Ok - Schiek FZJ

Ok - Salva Jasmyn Meyaz FZJ

OK - Thomas Preusser AMD

Ok - Helge Gose: Nvidia

OK - Geesthacht Hereon

OK - Erik Thiessenhusen HZDR

Schier, Lüders HZDR

Artificial Intelligence - After more than 50 years still a promise for the future? (FZ Jülich)

Michael Schiek (m.schiek@fz-juelich.de)

Summary:

Artificial intelligence was a topic in mythology, literature - here especially in the subject of science fiction - long before this topic became a scientific topic. Latest with A. Turing's famous question 'Can machines think' [Turing, 'Computing Machinery and Intelligence', Mind 49, 1950] scientific research on AI experienced public attention. Following a definition by DARPA "Artificial Intelligence is the programmed ability to process information" and can be divided in the three waves "Handcrafted Knowledge", "Statistical Learning", and "Contextual Adaption" [Launchbury "A DARPA perspective on artificial intelligence", 2018]. Although it is believed that we are already within the third wave of AI, there is still a lack in understanding the principles of biological learning/computing and it is doubt that the standard von Neumann computing architecture will be capable for a technical realization of these principle especially in an energy efficient way. The Research Center Jülich Neuromorphic Computing roadmap addresses both challenges: On the one hand, the discovery of biological learning principles shall be supported by the development of specific hard- and software solutions for accelerated simulation of large-scale natural density biological neuronal networks. On the other hand, the memristor CMOS co-integration shall help to realize new computing architecture devices.

SHORT INTRODUCTION TO RESERVOIR COMPUTING (FZ Jülich)

Salwa Yasmeen Neyaz (y.n.salwa@fz-juelich.de)

Summary:

Realizing prediction functions using the conventional Recurrent Neural Network on Von Neumann architecture suffer not only from the problem of exploding and vanishing gradient, but also from high amount of power consumption. High power consumption occurs due to the rigorous training using Back Propagation through time algorithm where the parameters are updated frequently, and this requires movement of data between the memory and processing unit. Unlike in the brain, the memory and the processor are separated in the Von Neumann computing systems, with processor speeds increasing significantly over the years while memory has not been able to keep up the pace, and therefore the speed mismatch between the two causes a kind of bottleneck, limiting the efficiency.

Therefore, the problem is twofold here. One is to reduce the training costs, and the other is to reduce power consumption, which are somehow interdependent. Reservoir Computing - a Recurrent Neural Network based framework for temporally dependent data, allows for **computation in memory (CIM)**, and provides an advantage in terms of easier training. Reservoirs consists of an input layer, a reservoir whose dynamics are left to evolve (no training), and an output layer which is trained. The reservoir needs to satisfy specific criteria to work as a reservoir: a) Fading memory b) Separation Property c) Approximation. Although, no training of the reservoir is required, it would still need to store the dynamics frequently in the memory and transfer them for processing. Now, what could be done is to replace the reservoir node with an element that has a memory and processing capabilities. One of the elements could be memristor or which has these properties. Several reservoirs are mentioned in the literature such as the memristor-based ones, whose performance is compared using the standard benchmarking tasks. One of the applications where the Reservoir Computing could be used is trajectory prediction in the context of obstacle avoidance for 3D navigating drones.

Innovative AI-Applications on Adaptive Computing Hardware (AMD)

Whereas the era of Moore growth in the development of general-purpose compute hardware is history, modern machine-learning workloads continue to grow their compute demands exponentially. Custom hardware specialization has become a key technique to bridge this growing gap. At the cost of fragmenting the available product market, it is able to turn generalization overhead into productive application performance. Programmable FPGA fabric is the silicon technology of choice offering affordable custom specialization with a fast, low-risk development cycle in increasingly fragmented market segments.

Machine-learning inference workloads pose several challenges, which are hard on cost and power budgets particularly in embedded and edge environments. Millions of model parameters demand storage, millions of operations (mostly multiply-accumulate) need to be performed and corresponding IO bandwidth needs to back the computation. Mitigation techniques offer critical relief. The technique of batching leverages bulk IO and parameter caching to increase the compute throughput at the cost of latency. The quantization of the compute, particularly of non-input layers, helps throughout by reducing storage, IO and power requirements. Exploiting the benefits of very low-precision implementations below 8 bits, however, typically requires a quantization-aware retraining of the underlying model. Finally, pruning and sparsity allow to chop away on the compute requirements. While architectures leveraging a designated, physically separated compute engine must choose safe compromises for quantization and sparsity that fit all implemented layers, it is dataflow architectures spatially unrolled in FPGA fabric that can completely dimension and customize each layer individually. This regularly makes very low-precision compute with 4-bit or even binary operands a viable and attractive option.

AMD offers various solutions that support both (a) architectures with a designated compute engine (Vitis AI) and (b) co-design flows for quantized custom dataflow implementations (FINN/Brevitas, LogicNets). All of these offers are backed by open-source repositories on GitHub.

Edge Systeme und KI für Datenfassung und –auswertung (NVIDIA)

NVIDIA unterstützt als bekannter GPU-Hersteller eine Reihe unterschiedlicher Plattformen, die sich durch spezielle Architekturen und hochparallele Verarbeitung von CPU's abgrenzen.

Dabei wird der gesamte Technologiestack von der GPU bis zur Anwendung unterstützt. Cuda als dedizierte Sprache hat sich durchgesetzt und wird von unterschiedlichen Plattformen wie beispielsweise „Tensorflow“ oder PyTorch als GPU-Bibliothek zur Implementierung von AI-Algorithmen verwendet. GPU werden in unterschiedlichen Leistungsklassen mit unterschiedlichen Berechnungs-Features und Formfaktoren angeboten. Ein Abgleich mit der jeweils vorhandenen Hardware ist daher empfehlenswert.

NVIDIA bietet verschiedene Pakete wie beispielsweise Docker für die Implementierung an und unterstützt verschiedene Ökosysteme und AI-Modelle für eine rasche Implementierung.

Einige Links sollen die Orientierung erleichtern und den Weg zu Repositorien:

NVIDIA NGC Software Repository: https://catalog.ngc.nvidia.com/

Vortrainierte Modelle: https://catalog.ngc.nvidia.com/models

NVIDIA Tools & Ecosystem : https://developer.nvidia.com/tools-ecosystem

Bibliotheken optimiert für GPU: https://developer.nvidia.com/gpu-accelerated-libraries

Robotics und Embedded SOC:

https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/

Datacenter GPU: https://www.nvidia.com/en-us/data-center/products/

Klassifizierung der Neutronendetektorsignale mit Hilfe von maschinellem Lernen auf FPGAs (HEREON) joerg.burmester@hereon.de

Bei dieser Masterarbeit soll untersucht werden, ob es möglich ist mit Hilfe des maschinellen Lernens Neutronendetektor-Impulse zu kategorisieren. Es sollen mehrere Methoden überwacht und unüberwachte angewendet werden und hinsichtlich der Trennschärfe analysiert werden. Die entwickelten Algorithmen sollen dann in ein FPGA übertragen werden, so dass es möglich ist jeden einzelnen Neutronenimpuls einzuordnen nach Rauschen, Störung, Gamma- und Neutronen-Impuls. Die höchst möglichste Erfassungsrate sollte ermittelt werden, um eine Kosten zu Aufwandschätzung machen zu können.

Reconstruction of Small-Angle X-ray Scattering data using Invertible Neural Networks (HZDR)

The understanding of laser-solid interactions is important to the development of future laser-driven particle and photon sources, e.g., for tumor therapy, astrophysics or fusion. Currently, these interactions can only be modeled by simulations which need verification in the real world. Consequently, in 2016, a pump-probe experiment was conducted by Thomas Kluge to examine the laser-plasma interaction that occurs when an ultrahigh-intensity laser hits a solid density target. To handle the nanometer spatial and femtosecond temporal resolution of the laser-plasma interactions, Small-Angle X-Ray Scattering.

(SAXS) was used as a diagnostic to reconstruct the laser-driven target. However, the reconstruction of the target from the SAXS diffraction pattern is an inverse problem which are often ambiguous, due to the phase problem, and has no closed-form solution. We aim to simplify the process of reconstructing the target from SAXS images by employing Neural Networks, due to their speed and generalization capabilities. To be more specific, we use a conditional Invertible Neural Network (cINN), a type of Normalizing Flows, to resolve the ambiguities of the target with a probability density distribution. The target in this case is modelled by a simple grating function with three parameters. We chose this analytically well-defined and relatively simple target as a trial run for Neural Networks in this field to pave the way for more sophisticated targets and methods. Unfortunately, we don't have enough and reliable experimental data that could be used as training. So, in consequence, the network is trained only on simulated diffraction patterns and their respective ground truth parameters. The cINN is able to accurately reconstruct simulated- as well as preshot data. The performance on main-shot data remains unclear due to the fact that the simulation might not be able to explain the governing processes.

Erstellung und Integration von Convolutional Neural Networks zur Elektronikschrotterkennung (HZDR)

Auf Convolutional Neural Networks (CNNs), zu Deutsch Faltungsnetze, liegt seit mehreren Jahren ein besonderer Fokus im Bereich des computerbasierten Sehens. Das liegt zum Teil an den vielseitigen Einsatzmöglichkeiten in verschiedenen Anwendungsbereichen. Zu diesen Anwendungsbereichen zählen beispielsweise die Detektion und Klassifizierung von Objekten auf Bildern. Aus diesem Grund wurde ein System zur Detektion von Elektronikschrott mit Hilfe eines CNNs entwickelt. Dieses System wurde auf die Erkennung und Lokalisierung von elektronischen Bauteilen wie beispielsweise Transistoren, Kondensatoren oder Spulen trainiert. Speziell wurde hierfür die YOLO-Architektur verwendet, die für eine schnelle Detektion auf einem einzelnen Bild optimiert ist. Ein weiterer Aspekt des Ansatzes ist eine Software zur Erstellung dynamischer Bildverarbeitungspipelines. Durch diese Software entsteht ein anpassbares System, um die Ergebnisse des Faltungsnetzes zu optimieren und verschiedene Szenarien zu generieren.

So kann die Entwicklung in zwei Schritte unterteilt werden. Der Erste bezieht sich auf die initiale Erstellung und dem Training des CNNs. Hierbei stehen die Methoden und Programme für die Entwicklung der CNNs im Mittelpunkt. Der zweite Schritt fokussiert sich auf die Integration eines Netzes in eine Bildverarbeitungspipeline mithilfe des entwickelten Softwaresystems. Dadurch können die Faltungsnetze einfach getestet und die Eingangsdaten an das CNN angepasst werden. Zudem ist es möglich, mittels Data Augmentation die Ergebnisse der Bildverarbeitungspipeline für die weitere Verbesserung der Erkennungsleistung des Netzes zu verwenden. Durch diese Herangehensweise können verschiedene Netze, die in einem geeigneten Framework trainiert wurden, problemlos eingesetzt und verbessert werden. Aufgrund der Nutzung von OpenCV existieren Schnittstellen zu allen gängigen Machine-Learning-Frameworks.