Integrating Deep Learning in Domain Sciences at Exascale

Supercomputer: Forschung und Innovtion

Frederic Voigt

Arbeitsbereich Wissenschaftliches Rechnen
Fachbereich Informatik
Fakultät für Mathematik, Informatik und Naturwissenschaften
Universität Hamburg

2021-21-01

Gliederung (Agenda)

1 Einleitung

2 Exascale

3 Maschine Learning

4 Parallelismus

5 HPC + AI

6 Schluss

7 Literatur

Paper

- Integrating Deep Learning in Domain Sciences at Exascale [Arc20]
- Rick Archibald, Edmond Chow, Eduardo D'Azevedo, Jack Dongarra, Markus Eisenbach, Rocco Febbo, Florent Lopez, Daniel Nichols, Stanimire Tomov, Kwai Wong and Junqi Yin
- Oak Ridge National Laboratory, Georgia Institute of Technology, University of Tennessee, Knoxville

Exascale

- Exascale
  - Ausdruck der aktuell möglichen Rechenleistungen
  - Exa: 10^{18}
  - Rechenleistung in FLOP/S (Floating Point Operations / Second)
- Supercomputer Fugaku
  - Platz 1 der Top 500 Liste [top21]
  - 442,010.0 TFLOP/S (Rmax) – (ungefähr ein halber EFLOP/S)

Herausforderungen des Exascale Computings

- Kosten
  - Hardwarekosten
  - Stromverbrauch
- Speicherbegrenzung (im DRAM und in der Ein-/Ausgabe)
- Latenzbegrenzung
- Effiziente Anwendungsparallelisierung: "Billion-Way Parallelism"[Tur21]

[SDM11] [Tur21]

Machine Learning

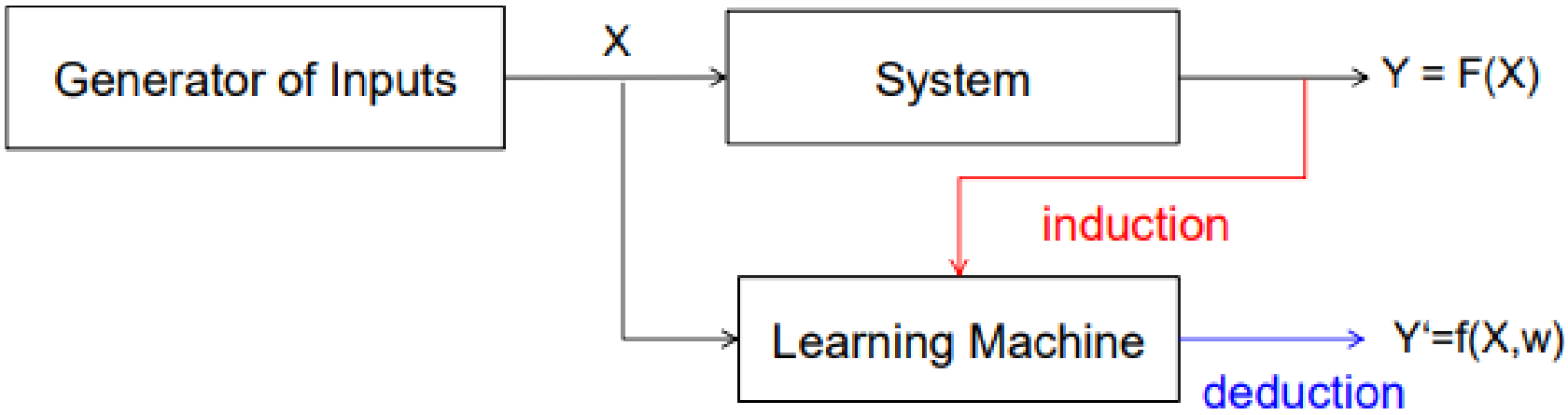

Abbildung: Abstrakte Darstellung eines Machine Learning Modells [aip18]

Deep Learning

- Teilbereich des Machine Learnings (ML), der auf neuronalen Netzen basiert
- Modelle werden "trainiert"
- Charakteristiken
  - (Viele) wiederholte Matrixmultiplikationen mit kleinen Gleitkommawerten
  - Große Eingabedatenmengen
  - Großer Bedarf an FLOP/S, Ein-/Ausgabe und Parallelisierung
  - Graphics Processing Unit (GPU) Einsatz
  - Wiederholte Durchführung von Stochastic Gradient Decent (SDG)

Daten-Parallelismus

- Zwei Wege Berechnungen zu parallelisieren
- Daten-Parallelismus
  - Gleicher Code auf verschiedenen Recheneinheiten
  - Berechnet einen Teil der Daten
  - w wird verteilt

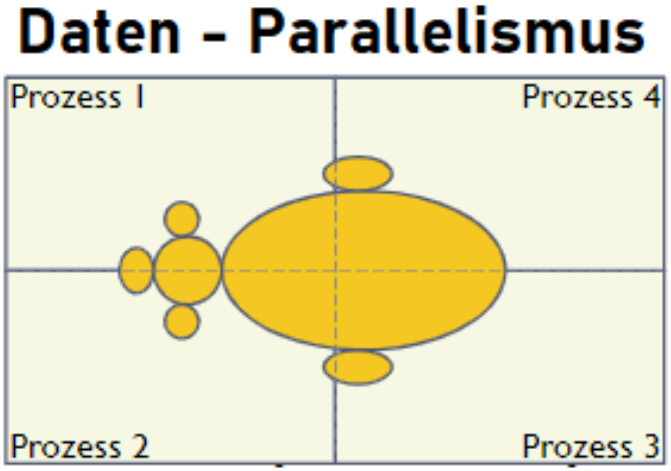

Abbildung: Daten-Parallelismus [Rec20, 266]

Model-Parallelismus

Model-Parallelismus

- Unterschiedlicher Code läuft auf den Recheneinheiten
- Alle Daten gehen durch jeden Abschnitt
- f wird verteilt

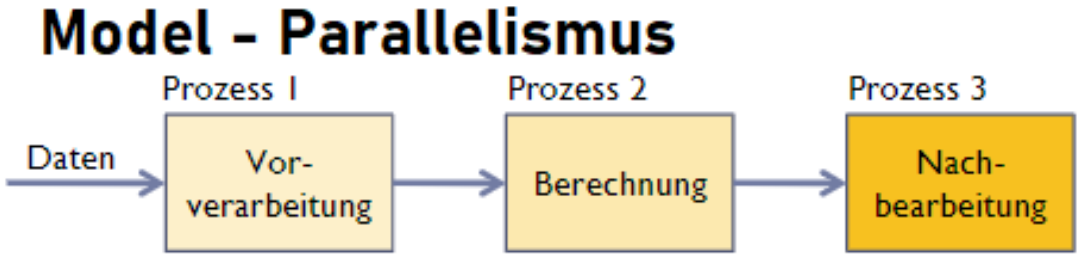

Abbildung: Model-Parallelismus [Rec20, 268]

Motivation

- "HPC + AI"
- High Performance Computing (HPC): geeignete Systeme (im Exascale Bereich)
- AI (Artificial Intelligence): Software
- „+“: innovative Algorithmen, die die Portierung auf Exascale Systeme leisten können

Herausforderungen der ML Portierung auf HPC Systeme

- Einsatz heterogener Hardware
- Zu langsame Ein-/Ausgabe
- Speicherbedarf
- ML Modelle reizen die Rechenleistung nicht aus
- Parallelisierung

Lösungsansätze

- Asynchrone Methoden
- Model- und Daten- Parallelismus
- Reduzierte Genauigkeit

Frameworks

- Matrix Algebra on GPU and Multicore Architectures, Deep Neural Networks (MagmaDNN)
  - ML Framework
- openDIEL
  - Workflow Software der University of Tennessee, Knoxville
  - Pipeline Aufbau
  - Hyperparameterverwaltung

Probleme des SGD

- Sperren der Parameter w während eines Updates nach SGD
- Alternative: Kopien von w auf jedem Knoten
  - Nach der Berechnung erfolgt synchroner Austausch unter allen Knoten
  - Alle Knoten warten auf langsamsten
- SGD wird zum Flaschenhals

Parameter Server Ansatz

- Arbeit auf verschiedenen Knoten
- Dedizierte Knoten (Parameter Server) speichern w und bekommen SDGs von Rechnerknoten
- Neues w wird an die Rechnerknoten gesendet

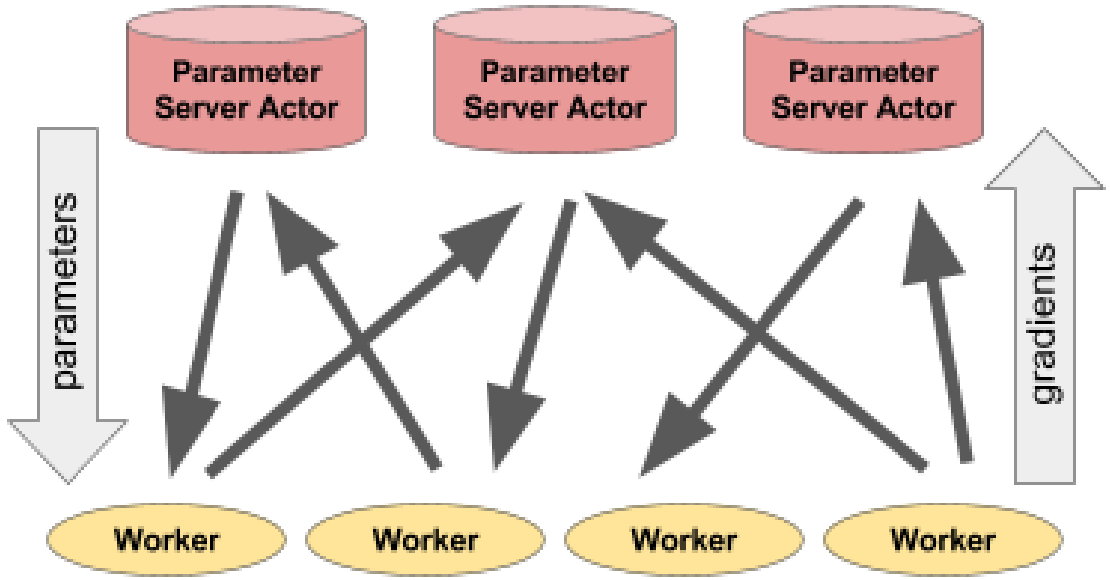

Abbildung: Parameter Server Ansatz Skizze [Tea21]

Konsequenzen des Parameter Server Ansatzes

- Daten-Parallelismus
- Vergrößerter Hauptspeicher
- Asynchrone Arbeitsweise und damit deutlich schneller
- Zwischenergebnisse werden inkonsistent und Berechnungen nicht-deterministisch
- Größere Datenmenge handhabbar

Genauigkeit der Zahlendarstellung

- DL Modelle können mit niedriger Genauigkeit effizient trainiert werden
- Zukünftige heterogene Architekturen werden unterschiedliche Zahlenformate mitbringen
- ***fp*16** als IEEE Standard mit 16-Bit Darstellung (Half-Precision)
- ***bfloat*16** als Alternative, in der größere Zahlen aber dafür weniger genau dargestellt werden können
- Gleiche Exponentengröße von ***fp*32** auf ***bfloat*16**

Format	*Mantisse*	*Exponent*	*min*	*max*
bfloat16	8 bits	8 bits	1.18e-38	3.39e+38
fp16	11 bits	5 bits	6.10e-05	6.55e+04
fp32	24 bits	8 bits	1.18e-38	3.40e+38

Tabelle: Vergleich von Gleitkommazahlformaten [Hig18]

Anwendung

- Als Zwischenschritt in einer Berechnungspipeline, um teure Berechnungen zu approximieren
  - Beispiel: Temperaturabhängigkeiten von Atomkombinationen aproximieren
  - Kombinationen liegen in $O(x^n)$
  - Trainingsset für ML Model in $O(1.000 - 10.000)$
  - Endergebnisse wieder ins Trainingsset
- Zur Generierung von künstlichen Datensets
  - Echte Messungen physikalisch aufwendig / nicht möglich
  - Elektronenmikroskopbilder von Kristallstrukturen
- Im Zusammenhang mit Bildkompressionen
  - Mannigfaltigkeit von Simulationsdaten und ML Daten
  - Komprimierte Bilder für Simulationen verwenden
  - Korrektur der Auflösung durch ML
  - Speichereinsparungen

Zusammenfassung

"HPC + AI"

- Herausforderungen
  - Heterogene Hardware
  - Kommunikation
- Native Frameworks (wie MagmaDNN) als Basis
- openDIEL als Verwaltung und Verbindung von Modulen
- Model- und Daten- Parallelismus
- ML-Einsatz als Pipelineteil (beispielsweise rechenintensive Vorberechnungen) oder für abgeschlossene Aufgaben (beispielsweise Wettervorhersagen auf Basis von Sensordaten)

Literatur

[aip18] Data-driven intelligent systems. university of hamburg. wtm department. lecture 11 theory of learning, evaluation. cornelius weber. 11 2018.

[Arc20] Rick et. all Archibald. Integrating deep learning in domain sciences at exascale. In *Driving Scientific and Engineering Discoveries Through the Convergence of HPC, Big Data and AI*, Cham, 2020. Springer International Publishing.

[Hig18] Nick Higham. Half precision arithmetic: fp16 versus bfloat16. 12 2018.

[Rec20] Prof. Dr. Thomas LudwigUniversität Hamburg – Informatik – Wissenschaftliches Rechnen. Hochleistungsrechnenvorlesung im wintersemester 2020/21, 2020.

[SDM11] John Shalf, Sudip Dosanjh, and John Morrison. Exascale computing technology challenges. In José M. Laginha M. Palma, Michel Daydé, Osni Marques, and João Correia Lopes, editors, *High Performance Computing for Computational Science – VECPAR 2010*, pages 1–25, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[Tea21] The Ray Team. Parameter server. 12 2021.

[top21] Top 500 liste. 11 2021.

[Tur21] Coury Turczyn. Sciencetechnology exascale computing's four biggest challenges and how they were overcome, 10 2021.