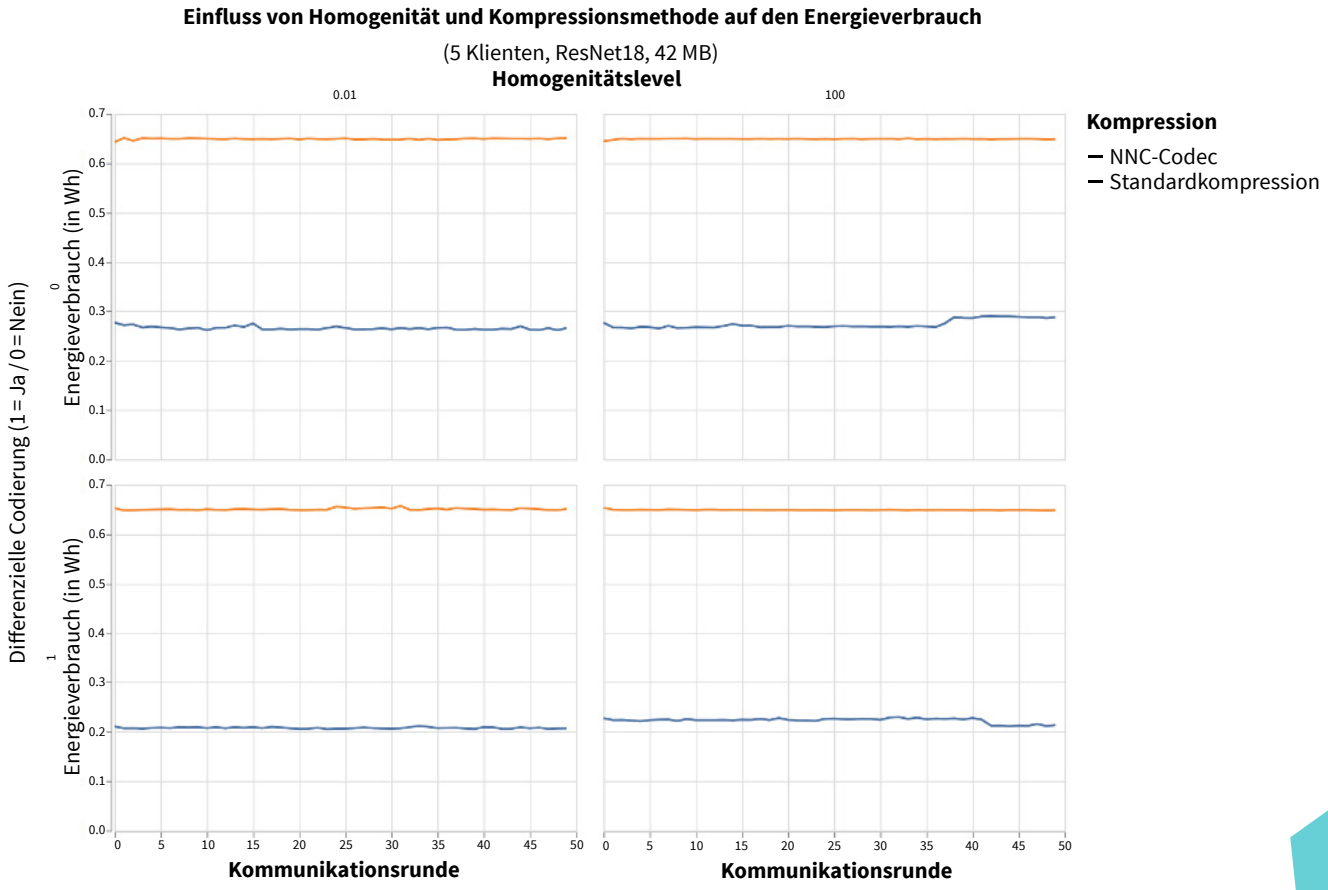


Abbildung 19: Energieaufwand für komprimierte und originale Übertragung über alle 50 Kommunikationsrunden bei heterogener Datenverteilung (links) und homogener Datenverteilung (rechts) sowie bei komprimierter Übertragung (oben) und differenziell-komprimierter Übertragung (unten)

Des Weiteren wurde untersucht, wie sich unterschiedliche Datenverteilungen auf den Energieaufwand für Kompression und Übertragung auswirken. Je nach föderiertem Trainingsszenario können dabei die lokalen Daten der Klienten-Geräte homogen oder heterogen verteilt sein. Zur Erklärung wird nun ein Beispiel mit lokalen Bilddatensätzen von Bauteilen angenommen, wobei ein Teil der Bilder korrekte bzw. intakte Bauteile zeigt und ein anderer Teil beschädigte oder defekte Bauteile. Hier wäre dann das Ziel des föderierten Lernens, das KI-Modell so zu trainieren, dass es Bilder korrekter von Bildern fehlerhafter Bauteile unterscheiden kann. Bei einer homogenen Datenverteilung hat nun jeder Klient eine etwa gleiche Anzahl von Bildern ähnlicher Bauteile sowie auch ein ähnliches Verhältnis an Bildern mit korrekten und fehlerhaften Bauteilen. Im Gegensatz dazu bedeutet eine heterogene Verteilung auch eine Ungleichverteilung von Bauteiltypen sowie des Verhältnisses fehlerhafter und korrekter Bauteile zwischen den Klienten. Das kann im Extremfall auch ein Szenario sein, bei dem ein Klient nur eine Art von Bauteilen hat, die kein anderer Klient aufweist. Dies kann allerdings dazu führen, dass der verteilte Trainingsprozess keine akzeptable Genauigkeit erreicht und dann nicht eingesetzt werden kann.

In den durchgeführten Tests kann die Datenhomogenität variiert werden. Wie in Abbildung 19 dargestellt, bedeutet ein Wert von 0,01 (links) eine sehr große Heterogenität, während ein Wert von 100 (rechts) eine homogene Verteilung darstellt. Im Ergebnis ist ersichtlich, dass die Datenhomogenität kaum Einfluss auf den Energieaufwand für originale, komprimierte und differenziell-komprimierte Übertragung hat. Damit konnte auch gezeigt werden, dass NNC als Kompressionsmethode sehr gute Ergebnisse unabhängig von unterschiedlich trainierten KI-Modellvarianten aufweist.

Wie bereits beschrieben, wird für die Kompression und Dekompression der KI-Modelle NNC derart verwendet, dass eine möglichst hohe (wenn auch mathematisch verlustbehaftete) Datenkompression ohne Qualitätseinbußen bezüglich der Genauigkeit des jeweiligen Modells für seine Anwendung erreicht wird. Dazu sind im Vergleich die erzielten Genauigkeiten für die komprimierte und die originale Übertragung jeweils als orange und rote Kurve über die 50 Trainingsrunden in Abbildung 20 dargestellt.

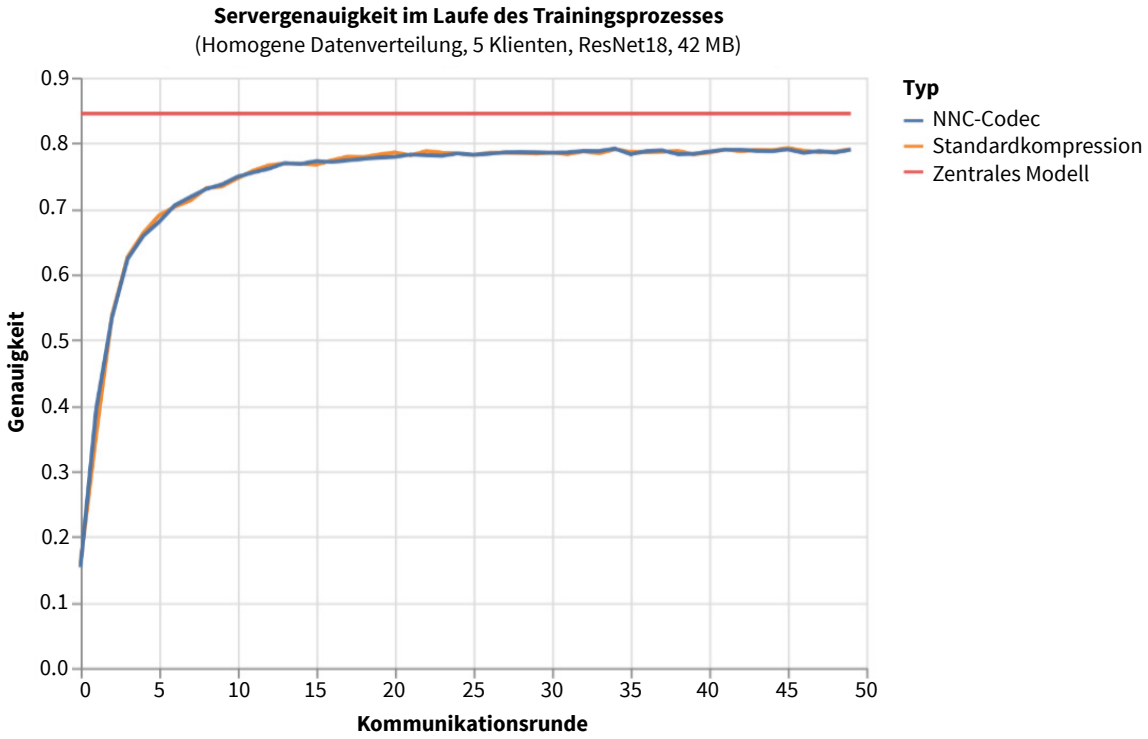


Abbildung 20: Erreichte Genauigkeiten für originale und komprimierte Übertragung beim föderierten Training (orange und rote Kurve) sowie als Referenz das konstante Modell (in Blau) für einen zentralen Ansatz (nicht verteiltes Lernen)

Hierbei ist zunächst zu sehen, dass beide Kurven sehr ähnliche Werte erreichen. Das heißt, es wird gezeigt, dass die Genauigkeit des KI-Modells bei komprimierter Übetragung erhalten bleibt. Grundsätzlich ist weiterhin zu sehen, wie die Genauigkeit mit jeder Trainingsrunde zunächst sehr stark steigt. Bis etwa Runde 25 wird damit mit jeder Trainingsrunde das Ergebis verbessert. Ab Runde 25 tritt dann eine Sättigung ein. Ab hier ist keine Verbesserung mehr zu beobachten, sodass eine reale Anwendung das Training hier beenden würde. Als Referenz ist zudem in Abbildung 20 das sogenannte zentrale Modell gezeigt. Dabei handelt es sich um einen Richtwert, bei dem die lokalen Daten, die individuell bei jedem Klienten vorhanden sind, in einem einzigen Trainingsszenario zusammengeführt werden und damit eine einzige vollständige Trainingsrunde erfolgt. Dieser Wert ist in den meisten föderierten Anwendungen höher, da für das Training alle Daten vorhanden sind. Der erhaltene Richtwert zeigt dann, wie nahe das reale Szenario diesem Wert kommen kann. Zu beachten ist auch, dass in realen Szenarien lokale Daten nicht ausgetauscht werden, da in den meisten Fällen ein Datenschutzinteresse besteht. Zum anderen benötigt der zentrale Ansatz (wenn er denn überhaupt durchgeführt werden kann und darf) oft wesentlich mehr Energie, um das KI-Modelltraining auf dem einen rieigen Trainingsdatensatz aller lokalen Klienten durchzuführen.

Mehrmalige differenzielle Übertragung mittels NNC

Wie in Abbildung 20 ersichtlich, verändert sich die Genauigkeit des zentralen Modells nach ca. 25 Epochen nicht mehr sichtlich, das heißt, der Prozess ist gegen eine konstante erreichbare Genauigkeit konvergiert. Die Konvergenz in der Genauigkeit korrespondiert üblicherweise auch mit der Konvergenz der Modellgewichte: Damit ändern sich nach 25 Kommunikationsrunden auch die Gewichte der Modelle nicht mehr nennenswert.
Der Fakt, dass das Modell, das der Server in jeder Kommunikationsrunden auch die Gewichte der Modelle nicht mehr

nennenswert. Der Fakt, dass das Modell, das der Server in jeder Kommunikationsrunde an die Klienten verteilt, sich während des Trainings nur leicht verändert, lässt sich in der Kompressionsphase ausnutzen. Dazu wurde untersucht, inwieweit sich differenzielle Modelle besser komprimieren lassen als Originalmodelle. Das heißt, in jeder Trainingsrunde wird statt der Kompression des Originalmodells nun die Differenz des Modells zu seinem zeitlichen Vorgänger komprimiert. Im Speziellen wird die Differenzbildung für jedes einzelne Gewicht im Modell durchgeführt. Verändert sich nun ein Modell in einer Trainingsrunde kaum noch im Vergleich zum Modell aus der vorhergehenden Trainingsrunde (im Beispiel in Abbildung 20 ab Runde 25), entstehen sehr kleine Differenzwerte und insbesondere viele Werte = 0, die sich sehr stark komprimieren lassen.

Durch diese differenzielle Übertragung können unter geeigneten Bedingungen sehr hohe Kompressionsraten und somit weitere Energieeinsparungen erreicht werden. Der Nachteil dabei ist jedoch, dass für eine Rekonstruktion eines Modells auch das Modell aus der vorangegangenen Trainingsrunde gespeichert werden muss, da ja nun Differenzwerte erhalten werden, die erst auf ein Referenzmodell aufaddiert werden müssen. Benutzt man differenzielle Übertragung für den Upload und den Download, muss jeder Klient in jeder Runde fehlerfreie Daten erhalten, da Übertragungsfehler sich sonst im Laufe der Kommunikationsrunden aufsummieren.

Für die Experimente wurde wieder beispielhaft das ResNet18-Modell mit 46,8 MB verwendet. Für die Betrachtung der Energiebilanzen gelten die gleichen Annahmen, wie beim Vergleich der originalen und komprimierten Übertragung weiter oben beschrieben. Für die differenziell-komprimierte Übertragung wird zusätzlich der geringfügige Energieaufwand für die Differenzbildung mit bei der Kompression und der für das Aufaddieren mit bei der Dekompression berücksichtigt. Die Ergebnisse sind in Abbildung 21 dargestellt.

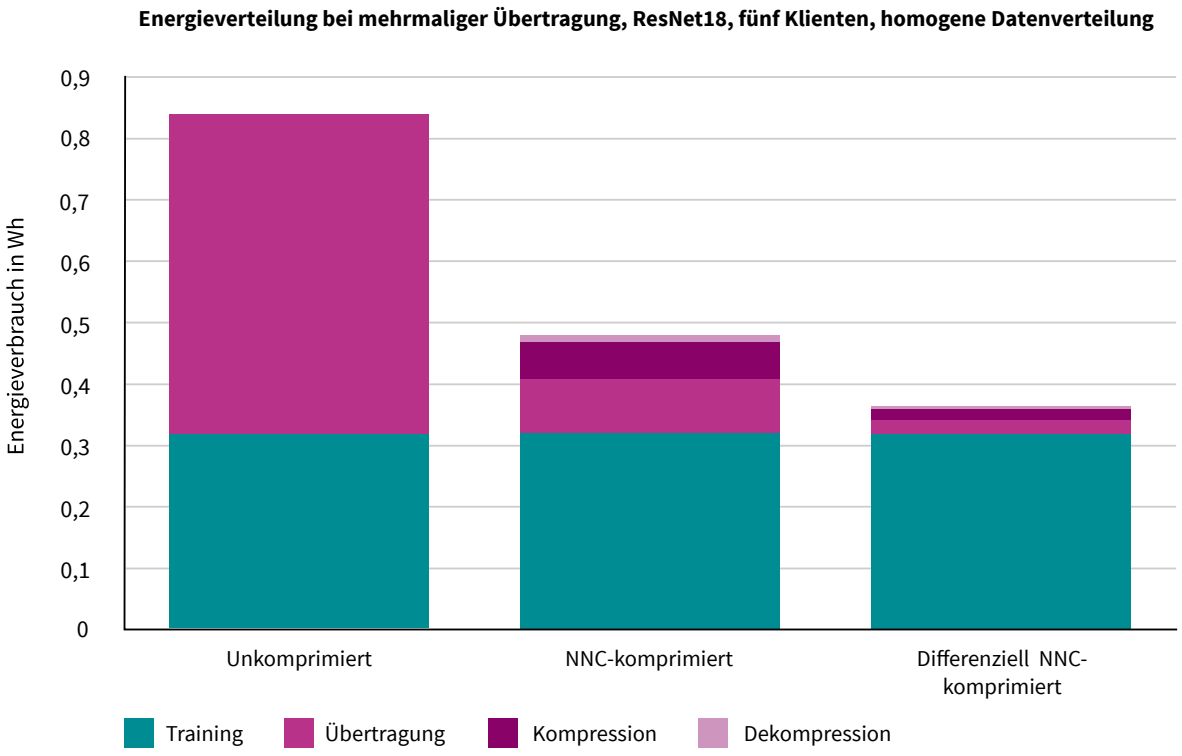


Abbildung 21: Gesamtenergiebilanz beim föderierten Lernen bei unkomprimierter Übertragung (links), komprimierter Übertragung (Mitte) und differenziell-komprimierter Übertragung (rechts)

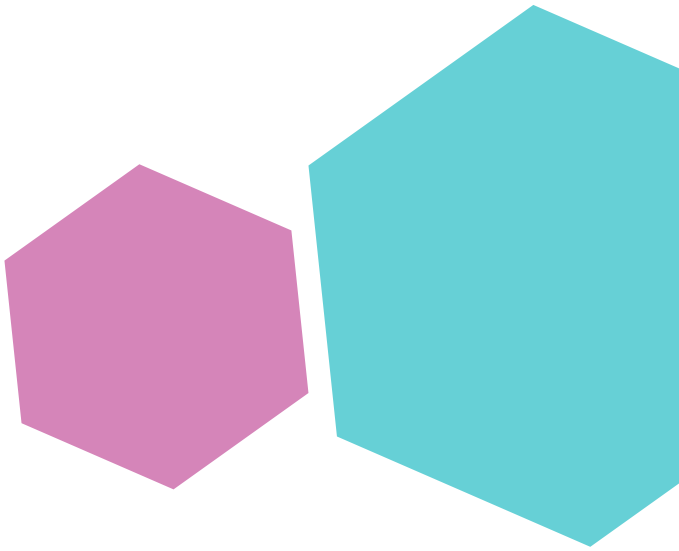
Insbesondere ist zu sehen, dass sich nicht nur der Energieaufwand für die eigentliche Übertragung stark verringert hat, sondern auch die Energieaufwände für die Codierung und Decodierung. Dadurch können in diesem Szenario durch differenzielle Kompression der Gewichte des KI-Modells bis zu 99 Prozent der Energie für die Übertragung im Vergleich zur originalen Übertragung eingespart werden. Der Gesamtenergiebedarf eines FL-Prozesses mit differenzieller Übertragung wird dann vom Energieverbrauch für das Training der Klienten dominiert, wie am konstanten Energieanteil für das Training bei allen Verfahren in Abbildung 21 zu sehen ist.

3.4.2 Testergebnisse für das Modell ResNet18 auf dem Datensatz CIFAR10

Im Folgenden sind die Detailergebnisse für die Kombination ResNet18 mit CIFAR10 in den folgenden vier Tabellen dargestellt. Der Datensatz CIFAR10 ist nach dem Canadian Institute for Advanced Research (CIFAR) benannt. Die ersten beiden Tabellen zeigen die Energieeinsparungen in Wattstunden (Wh), die jeweils bei komprimierter Übertragung im Vergleich zu unkomprimierter Übertragung gemessen wurden (bzw. für größere Teilnehmerzahlen interpoliert wurden, da die technische Plattform in der Menge der zu simulierenden Daten begrenzt ist). Die folgenden beiden Tabellen zeigen dann die entsprechenden CO_2-Einsparungen. Dazu wurde beispielhaft eine Kommunikation zwischen Deutschland und Spanien gewählt. Der jeweilige Energiemix beider Teilnehmerländer gibt dann den CO_2-Ausstoß pro kWh an, hier zum Beispiel 360 g CO_2/kWh, sodass die entsprechenden Ergebnisse für die CO_2-Einsparungen berechnet werden können. Weiterhin schließen alle Tabellen die drei eingangs definierten Szenarien ein, das heißt:

1) Einmalige Übertragung (1. Ergebniszelle)
2) Mehrmalige Übertragung an einen Teilnehmer (1. Ergebnisspalte)
3) Mehrmalige Übertragung an viele Teilnehmer (2. bis 7. Ergebnisspalte)

Tabelle 2 zeigt zunächst die Energieeinsparungen bei komprimierter Übertragung. Dabei ist zu erkennen, dass die Energieeinsparung linear mit der Anzahl der Teilnehmer skaliert, wie auch bereits bei der relativen Unabhängigkeit der Kompression von der Datenhomogenität gezeigt. Das bedeutet, dass selbst bei inhomogener Datenverteilung die komprimierte Bitstrom-Größe für jeden Teilnehmer nahezu gleich ist. Damit ergibt sich die Energieeinsparung für 10 Teilnehmer auch aus dem 10-Fachen der Energieeinsparung für einen Teilnehmer. Bezüglich der Anzahl der Kommunikationsrunden zeigt sich ein ähnliches Bild. Auch hier ergeben sich in etwa lineare Einsparungen, das heißt, bei 10 Kommunikationsrunden wird etwa 10-mal mehr Energie eingespart als bei einmaliger Übertragung.



Anzahl Kommunikationsrunden \ Anzahl Teilnehmer	1	10	100	1.000	10.000	100.000	1.000.000
1	0,07	0,74	7,35	73,51	735,06	7.350,55	73.505,52
5	0,38	3,79	37,87	378,66	3.786,65	37.866,46	378.664,59
10	0,76	7,61	76,07	760,68	7.606,85	76.068,48	760.684,76
20	1,52	15,20	152,00	1.519,99	15.199,94	151.999,36	1.519.993,60
30	2,28	22,79	227,92	2.279,25	22.792,49	227.924,92	2.279.249,21
40	3,03	30,31	303,07	3.030,70	30.306,97	303.069,69	3.030.696,88
50	3,75	37,51	375,10	3.750,95	37.509,51	375.095,14	3.7509.51,41

Tabelle 2: Energieeinsparung in Wh bei **komprimierter Übertragung** im Vergleich zu unkomprimierter Übertragung mit unterschiedlicher Anzahl an Kommunikationsrunden und Teilnehmern

Tabelle 3 zeigt die Energieeinsparungen bei differenziell-komprimierter Übertragung. Im Vergleich zu Tabelle 2 sind zunächst Einsparungen zu erkennen. Sie fallen jedoch eher moderat aus, da für alle Übertragungsvarianten immer der konstante Trainingsanteil mit berücksichtigt wird (siehe Kapitel 3.4.1).

So konnten zum Beispiel bei 10 Teilnehmern und 50 Kommunikationsrunden 13 Prozent Energie eingespart werden (37,5 Wh bei komprimierter und 42,6 Wh bei differenziell-komprimierter Übertragung).

Anzahl Kommunikationsrunden \ Anzahl Teilnehmer	1	10	100	1.000	10.000	100.000	1.000.000
1	0,09	0,85	8,52	85,21	852,13	8.521,26	85.212,61
5	0,43	4,26	42,64	426,38	4.263,82	42.638,24	426.382,42
10	0,85	8,52	85,21	852,06	8.520,62	85.206,23	852.062,31
20	1,70	17,02	170,18	1.701,77	17.017,68	170.176,80	1.701.767,99
30	2,55	25,50	254,99	2.549,92	25.499,18	254.991,77	2.549.917,70
40	3,39	33,95	339,46	3.394,64	33.946,38	339.463,83	3.394.638,26
50	4,26	42,61	426,11	4.261,09	42.610,88	426.108,79	4.261.087,86

Tabelle 3: Energieeinsparung in Wh bei **differenziell-komprimierter Übertragung** im Vergleich zu unkomprimierter Übertragung mit unterschiedlicher Anzahl an Kommunikationsrunden und Teilnehmern

Tabelle 4 zeigt die CO_2-Einsparung für Tabelle 2 an. Dabei wurde beispielhaft eine Kommunikation zwischen Deutschland und Spanien angenommen. Durch den aktuell verwendeten Energiemix beider Länder zur Stromerzeugung ergeben sich hier Werte von ca. 360 g CO_2/kWh und damit die erzielten CO_2-Einsparungen in Tabelle 4.

Anzahl Kommunikationsrunden \ Anzahl Teilnehmer	1	10	100	1.000	10.000	100.000	1.000.000
1	0,03	0,26	2,64	26,38	263,83	2.638,33	26.383,34
5	0,14	1,36	13,59	135,91	1.359,14	13.591,41	135.914,08
10	0,27	2,73	27,30	273,03	2.730,33	27.303,26	273.032,58
20	0,55	5,46	54,56	545,57	5.455,71	54.557,13	545.571,30
30	0,82	8,18	81,81	818,09	8.180,91	81.809,09	818.090,92
40	1,09	10,88	108,78	1.087,81	10.878,08	108.780,80	1.087.808,03
50	1,35	13,46	134,63	1.346,33	13.463,29	134.632,90	1.346.328,99

Tabelle 4: CO_2-Einsparung in g CO_2 bei **komprimierter Übertragung** im Vergleich zu unkomprimierter Übertragung mit unterschiedlicher Anzahl an Kommunikationsrunden und Teilnehmern. Annahme: Datenübertragung zwischen Deutschland und Spanien mit ca. 360 g CO_2/kWh

Tabelle 5 zeigt analog die CO_2-Einsparung für Tabelle 3 an. Entsprechend ergeben sich durch die differenziell-komprimierte Übertragung höhere Einsparungen.

Anzahl Kommunikationsrunden \ Anzahl Teilnehmer	1	10	100	1.000	10.000	100.000	1.000.000
1	0,03	0,31	3,06	30,59	305,85	3.058,54	30.585,36
5	0,15	1,53	15,30	153,04	1.530,41	15.304,14	153.041,44
10	0,31	3,06	30,58	305,83	3.058,31	30.583,07	305.830,73
20	0,61	6,11	61,08	610,82	6.108,16	6.1081,56	610.815,59
30	0,92	9,15	91,52	915,24	9.152,42	91.524,20	915.241,96
40	1,22	12,18	121,84	1.218,44	12.184,38	121.843,75	1.218.437,51
50	1,53	15,29	152,94	1.529,43	15.294,32	152.943,23	1.529.432,27

Tabelle 5: CO_2-Einsparung in g CO_2 bei **differenziell-komprimierter Übertragung** im Vergleich zu unkomprimierter Übertragung mit unterschiedlicher Anzahl an Kommunikationsrunden und Teilnehmern. Annahme: Datenübertragung zwischen Deutschland und Spanien mit ca. 360 g CO_2/kWh

3.4.3 Testergebnisse für das Modell ResNet152 auf dem Datensatz CIFAR10

Im Folgenden sind die Detailergebnisse für die Kombination ResNet152 mit CIFAR10 in den folgenden vier Tabellen dargestellt. Die ersten beiden Tabellen zeigen wieder die Energieeinsparungen in Wh, die jeweils bei komprimierter Übertragung im Vergleich zu unkomprimierter Übertragung erzielt werden.

Die folgenden beiden Tabellen zeigen dann wieder die entsprechenden CO_2-Einsparungen bei der Beispielkommunikation zwischen Deutschland und Spanien. Grundsätzlich ergeben sich analoge Zusammenhänge wie im vorangegangenen Abschnitt. Hervorzuheben ist hier, dass sich durch die wesentlich komplexere KI-Modellstruktur des ResNet152 im Vergleich zum ResNet18 auch wesentlich höhere Energie- und CO_2-Einsparungen für alle Teilnehmer bzw. Kommunikationsrunden-Kombinationen ergeben.

Anzahl Kommunikationsrunden \ Anzahl Teilnehmer	1	10	100	1.000	10.000	100.000	1.000.000
1	0,40	4,02	40,20	401,95	4.019,54	40.195,41	401.954,13
5	1,67	16,71	167,12	1.671,20	16.712,00	167.120,01	1.671.200,14
10	3,01	30,08	300,84	3.008,45	30.084,45	300.844,54	3.008.445,39
20	5,92	59,19	591,90	5.919,04	59.190,43	591.904,25	5.919.042,52
30	9,08	90,81	908,13	9.081,26	90.812,56	908.125,62	9.081.256,20
40	12,42	124,17	1.241,72	12.417,17	124.171,70	1.241.716,97	12.417.169,75
50	15,48	154,80	1.547,99	15.479,89	154.798,91	1.547.989,09	15.479.890,91

Tabelle 6: Energieeinsparung in Wh bei **komprimierter Übertragung** im Vergleich zu unkomprimierter Übertragung mit unterschiedlicher Anzahl an Kommunikationsrunden und Teilnehmern

Anzahl Kommunikationsrunden \ Anzahl Teilnehmer	1	10	100	1.000	10.000	100.000	1.000.000
1	0,42	4,15	41,54	415,40	4.153,97	41.539,73	415.397,30
5	2,07	20,70	206,98	2.069,83	20.698,34	206.983,37	2.069.833,75
10	4,05	40,51	405,15	4.051,48	40.514,81	405.148,09	4.051.480,91
20	7,97	79,72	797,23	7.972,31	79.723,10	797.230,97	7.972.309,72
30	11,94	119,35	1.193,52	11.935,23	119.352,32	1.193.523,24	11.935.232,39
40	15,89	158,91	1.589,06	15.890,58	158.905,76	1.589.057,65	15.890.576,46
50	20,19	201,86	2.018,56	20.185,57	201.855,68	2.018.556,76	20.185.567,64

Tabelle 7: Energieeinsparung in Wh bei **differenziell-komprimierter Übertragung** im Vergleich zu unkomprimierter Übertragung mit unterschiedlicher Anzahl an Kommunikationsrunden und Teilnehmern

Anzahl Kommunikationsrunden \ Anzahl Teilnehmer	1	10	100	1.000	10.000	100.000	1.000.000
1	0,14	1,44	14,43	144,27	1.442,73	14.427,34	144.273,40
5	0,60	6,00	59,98	599,84	5.998,44	59.984,39	599.843,86
10	1,08	10,80	107,98	1.079,82	10.798,21	107.982,13	1.079.821,30
20	2,12	21,25	212,45	2.124,52	21.245,22	212.452,19	2.124.521,93
30	3,26	32,60	325,95	3.259,54	32.595,35	325.953,53	3.259.535,29
40	4,46	44,57	445,69	4.456,89	44.568,95	445.689,47	4.456.894,74
50	5,56	55,56	555,62	5.556,20	55.561,97	555.619,72	5.556.197,24

Tabelle 8: CO_2-Einsparung in g CO_2 bei **komprimierter Übertragung** im Vergleich zu unkomprimierter Übertragung mit unterschiedlicher Anzahl an Kommunikationsrunden und Teilnehmern. Annahme: Datenübertragung zwischen Deutschland und Spanien mit ca. 360 g CO_2/kWh

Anzahl Kommunikationsrunden \ Anzahl Teilnehmer	1	10	100	1.000	10.000	100.000	1.000.000
1	0,15	1,49	14,91	149,10	1.490,99	14.909,86	149.098,55
5	0,74	7,43	74,29	742,93	7.429,25	74.292,54	742.925,43
10	1,45	14,54	145,42	1.454,20	14.541,98	145.419,80	1.454.198,04
20	2,86	28,62	286,15	2.861,50	28.615,01	286.150,11	2.861.501,13
30	4,28	42,84	428,39	4.283,91	42.839,13	428.391,30	4.283.912,96
40	5,70	57,04	570,36	5.703,60	57.036,05	570.360,46	5.703.604,61
50	7,25	72,45	724,52	7.245,21	72.452,06	724.520,58	7.245.205,79

Tabelle 9: CO_2-Einsparung in g CO_2 bei differenziell-komprimierter Übertragung im Vergleich zu unkomprimierter Übertragung mit unterschiedlicher Anzahl an Kommunikationsrunden und Teilnehmern. Annahme: Datenübertragung zwischen Deutschland und Spanien mit ca. 360 g CO_2/kWh

3.5 Webbasierter Demonstrator

Parallel zu den beschriebenen Experimenten wurde ein Web-Demonstrator entwickelt, der die Erkenntnisse aus den durchgeführten Experimenten im föderierten Setting visualisiert. In dieser Oberfläche, die in die Website des Future Energy Lab eingebettet ist, erfolgen die Visualisierung der beschriebenen Experimente und eine Umrechnung der erzielten Ersparnisse durch Komprimierung von neuronalen Netzwerken in allgemein bekannte Metriken. Die Oberfläche lässt sich dabei in drei Teile, auch „Widgets“ genannt, unterteilen, die eine Zusammenstellung und Analyse eines Szenarios erlauben.

1.) Auswahl-Widget

Das Auswahl-Widget wird genutzt, um ein föderiertes Lernszenario interaktiv zusammenzustellen. Dabei können folgende Optionen angepasst werden:

Server-/Gerätestandort

Um einen globales föderiertes Lernsystem zu simulieren, können sowohl der Standort des Servers als auch der Standort der Geräte, auf denen die Modelle trainiert werden, die sogenannten Klienten, ausgewählt werden. Da jedes Land eine andere Kohlenstoffintensität pro kWh Strom hat, ist die schlussendliche CO_2-Ersparnis abhängig von diesen Parametern.

KI-Modell und Datensatz

Ein weiterer wichtiger Faktor für die energetische Betrachtung von verteiltem Lernen sind die Art eines Modells und der Datensatz, auf dem trainiert wird. In den Experimenten wurden Modelle mit den Architekturen ResNet18, ResNet152 und Faster R-CNN untersucht, die in der Oberfläche mit ihren Datensätzen (CIFAR10 für Bildklassifizierung und WIDERFace für Objekterkennung) auswählbar sind. Je größer ein Modell und je größer der Trainingsdatensatz ist, desto größer ist auch die absolute Einsparung an Energie durch komprimierte Übertragung.

Anzahl der simulierten Geräte, Anzahl der Kommunikationsrunden, Heterogenitätslevel

Weiterhin können die Anzahl an Geräten und damit der Klienten, die an dem föderierten Lernszenario teilnehmen, sowie die Laufzeit des dezentralen Trainings durch die Auswahl der Kommunikationsrunden dynamisch angepasst werden. Darüber hinaus lässt sich über einen Schalter zwischen einem Szenario mit homogenen Daten, sprich: alle Klienten haben sehr ähnliche Datensätze, und einem heterogenen Setting, bei dem jeder Klient meist nur Beispiele einer Klasse in seinem Trainingsdatensatz besitzt, umschalten.

Fortgeschrittene Filter: Upload-/Download-Geschwindigkeiten

Für die Berechnung der Upload- und Download-Dauer der Modelle während der Übertragung wurden die Weltdurchschnittswerte von Speedtest genutzt. Da diese Werte jedoch von Land zu Land sehr unterschiedlich sind, gibt es im fortgeschrittenen Menü die Möglichkeit, die Upload- und Download-Geschwindigkeiten händisch anzupassen. Im Allgemeinen kann man sagen, dass, je langsamer die Übertragung ist, desto mehr Ersparnis durch Kompression erzielt wird. Dies hängt damit zusammen, dass die Leerlaufleistung der Klienten konstant ist.

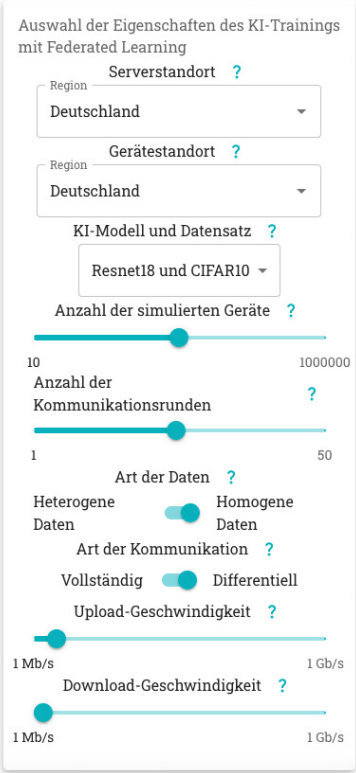


Abbildung 22: Auswahl-Widget

2.) Energieverbrauchsdiagramm

Das Energieverbrauchsdiagramm visualisiert den Gesamtenergieverbrauch über die Kommunikationsrunden für das ausgewählte Szenario für einen Federated-Learning-Prozess mit Standardkompression und NNC-Codec im Vergleich. Diese Ansicht soll dem Nutzer eine Vorstellung von den Energieersparnissen bei der Nutzung einer Standardkompression im Vergleich zur Nutzung des NNC-Codec über die Zeit hinweg geben.

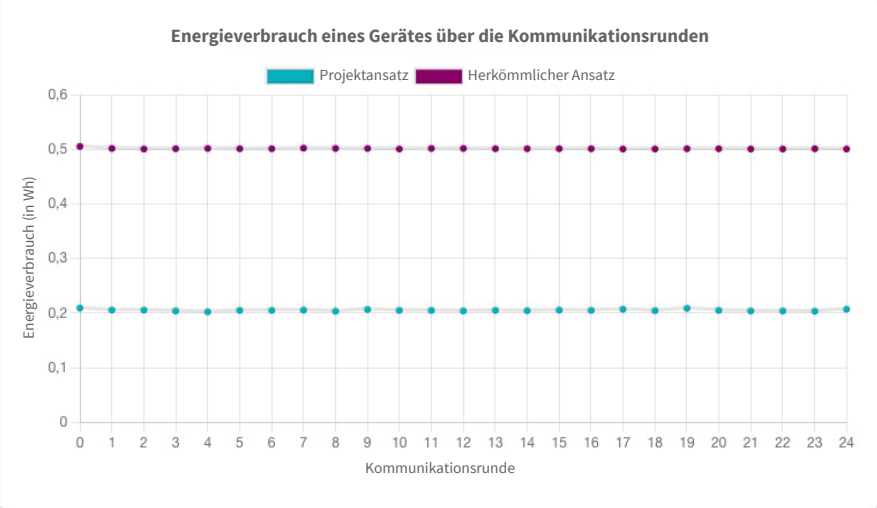


Abbildung 23: Energieverbrauchs-Widget

3.) Äquivalenz-Widget

In diesem Widget wird die durch die Nutzung des NNC-Coders gesparte Energie im ausgewählten Szenario mithilfe der CO_2-Intensitäten der ausgewählten Server- und Klienten-Standorte in eingesparte CO_2-Mengen umgerechnet. Um diese Einsparungen besser greifbar zu machen, wird eine Fahrt mit einem Mittelklasseauto mit Verbrennungsmotor bei fossiler Brennstoff nutzung angenommen, bei der die gleiche Menge CO_2 ausgestoßen würde, wie die Nutzung des NNC-Codec eingespart hat. Da die Experimente durch die technische Plattform in der Menge der zu simulierenden Daten begrenzt sind, wurden höhere Klienten-Anzahlen extrapoliert.

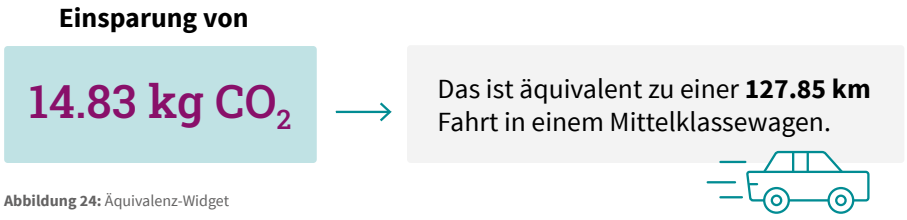


Abbildung 24: Äquivalenz-Widget

4. Weiterführende Energieeffizienzpotenziale

Die Resultate der beiden technischen Hauptrichtungen des EEKI-Projekts konnten im Bereich Ausführung und Übertragung von KI-Modellen Energieeinsparungen aufzeigen. Dies wurde im Bereich Ausführung durch eine optimierte Hardware-Architektur erreicht. Im Bereich Übertragung von KI-Modellen wurden Einsparungen durch eine Optimierung der Komprimierung erzielt. Letzteres war zunächst nicht zu erwarten, da bei komprimierter Übertragung zwar eine wesentlich geringere Datenrate erreicht wird, sodass hier die eigentliche Übertragung sehr energiesparend durchgeführt werden kann. Andererseits muss aber für die Kompression beim Sender und die Dekompression beim Empfänger zusätzlich Energie aufgewendet werden. Dass sich nun bereits bei einmaliger Übertragung eine bessere Energiebilanz ergibt, liegt an der verwendeten Kompressionsmethode NNC, die hohe Kompressionsraten von KI-Modellen bei gleichzeitig moderatem Energieverbrauch ermöglicht. Damit konnten die initial erwarteten Ergebnisse im Projekt übertroffen werden. Auf dieser Basis können weiterführende energieeffiziente KI-Themen entwickelt werden. Dazu werden im Folgenden vier wichtige Entwicklungen aufgezeigt, die bereits mit Expertinnen und Experten sowie Marktakteuren aus Energie- und Digitalwirtschaft im Rahmen eines Projekt-Workshops diskutiert wurden und die als relevant und wesentlich für die weitere Entwicklung auf dem Gebiet der energieeffizienten KI bewertet werden.

4.1 Weiterführende modulare Hardware-Optimierung für alle KI-Prozesse

Das Ziel der Hardware-Optimierung im Projekt EEKI war bisher die Ausführung von KI-Modellen, bei denen gezeigt wurde, das bereits trainierte neuronale Netze auf netzwerkgekoppelten FPGA-Beschleunigern weniger Energie benötigen. Eine erste weiterführende Optimierung ergibt sich durch eine verbesserte Technologie im Bereich FPGAs. Hier wurden kürzlich neueste FPGAs entwickelt, die eine höhere Energieeffizienz und Performance bieten. Dabei wirkt sich insbesondere der Wechsel der Speichertechnologie auf gestapelte HBM2e-Speicher (High Bandwidth Memory) aus, da hier eine wesentlich höhere Speicherbandbreite (ca. 1 TB/s gegenüber 22 GB/s bei DDR3-SDRAM) zur Verfügung steht. Durch diese enge Speicherkopplung innerhalb des FPGA-Package lassen sich bereits für die entwickelten Verfahren bei der KI-Modellausführung weitere Energieeinsparungen erzielen.

Als Weiterentwicklung der bisher erreichten Energieeinsparungen durch optimierte Hardware für die Ausführung von KI-Modellen ergibt sich die Ausdehnung der Verfahren auf das Training von KI-Modellen. Hier wurde insbesondere auch im Abschnitt zu den Energieeinsparungen bei der Übertragung von KI-Modellen in der Anwendungsumgebung des föderierten Lernens gezeigt, dass ein größerer Anteil der Gesamtenergiebilanz auf das Training entfällt. Entsprechend ergibt sich ein Forschungs- und Entwicklungsbedarf im Bereich des energieoptimierten Trainings. Dazu zählt die Erweiterung der entwickelten Hardware, also der NAAs mit FPGA-Technologie, auf das Training als Gesamtprozess. Zusätzlich können sich weitere Energiesparpotenziale durch eine Parallelisierung des KI-Trainings ergeben: Eine Möglichkeit wäre hier eine optimierte Lösung mit mehreren kleineren Trainingseinheiten, zum Beispiel als netzwerkgekoppelte FPGA-Beschleuniger, die zusammengeschaltet werden und dann in der Summe weniger Strom verbrauchen als ein einziges größeres Modul.

Des Weiteren kann durch eine Skalierung der Hardware eine direkte Anpassung an unterschiedliche Netzwerkarchitekturen erfolgen. Je nach Größe eines KI-Modells werden dann unterschiedlich viele Module zusammengeschaltet, das heißt, kleinere Netze, wie MobileNet, werden auf sehr wenigen Modulen trainiert, während größere neuronale Netze, wie ResNet50 oder VGG16, auf wesentlich mehr Modulen trainiert werden. Bei dieser modularen Architektur benötigen dann jeweils nur die aktiven Module Energie, sodass je nach trainierter KI-Architektur immer nur die minimal notwendige Energie verbraucht wird. Mit der Ankunft der neuartigen generativen KI-Modelle im Bereich Textvorhersage und -generierung wird der Aspekt der modularen Hardware-Architektur noch relevanter, da sich die Komplexität dieser neuartigen Modelle exponentiell erhöht hat, das heißt, die Anzahl trainierbarer Parameter pro Modell liegt hier um den Faktor 1.000 höher als bei bisherigen Modellen. Für eine Umsetzung dieses Vorhabens im großen Stil müssten die Module in Serie produziert werden und dafür auch genügend Bauteile, wie die FPGAs, zur Verfügung stehen. Das bedeutet, hier sind Industriepartner anzusprechen und gegebenenfalls auch zu subventionieren, um die neuen Module in ausreichender Anzahl für einen zukünftigen flächendeckenden Einsatz zur Verfügung zu stellen. Für die Ausstattung in den Rechenzentren wäre als Handlungsempfehlung der Austausch der Hardware nur im turnusmäßigen Zyklus der Hardware-Erneuerung in den Rechenzentren durchzuführen, während bei Erweiterungen und Neuanschaffung, beispielsweise zur Vergrößerung der Rechenkapazität, die neuen netzwerkgekoppelten FPGA-Beschleuniger sofort eingesetzt werden könnten.

4.2 KI-Modell-Recycling

Neben den sehr guten Ergebnissen bei der Übertragung komprimierter KI-Modelle benötigt das Training der KI-Modelle bisher einen konstanten Energieanteil unabhängig von der Art der Übertragung, wie in den Arbeiten gezeigt. Um die Gesamtenergiebilanz weiterhin zu verbessern, müssen weiterführende Arbeiten auch hier im Bereich KI-Training ansetzen. Dazu ist eine Ausweitung von der einzelnen KI-Anwendung auf die aktuelle KI-Landschaft und deren Anwendungsgebiete notwendig. Wie im vorangegangenen Unterkapitel beschrieben, ergeben

sich zunächst Möglichkeiten zum Beispiel durch verbesserte und modulare Hardware, das individuelle Training von KI-Modellen energieeffizienter zu gestalten.

Bei diesen Ansätzen wurde bisher standardmäßig jedes KI-Training individuell und unabhängig von anderen (Trainings-)Prozessen betrachtet. Das heißt, in der heutigen KI-Landschaft findet für jede einzelne Anwendung und für jedes KI-Modell ein vollständiges separates Training statt, was eben auch einen großen Energieaufwand bedeutet. Schaut man sich einzelne Anwendungsgebiete genauer an, stellt man fest, dass oft ähnliche, wenn nicht gleiche Modelle auf ähnlichen Daten trainiert werden. Ein weiterführender Forschungsansatz für größere Energieeinsparungen ergibt sich daher durch differenzielles Training ähnlicher Anwendungen auf einem gemeinsamen vortrainierten Netz für definierte Standardanwendungen. Diese Standardanwendungen sind oft bereits als Referenzanwendungen vorhanden und werden in wissenschaftlichen Arbeiten als Vergleichsmethoden angegeben. Bezüglich der Energiebilanz würde nun ein Teil des anfänglichen Trainings nur ein einziges Mal erfolgen, während der andere Teil dann von jeder Anwendung für sich durchgeführt wird. Ziel wäre es dann, zu untersuchen, wie groß der Anteil am vortrainierten Modell sein muss, um weiterhin korrekte Ergebnisse in den einzelnen Anwendungen zu bekommen.

Ein wesentlicher Punkt für dieses sogenannte Modell-Recycling wären zunächst die Katalogisierung und Sammlung vortrainierter neuronaler Netze für bestimmte und wesentliche Anwendungsaufgaben. Ein Beispiel ist hier ein ResNet50-Modell, das auf ImageNet trainiert wurde, also ein oft verwendetes Modell, das auf einem weit verbreiteten Datensatz von Bildern zur Bildklassifizierung (also ImageNet) trainiert wurde. Weitere Anwendungsbereiche neben der Bilderkennung sind die Objektsegmentierung, die Spracherkennung, die Textanalyse und -erkennung und Weitere. Hier kann auch berücksichtigt werden, welche weiteren Anwendungen in den kommenden Jahren relevant werden. Das heißt, das Ziel dieser weiterführenden Untersuchungen wäre die Erstellung einer Datenbank vortrainierter Netze für wesentliche Anwendungsbereiche. Im Anschluss wird nun das Modell-Recycling angewendet, bei dem für eine neue Anwendung ein bereits vortrainiertes Netz einer ähnlichen Anwendung verwendet und lediglich an den (leicht) veränderten Datensatz angepasst, also weitertrainiert, wird.

Hier wäre weiterhin eine Handlungsempfehlung an Industrie und Politik, eine verlässliche und vertrauenswürdige Basis mittels Standards zu schaffen, sodass KI-Anwender auch auf diese vortrainierten KI-Netzwerke und -Datenbanken zurückgreifen können. Eine Möglichkeit dazu wäre, nationale und internationale Standards zu schaffen, etwa über die Beteiligung des Deutschen Instituts für Normung (DIN) an den Aktivitäten der Internationalen Standardisierungsorganisation (ISO). Bei der ISO gibt es verschiedene Sektionen, die sich mit unterschiedlichen Standardisierungsthemen befassen. Eine davon ist die Sektion 42, die sich mit Fragen und Themen zur KI beschäftigt. Im Rahmen des Projekts wurde ja bereits der KI-Standard NNC für die energieeffiziente Übertragung verwendet, der auch von der ISO standardisiert wurde. Entsprechend könnte eine größere Standardisierung im Bereich Grüne KI, hier im Speziellen mit Standards zu vortrainierten KI-Modellen und Datensätzen, erfolgen, sodass im Ergebnis ein verlässlicher internationaler ISO-Standard zur Verfügung steht, der vom DIN übernommen wird und mit dem eine Vielzahl von Marktakteuren arbeiten wird.

4.3 Transferlernen (als selektives KI-Modell-Recycling)

Im vorangegangenen Unterkapitel wurde das Modell-Recycling mit standardisierten Basisanwendungen beschrieben. Um hier eine verlässliche Grundlage zu schaffen, wurde dazu eine anerkannte Standardisierung im Rahmen des DIN bzw. der ISO empfohlen. Auf dessen Basis können dann KI-Anwender ihre Modelle weitertrainieren. Da sich auch die Architekten der KI-Modelle weiterentwickeln und verändern, sollte das Modell-Recycling entsprechend erweitert werden. Dies wird etwa durch das Transferlernen ermöglicht, das auch bei veränderter Architektur Energieeinsparungen möglich macht. Während das Weitertrainieren beim Modell-Recycling immer das Weitertrainieren des gesamten Modells bedeutet, stellt das Transferlernen eine Erweiterung dar, bei der nur relevante Teile des Modells weitertrainiert werden. Zusätzlich kann sich eine Änderung des Aufbaus bzw. der Architektur des Modells ergeben, um zum Beispiel ein Modell für eine ähnliche Aufgabe mit anderen Parametern zu ermöglichen. Ein klassisches Beispiel für Transferlernen ist wieder im Bereich Bildklassifizierung angesiedelt: So kann ein neuronales Netz auf die Erkennung von zehn verschiedenen Klassen (z. B. 9 x unterschiedliche Objekte + 1 x „kein Objekt“) trainiert worden sein. Durch Transferlernen soll nun dieses vortrainierte Netz auf den Anwendungsfall mit zwei Klassen („Objekt vorhanden“ vs. „Objekt nicht vorhanden“) umgelernt werden. Hier wird nun lediglich die letzte Schicht geändert und neu trainiert, da sie nun statt zehn nur zwei Neuronen benötigt. Hier kann sich eine verbesserte Energiebilanz im Vergleich zum Modell-Recycling ergeben, da das Weitertrainieren ausschließlich auf die letzte Schicht beschränkt bleibt.

Für das Transferlernen ergeben sich weiterführende Forschungsfragen: Welche Teile der KI-Modelle sind besonders relevant und müssen weitertrainiert werden? Welche Schichten müssen weitertrainiert werden? Wie stark kann die Modellarchitektur verändert werden? Gibt es andere relevante Komponenten im Modell, die für das Transferlernen besser geeignet sind (etwa wichtige Pfade entlang des Netzwerks statt einzelner Schichten)? Als

Handlungsempfehlung ergibt sich auch hier die möglichst weitreichende Standardisierung der Verfahren, um KI-Anwendern eine verlässliche Basis der Technologien zu geben und insbesondere das Vertrauen in die Methoden zu etablieren, damit ein weitreichender Einsatz erfolgen kann und somit Energieeinsparungen im großen Stil möglich werden.

4.4 KI-Modelloptimierung und Energieeffizienz-Klassifizierung

In den vorangegangenen Unterkapiteln wurden Methoden zum optimierten KI-Modelltraining dargestellt, bei denen zunächst ein gemeinsames vortrainiertes Basisnetz für viele ähnliche KI-Anwendungen verwendet wird. Anschließend wird dieses Modell insgesamt oder in Teilen von jeder Anwendung weitertrainiert. Ein weiterer Forschungsansatz ist die KI-Modelloptimierung, wobei hier die Anwendungsgeräte berücksichtigt werden. Dazu wird wieder von einer Katalogisierung wesentlicher KI-Anwendungen und zugehöriger neuronaler Netze ausgegangen, die als Teil internationaler Standards zu Grüner KI eine Vertrauensgrundlage schaffen, um hier breite Marktakzeptanz und damit einen umfassenden Einsatz der Verfahren sicherzustellen. Als nächste Ausbaustufe der Verfahren kann nun die spezielle Anwendung von KI in unterschiedlichen Endgeräten individuell betrachtet werden. Das heißt, dass zusätzlich die Architektur der Modelle an das jeweilige Endgerät angepasst wird. Im Speziellen soll das optimale Modell pro Gerätetyp (verschiedene Typen von Handys und Laptops, aber auch IoT-Geräte mit KI-Fähigkeiten) bezüglich der Energieeffizienz gefunden und spezifiziert werden. Im Ergebnis könnte dann das minimal notwendige neuronale Netz (bezüglich der Komplexität der Architektur sowie der Anzahl der Schichten und damit der trainierbaren Hyperparameter) identifiziert werden, das die jeweilige Analyseaufgabe (z. B. Bildklassifizierung) ohne Qualitätseinbuße ausführen kann. Auf der Basis der dann entwickelten Grüne-KI-Standards (z. B. im Rahmen der ISO) ergibt sich die Möglichkeit, eine Energieeffizienz-Klassifizierung von KI-Anwendungen, ähnlich der Energieverbrauchs-Klassifizierung elektrischer Geräte, zu spezifizieren. Im Detail wird die energieeffizienteste KI-Variante pro Gerätetyp als Referenz verwendet und stellt die beste Energieeffizienzklasse mit dem geringsten Energieverbrauch dar. Auch dies sollte wieder von vertrauenswürdigen Standards abgedeckt werden. Für weitere Anwendungen kann dann (auch geräteübergreifend) ermittelt werden, wie viel mehr Energie ein Gerät für den KI-Prozess im Vergleich zur Referenzanwendung benötigt, und auf dieser Basis ein entsprechendes Label vergeben werden. Als Handlungsempfehlung sollte auch hier die Akzeptanz der Industrie und der Verbraucher berücksichtigt werden. Das heißt, neben der technischen Entwicklung weiterer energieeffizienter KI-Verfahren sollte auch die Vergabe eines KI-Energiesparlabels hinsichtlich Nutzerakzeptanz untersucht werden. Dazu gehört es, zu überprüfen, ob ein neues Label geschaffen wird oder beispielsweise die bisherigen Energieeffizienzlabel (für elektrische Geräte) in ihrer jetzigen Form oder nur leicht verändert genutzt werden.

5. Handlungsempfehlungen

5.1 Handlungsempfehlungen auf Basis der neuen Hardware-Architektur für den Einsatz in Rechenzentren:

Das Zitat von Gordon Moore, einem der Mitgründer der Firma Intel, wonach sich die Effizienz von Computerchips im Durchschnitt alle zwei Jahre verdoppelt, ist weltberühmt. Dabei ist bekannt, dass diese Entwicklung aufgrund physikalischer Einschränkungen irgendwann zu einem Ende kommt. Die Ergebnisse des Projekts zeigen exemplarisch, dass in der Informations- und Kommunikationstechnik (IKT) auch noch andere Energieeffizienzpotenziale existieren – beispielsweise indem Chips, wie in diesem Projekt vorgenommen, neuartig verschaltet werden. Besonders hervorzuheben ist dabei ein in der Regel sehr positives Kosten-Nutzen-Verhältnis: Auf volkswirtschaftlicher Ebene werden die Entwicklungskosten für Effizienzsteigerungen normalerweise überkompensiert. Lässt man Rebound-Effekte unberücksichtigt, besteht auch aus einer Nachhaltigkeitsperspektive eine positive Kosten-Nutzen-Relation. Das Besondere an den in diesem Projekt verfolgten Ansätzen ist, dass keine direkten Rebound-Effekte zu erwarten sind, da die Energieeffizienz nicht durch eine gesteigerte Recheneffizienz der Hardware erreicht wurde, auf der KI-Anwendungen selbst laufen. Stattdessen wurde sie durch die Reduzierung der Serverleistung zur Steuerung dieser Hardware erzielt. Die Förderung der Energieeffizienzforschung in der IKT ist lohnenswert, da sie in der Regel zu enormen Skaleneffekten führt. **Es ist daher empfehlenswert, die öffentliche und private Forschung auf diesem Gebiet weiter voranzutreiben.**

Die Förderung der Forschung zu energieeffizienten KI-Anwendungen ist auch dann sinnvoll, wenn diese Anwendungen netto für mehr Energieeinsparung als -verbrauch sorgen. Denn der Verbrauch der Anwendung selbst bleibt immer positiv. Je mehr er gesenkt werden kann, desto besser. Selbst wenn in Zukunft die Energieversorgung zu 100 Prozent erneuerbar sein sollte, sind Energieeffizienzsteigerungen nach wie vor sinnvoll.

Damit die im Projekt erreichten Effizienzsteigerungen einen klimawirksamen Effekt erzielen, müssen sie in einem nächsten Schritt in der Praxis zur Anwendung kommen. **Wir empfehlen daher, die unter Laborbedingungen getesteten Methoden in der Praxis einzusetzen, um die Ergebnisse im Feld zu bestätigen.** Für die entwickelte Hardware-Architektur mittels FPGA bedeutet dies, diese Architektur in einem Rechenzentrum zu verbauen und nach Bestätigung der Ergebnisse Anreize zu setzen, diese Methode unter kommerziellen Bedingungen anzuwenden. Perspektivisch könnten Rechenzentren im großen Stil die entsprechende Hardware-Architektur implementieren und so eine Verringerung des Gesamtstromverbrauchs erreichen. Heute sind Rechenzentren größtenteils noch wenig spezialisiert: Das bedeutet, sie sind darauf ausgerichtet, alle möglichen Rechenaufgaben zu übernehmen anstatt ausgewählte, für deren Umsetzung ihre Hardware besonders geeignet ist. Aus Sicht des Betreibers eines Rechenzentrums mag dies heute noch sinnvoll sein. In Zukunft wird der Rechenaufwand voraussichtlich aber so stark steigen, dass die Spezialisierung auf bestimmte Rechendienstleistungen einen Wettbewerbsvorteil darstellen kann. **Es ist daher empfehlenswert, frühzeitig die notwendigen Rahmenbedingungen zu schaffen, um Anreize für spezialisierte Rechenzentren zu setzen**. Hier sind Gesetzgeber zur Anreizschaffung und die Betreiber von Rechenzentren zur Adaption gleichermaßen angesprochen. **Der Anreiz zum Einbau energiesparender Hardware sollte durch weitere, schärfere Effizienzvorgaben bzw. -anreize bei neuen Rechenzentren oder für neu eingebaute Hardware-Komponenten gefördert werden.**

5.2 Handlungsempfehlungen auf Basis der Kompression der Übertragung beim föderierten Lernen

Das föderierte Lernen wird sich absehbar stark verbreiten. Ein Grund ist die Datenschutzfreundlichkeit dieses Ansatzes: So müssen Unternehmen und Organisationen, die die Entwicklung einer eigenen KI-Anwendung beauftragen, dafür keine oder nur wenige Daten teilen. Stattdessen kann die KI-Anwendung auf eigenen Servern und nicht auf den Servern des beauftragten Dienstleisters trainiert werden. **Es ist daher empfehlenswert, den im Projekt entwickelten Ansatz für eine energieeffizientere Datenübertragung zügig in der breiteren Anwendung zu testen, damit er sich idealerweise parallel mit dem föderierten Lernen verbreitet.** Dies sollte im Sinne der Auftraggeber und Auftragnehmer der Entwicklung von KI-Anwendungen sein, da hierdurch Strom und damit Kosten gespart werden. Zusätzlich empfehlen wir, zu untersuchen, inwiefern die Einführung eines Industriestandards an dieser Stelle sinnvoll ist, der eine Nutzung energiesparender Ansätze des föderierten Lernens etabliert. **Perspektivisch können die Ergebnisse auch Ausgangspunkt für die Entwicklung eines Energieeffizienzstandards für KI-Modelle sein – hardware- wie softwareseitig.** So ließen sich für verschiedene KI-Modelle Benchmarks für den Stromverbrauch setzen und es ließe sich eine Klassifizierung erstellen – vergleichbar mit der EU-Effizienzklassifizierung für Elektrogeräte. Dieser und weitere Vorschläge wurden in Kapitel 4 aus einer technischen Perspektive näher betrachtet und diskutiert.

6. Fazit

Im EEKI-Projekt wurden zwei wesentliche, weil zunehmend genutzte KI-Anwendungsbereiche auf mögliche Stromeinsparungen hin untersucht. Der erste Anwendungsbereich war die Nutzung von FPGAs zur Ausführung fertiger KI-Anwendungen. FPGAs eignen sich dafür in der Regel besser als klassische GPUs oder CPUs, da sie aufgrund ihrer flexiblen Schaltkreise besser an spezifische KI-Anwendungen angepasst werden können. Das bedeutet, die KI-Anwendung kann schneller ausgeführt werden. Es ist daher davon auszugehen, dass FPGAs in Zukunft noch stärker zur Ausführung von KI-Anwendungen genutzt werden. Im diesem Projekt wurde zunächst der Energieverbrauch von FPGAs ermittelt und anschließend ein Weg gefunden, diesen Energieverbrauch noch weiter zu senken. Erreicht wurde dies mittels einer neuartigen Verschaltung der FPGAs, wodurch in Rechenzentren weniger Server benötigt werden und so Energie eingespart wird. Dabei konnten Energieeinsparungen von bis zu 31 Prozent nachgewiesen werden.

Für den zweiten Anwendungsfall wurde zunächst der Stromverbrauch bei der Übertragung neuronaler Netze untersucht, wie sie beim föderierten Lernen zum Einsatz kommt. Anschließend wurde der Stromverbrauch bei der Übertragung durch den Einsatz energieeffizienter Kompressionsverfahren verringert und am Beispiel des föderierten Lernens evaluiert. Durch die hohe Kompression neuronaler Netze ohne Qualitätsverlust konnte die Übertragung der komprimierten Daten wesentlich energieeffizienter erfolgen als ohne Datenkompression. Dies schließt auch die zusätzlichen Energieaufwände für Kompression und Dekompression mit ein, sodass sich bereits bei einmaliger Übertragung an ein Gerät, das am Training teilnimmt, relativ große Energieeinsparungen ergeben. Die Energieeinsparungen addieren sich mit zunehmenden Übertragungen. Sie sind absolut wesentlich höher für Anwendungen, bei denen ein häufiges Modell-Update stattfindet, sowie bei föderierten Lernverfahren mit stetigem Datenaustausch. Bei diesem Verfahren konnten Einsparungen von 65 Prozent ermittelt werden.

Beide Verfahren wurden entwickelt, da aufgrund der stark wachsenden Verbreitung des Einsatzes von KI davon auszugehen ist, dass sowohl FPGAs als auch das föderierte Lernen vermehrt eingesetzt werden. Dadurch bieten Energieeffizienzsteigerungen an diesen Ansatzpunkten große Mehrwerte. Das Projekt hat gezeigt, dass erhebliche Stromeinsparungen realisierbar sind, was zuvor nicht zu erwarten war. Damit die Ergebnisse ihre Wirkung entfalten, gilt es nun, die Energieeinsparungen unter Realbedingungen zu bestätigen. Das bedeutet, die entwickelte Hardware-Komponente mit FPGAs in einem Rechenzentrum zu installieren sowie die Kompressionsverfahren in einem großflächigen föderierten Lernprozess anzuwenden. Anschließend können beide Verfahren in der Breite zum Einsatz kommen und damit einen erheblichen Beitrag zum Klimaschutz leisten. Darüber hinaus wurden im Projekt auf Basis der erhaltenen Ergebnisse weiterführende Entwicklungsmöglichkeiten identifiziert, mit denen weitere Energieeinsparungen erreicht werden könnten. Die erzielten Ergebnisse sind damit potenziell Ausgangspunkt für eine Reihe an Energiesparmöglichkeiten. Insbesondere die Nutzung der FPGA-Hardware-Architektur für das Training von KI-Anwendungen, auch in Kombination mit föderiertem Lernen unter Einsatz des entwickelten Kompressionsverfahrens, ist hier hervorzuheben, da beide Verfahren kombiniert werden könnten.

Das EEKI-Projekt markiert das erste Pilotprojekt des Future Energy Lab, in dem Hard- und Software-Teile gebaut wurden, um KI energieeffizienter zu machen. Neben vielen Projekten, in denen bereits das Anwendungspotenzial von KI für die Energiewende und den Klimaschutz aufgezeigt und erforscht wurde, soll dieses Projekt ausdrücklich zeigen, dass dabei der Stromverbrauch und der ökologische Fußabdruck von KI selbst nicht außer Acht gelassen werden dürfen. Dem Projekt lag ein explorativer Ansatz zugrunde. Das Ergebnis hätte auch lauten können, dass keine oder nur wenig Energieeinsparungen möglich sind. Dass die Einsparungen so hoch ausfallen, war nicht zu erwarten und stellt damit einen potenziellen Meilenstein auf dem Weg zu nachhaltigerer KI dar.

Abbildungsverzeichnis

Abbildung 1:	Schematische Darstellung eines KNN mit vier Schichten	10
Abbildung 2:	Typische Private-Cloud-Architektur mit GPU und FPGA Compute Nodes für KI-Anwendungen	14
Abbildung 3:	Rechnersysteme und ihre KI-spezifische Leistungsfähigkeit sowie Energiebedarf/Flexibilität	15
Abbildung 4:	PCIe-basierte Anbindung an FPGA- bzw. GPGPU-basierte Rechenbeschleuniger	17
Abbildung 5:	Anbindung der netzwerkbasierten FPGA-Beschleuniger an einen Steuerserver (Compute Node)	18
Abbildung 6:	NAA-Hardware-Framework	19
Abbildung 7:	NAA-Gehäuse	20
Abbildung 8:	Ausschnitt der Testanwendung zur MobileNetV2-Klassifizierung	21
Abbildung 9:	Prototypischer Gesamtsystemaufbau mit acht NAAs	22
Abbildung 10:	Messkonzept für NAA-Beschleunigerkarten	23
Abbildung 11:	Dashboard zur Energiemessung eines NAA	23
Abbildung 12:	Referenzenergiemessung der MobileNetV2-Inferenz	24
Abbildung 13:	Auswahl-Widget	26
Abbildung 14:	Verarbeitungsschritte für die Energiebilanz bei originaler KI-Modell-Übertragung und komprimierter Übertragung	29
Abbildung 15:	Schematische Darstellung der NNC-Struktur mit Encoder und Decoder	30
Abbildung 16:	Schematische Übersicht des föderierten Lernens	33
Abbildung 17:	Energiebilanz bei einmaliger Datenübertragung: unkomprimierte Übertragung und komprimierte Übertragung	35
Abbildung 18:	Gesamtenergiebilanz beim föderierten Lernen mit fünf Klienten und 50 Trainingsrunden bei unkomprimierter Übertragung und komprimierter Übertragung	36
Abbildung 19:	Energieaufwand für komprimierte und originale Übertragung über alle 50 Kommunikationsrunden bei heterogener Datenverteilung und homogener Datenverteilung sowie bei komprimierter Übertragung und differenziell-komprimierter Übertragung	37
Abbildung 20:	Erreichte Genauigkeiten für originale und komprimierte Übertragung beim föderierten Training sowie als Referenz das konstante Modell für einen zentralen Ansatz (nicht verteiltes Lernen)	38
Abbildung 21:	Gesamtenergiebilanz beim föderierten Lernen bei unkomprimierter Übertragung, komprimierter Übertragung und differenziell-komprimierter Übertragung	39
Abbildung 22:	Auswahl-Widget	45
Abbildung 23:	Energieverbrauchs-Widget	46
Abbildung 24:	Äquivalenz-Widget	46

Tabellenverzeichnis

Tabelle 1: Codier-Ergebnisse ohne Qualitätsverlust für unterschiedliche neuronale Netze 30

Tabelle 2: Energieeinsparung in Wh bei **komprimierter Übertragung** im Vergleich zu unkomprimierter Übertragung mit unterschiedlicher Anzahl an Kommunikationsrunden und Teilnehmern 41

Tabelle 3: Energieeinsparung in Wh bei **differenziell-komprimierter Übertragung** im Vergleich zu unkomprimierter Übertragung mit unterschiedlicher Anzahl an Kommunikationsrunden und Teilnehmern 41

Tabelle 4: CO_2-Einsparung in g CO_2 bei **komprimierter Übertragung** im Vergleich zu unkomprimierter Übertragung mit unterschiedlicher Anzahl an Kommunikationsrunden und Teilnehmern. Annahme: Datenübertragung zwischen Deutschland und Spanien mit ca. 360 g CO_2/kWh 42

Tabelle 5: CO_2-Einsparung in g CO_2 bei **differenziell-komprimierter Übertragung** im Vergleich zu unkomprimierter Übertragung mit unterschiedlicher Anzahl an Kommunikationsrunden und Teilnehmern. Annahme: Datenübertragung zwischen Deutschland und Spanien mit ca. 360 g CO_2/kWh 42

Tabelle 6: Energieeinsparung in Wh bei **komprimierter Übertragung** im Vergleich zu unkomprimierter Übertragung mit unterschiedlicher Anzahl an Kommunikationsrunden und Teilnehmern 43

Tabelle 7: Energieeinsparung in Wh bei **differenziell-komprimierter Übertragun**g im Vergleich zu unkomprimierter Übertragung mit unterschiedlicher Anzahl an Kommunikationsrunden und Teilnehmern 43

Tabelle 8: CO_2-Einsparung in g CO_2 bei **komprimierter Übertragun**g im Vergleich zu unkomprimierter Übertragung mit unterschiedlicher Anzahl an Kommunikationsrunden und Teilnehmern. Annahme: Datenübertragung zwischen Deutschland und Spanien mit ca. 360 g CO_2/kWh 8 44

Tabelle 9: CO_2-Einsparung in g CO_2 bei **differenziell-komprimierter Übertragung** im Vergleich zu unkomprimierter Übertragung mit unterschiedlicher Anzahl an Kommunikationsrunden und Teilnehmern. Annahme: Datenübertragung zwischen Deutschland und Spanien mit ca. 360 g CO_2/kWh 44

Literaturverzeichnis

1 Amazon-a (abgerufen am 16.09.2023). Abgerufen von https://aws.amazon.com/de/ec2/instance-types/f1/

2 Amazon-b (abgerufen am 16.09.2023). Abgerufen von https://aws.amazon.com/de/ec2/instance-types/p2/

3 Bianco, S., Cadene, R., Celona, L., & Napoletano, P. (2018): Benchmark analysis of representative deep neural network architectures. IEEE Access, 6, 64270-64277

4 Bitkom (2023). Abgerufen am 15.12.2023 von https://www.bitkom.org/Presse/Presseinformation/Europaeischer-KI-Markt-verfuenffacht-sich-binnen-fuenf-Jahren

5 Borderstep (abgerufen am 15.09.2023). Abgerufen von https://www.borderstep.de/facts-and-figures/rechenzentren-2022/

6 Bundesnetzagentur (2023). Abgerufen von https://www.bundesnetzagentur.de/SharedDocs/Pressemitteilungen/DE/2023/20230104_smard.html

7 Brown (2020). Abgerufen am 16.09.2023 von https://en.wikipedia.org/wiki/GPT-3

8 Brown, T. e. (20.07.2020): Language Models are Few-Shot Learners. Abgerufen am 16.09.2023 von https://arxiv.org/pdf/2005.14165.pdf

9 Chen, Y.-H., Krishna, T., Emer, J., & Sze, V. (2016): 14.5 Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. Proc. of IEEE International Solid-State Circuits Conference (ISSCC), 262-263

10 Deutsche Energie-Agentur (dena) (2022): Auf dem Weg zu energieeffizienter künstlicher Intelligenz – Welche Energieeinsparpotenziale bieten KI-Anwendungen? Abgerufen von https://www.dena.de/newsroom/publikationsdetailansicht/pub/dena-analyse-auf-dem-weg-zu-energieeffizienter-kuenstlicher-intelligenz-welche-energieeinsparpotenziale-bieten-ki-anwendungen

11 Deutsche Energie-Agentur (dena) (2023): Datenanalysen und künstliche Intelligenz im Stromverteilnetz

12 Fallahlalehzari, F. (ohne Datum). Abgerufen am 15.12.2021 von https://www.aldec.com/en/company/blog/167--fpgas-vs-gpus-for-machine-learning-applications-which-one-is-better

13 Falsafi, B., Dally, B., Singh, D., Chiou, D., Yi, J. J., & Sendag, R. (2017): FPGAs versus GPUs in Data centers. IEEE Micro, 37(1), 60-72

14 Future Energy Lab (2023). Abgerufen von https://future-energy-lab.de/projects/ki-in-fernwaerme/

15 Google (abgerufen am 16.09.2023). Abgerufen von https://cloud.google.com/tpu/docs/tpus

16 Hintemann, R. (2016). Abgerufen am 15.12.2021 von https://www.borderstep.de/wp-content/uploads/2017/03/Borderstep_Rechenzentren_2016.pdf

17 Hintemann, R., & Clausen, J. (2016): Green Cloud? The current and future development of energy consumption by data centers, networks and end-user devices. Proc. of ICT for Sustainability

18 Hintemann, R., Hinterholzer, S. & Seibel, H. (2023). Bitkom-Studie Rechenzentren in Deutschland: Aktuelle Marktentwicklungen – Update 2023. Berlin: Borderstep Institut. Abgerufen am 15.09.2023 von https://www.bitkom.org/sites/main/files/2023-05/BitkomStudieRechenzentreninDeutschland2023.pdf.

19 Kairouz, P., McMahan, H. B., Avent, B., ..., & Zhao, S. (2021): Advances and Open Problems in Federated Learning. arxiv:1912.04977

20 Kirchhoffer, H., Haase, P., Samek, W., Müller, K., Rezazadegan-Tavakoli, H., Cricri, F., ..., Bailer, W. (2021): Overview of the Neural Network Compression and Representation (NNR) Standard. IEEE Trans. Circuits Syst. Video Technol

21 Kratochwill, L., Richard, P., Babilon, L., Rehmann, F., Mamel, S., & Fasbender, S. (2020): Künstliche Intelligenz – vom Hype zur energiewirtschaftlichen Realität. Deutsche Energie-Agentur GmbH (dena)

22 Lobe, A. (2019). Abgerufen am 07.01.2022 von https://www.spektrum.de/news/kuenstliche-intelligenz-verbraucht-fuer-den-lernprozess-unvorstellbar-viel-energie/1660246

23 Luccioni, A. S.-G. (2023): Counting carbon: A survey of factors influencing the emissions of machine learning. Abgerufen von https://arxiv.org/pdf/2302.08476.pdf

24 Microsoft (abgerufen am 16.09.2023). Abgerufen von https://www.microsoft.com/en-us/research/project/project-catapult/

25 MPEG (2021): Common Test Conditions for Incremental Compression of Neural Networks. MPEG document WG04N0123, ISO/IEC JTC 1/SC 29/WG

26 Nurvitadhi, E., Boutros, A., Budhkar, P., Jafari, A., Kwon, D., Sheffield, D., ..., Naik, M. (2019): Scalable Low-Latency Persistent Neural Machine Translation on CPU Server with Multiple FPGAs. Proc. of International Conference on Field-Programmable Technology (ICFPT), 307-310

27 Nurvitadhi, E., Sheffield, D., Sim, J., Mishra, A., Venkatesh, G., & Marr, D. (2016): Accelerating Binarized Neural Networks: Comparison of FPGA, CPU, GPU, and ASIC. Proc. of International Conference on Field-Programmable Technology (FPT), 77-84

28 OpenAI – Models (abgerufen am 11.09.2023). Abgerufen von https://platform.openai.com/docs/models/overview

29 Putnam, A., Caulfield, A. M., Chung, E. S., & Burger, D. (2015): A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services. IEEE Micro, 35(3), 10-22

30 Qiu, X., Parcollet, T., Fernandez-Marques, J., Gusmao, P. P., Beutel, D. J., Topal, T., ..., Lane, N. D. (2021): A First Look into the Carbon Footprint of Federated Learning. arxiv:2102.07627

31 Ruloff, S. (2019): Konzept und Realisierung zur Erfassung der elektrischen Leistung von hardwarebeschleunigten Rechnerplattformen am Beispiel von FPGA PCI-Express Karten. Masterarbeit, HTW Berlin

32 Sattler, F., Marban, A., Rischke, R., & Samek, W. (2021): CFD: Communication-Efficient Federated Distillation via Soft-Label Quantization and Delta Coding. IEEE Trans. Netw. Sci. Eng

33 Sattler, F., Wiedemann, S., Müller, K.-R., & Samek, W. (2019): Sparse binary compression: Towards distributed deep learning with minimal communication. Proc. of International Joint Conference on Neural Networks (IJCNN), 1-8

34 Schaller, W. W. (August 2023). Künstliche Intelligenz: Chance oder Gefahr? Wie verändert der Einsatz von KI unsere Gesellschaft? ifo Schnelldienst.

35 Statista (abgerufen am 16.09.2023). Abgerufen von https://de.statista.com/statistik/daten/studie/3506/umfrage/monatliches-datenvolumen-pro-mobilfunknutzer-in-deutschland/

36 Strubell, E., Ganesh, A., & McCallum, A. (2019): Energy and Policy Considerations for Deep Learning in NLP. arxiv:1906.02243

37 Sze, V., Chen, Y.-H., Yang, T.-J., & Emer, J. (2017): Efficient processing of deep neural networks: A tutorial and survey. arxiv:1703.09039

38 Verdecchia, R., Sallou, J., & Cruz, L. (2023): A systematic review of Green AI. WIREs Data Mining and Knowledge Discovery, 1-26. doi:https://doi.org/10.1002/widm.1507

39 Wang, T., Geng, T., Li, A., Jin, X., & Herbordt, M. (2020): FPDeep: Scalable Acceleration of CNN Training on Deeply-Pipe lined FPGA Clusters. IEEE Trans. Comput., 69(8), 1143-1158

40 Wang, Y., Zhao, T., Li, L., Hou, Z., & Gu, J. (2018): Roofline Model Based Performance-Aware Energy Management for Scientific Computing. Proc. of 9th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)

41 Wiedemann, S., Kirchhoffer, H., Matlage, S., Haase, P., Marban, A., Marinč, T., ..., Samek, W. (2020): DeepCABAC: A Universal Compression Algorithm for Deep Neural Networks. IEEE J. Sel. Top. Signal Process., 14(4), 700-714

42 Xia, L., Diao, L., Jiang, Z., Liang, H., Chen, K., Ding, L., ..., Lin, W. (2019): PAI-FCNN: FPGA Based Inference System for Complex CNN Models. Proc. of IEEE 30th International Conference on Application-specific Systems, Architectures and Processors (ASAP), 107-114

Abkürzungsverzeichnis

API	Application Programming Interface
ARM	Advanced RISC Machine
ASIC	Application-Specific Integrated Circuit
ASIP	Application-Specific Instruction-Set Processor
CABAC	Context-Based Adaptive Binary Arithmetic Coding
CIFAR	Canadian Institute for Advanced Research
CPU	Central Processing Unit
CSR/CSC	Compressed Sparse Row/Column
(D)CNN	(Deep) Convolutional Neural Network
DDR3	Double Data Rate 3
DIN	Deutsches Institut für Normung
dena	Deutsche Energie-Agentur
EEKI	Energieeffiziente künstliche Intelligenz (Projektname)
FL	Föderiertes Lernen / Federated Learning
FLOPs	Floating Point Operations
FPGA	Field Programmable Gate Array
GB	Gigabyte
Gbps	Gigabit pro Sekunde
GB/s	Gigabyte pro Sekunde
GBit/s	Gigabit pro Sekunde
g CO_2/kWh	Gramm CO_2 pro Kilowattstunde
GELU	Gaußian Error Linear Unit
(G)FLOPs	(Giga) Floating Point Operations
(GP)GPU	(General Purpose) Graphics Processing Unit
GPT	Generative Predictive Transformer
HBM	High Bandwidth Memory
HHI	Fraunhofer Heinrich-Hertz-Institut
IKT	Informations- und Kommunikationstechnik
IoT	Internet of Things
ISO	International Standardisation Organisation
KB	Kilobyte
KI	Künstliche Intelligenz
KNN	Künstliches neuronales Netz
kWh	Kilowattstunde
MB	Megabyte
MBit/s	Megabit pro Sekunde
mJ	Megajoule
ML	Maschinelles Lernen / Machine Learning
NAA	Network-Attached Accelerator
nm	Nanometer
NN	Neuronales Netz
NNC	Neural Network Coding
PCIe	Peripheral Component Interconnect Express
QAT	Quantization Aware Training
QP	Quantisierungsparameter
RDMA	Remote Direct Memory Access
ReLU	Rectified Linear Unit
RoCEv2	RDMA over Converged Ethernet Version 2
SIMD	Single Instruction Multiple Data
TB/s	Terabyte pro Sekunde
TPU	Tensor Processing Unit
TWh	Terawattstunde
V	Volt
VGG	Visual Graph Group
Wh	Wattstunde

www.dena.de future-energy-lab.de

dena
Deutsche Energie-Agentur