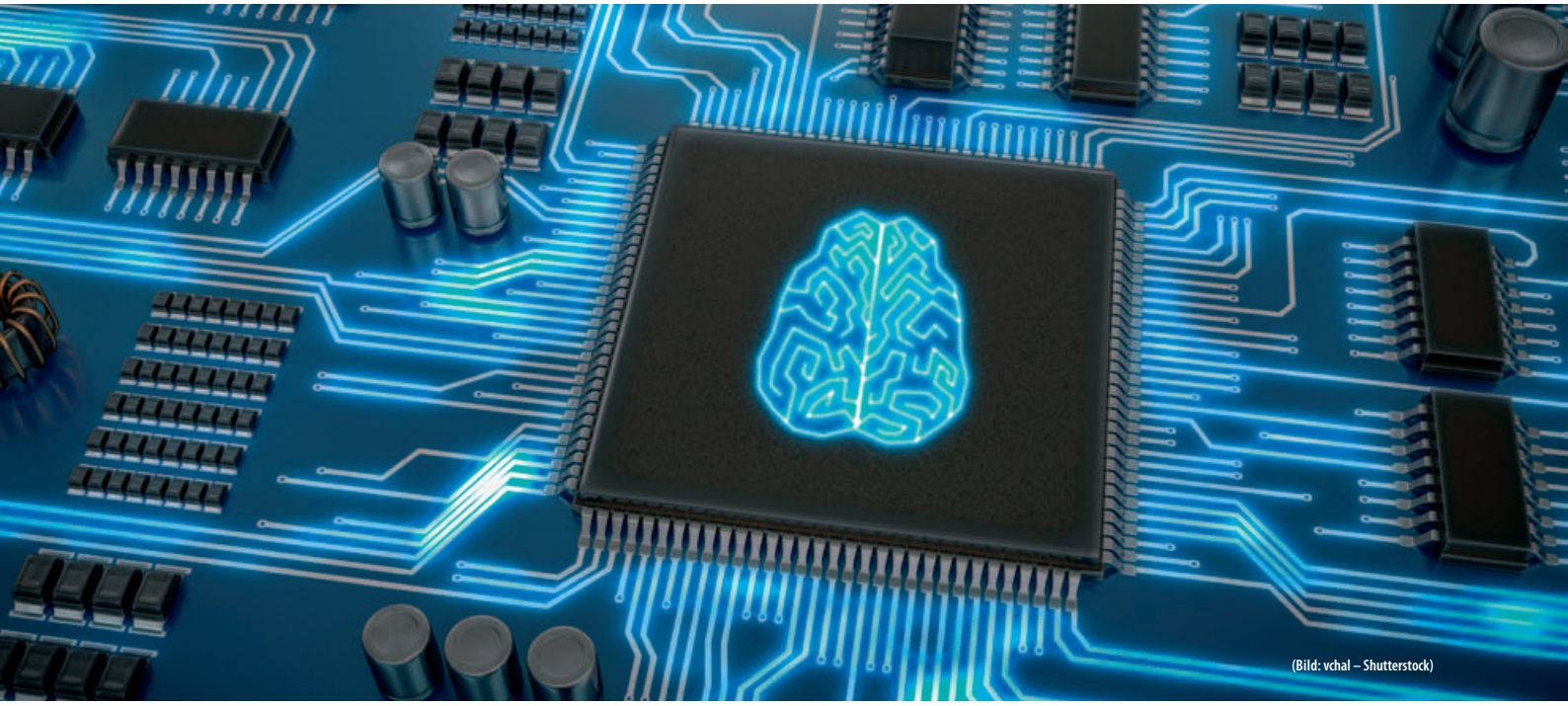
Machine Learning:

Elektronenhirn 4.0



(Bild: vchal – Shutterstock)

Die Anwendung von Deep Learning in Embedded-Geräten liegt stark im Trend. Eine Vielzahl an KI-Chips ist angekündigt, erste Produkte sind auf dem Markt. Sie versprechen intelligente Signalauswertung und Entscheidungen bei geringen Kosten und niedrigem Stromverbrauch.

Von Marco Breiling, Rastislav Struharik und Loreto Mateu

Lange Zeit war Deep Learning (DL) eine Domäne von leistungsfähigen PCs und (Cloud-) Servern. Während das Training von Deep Neural Networks (DNNs) sehr komplex ist, stellen bei der Inferenz simple Multiply-Accumulate-Anweisungen (MAC) den weitaus größten Anteil der Berechnungen dar. Im Gegensatz zu typischen Digital-Signal-Processing-Algorithmen (DSP) wie Filterung und Korrelation ist bei DNNs aber die Anzahl von Gewichtsfaktoren deutlich größer und geht bei typischen Bildverarbeitungs-DNNs in die Millionen. Schon früh wurde erkannt, dass Graphic Processing Units (GPUs) eine Struktur besitzen, mit der man sehr effizient diese Berechnungen durchführen kann. Fortan bauten GPU-Hersteller wie Nvidia gezielt Features in manche ihrer Architekturen ein, um deren DL-Eignung zu verbessern. Andere Unternehmen wie Google und Movidius erkannten allerdings das Potenzial, das dedizierte Chip-Architekturen für die Beschleunigung von DL besitzen. So brachte im Jahr 2016 Google seine TensorFlow Processing Unit (TPU) heraus. Parallel dazu hat Movidius – 2016 von Intel aufgekauft – eine Architektur namens Myriad 2 entwickelt. In Myriad 2 arbeiten zwölf Spezialprozessoren parallel zueinander. **Bild 1** zeigt beispielhaft in einer Embedded-Vision-Anwendung den Movidius Neural Compute Stick.

GPU – eine für alles?

Bild 2 verdeutlicht, was die Unterschiede der DL-Inferenz-Beschleuniger (DLI) gegenüber normalen CPUs und GPUs sind: Die Arithmetic Logic Unit (ALU) einer Single Core CPU kann gleichzeitig nur eine Berechnung durchführen. Noch kritischer ist aber, dass – je nach DNN – Millionen von Input-Werten und Gewichten über einen einzigen Datenbus von einer Random Access Memory (RAM) zur CPU gestreamt werden müssen. Trotz möglichst effizientem Pre-Caching dieser Werte begrenzt dieses sogenannte „von-Neumann-Bottleneck" der klassischen Computer-Architekturen den erreichbaren Berechnungsdurchsatz. Der Vorteil von GPUs ist dagegen, dass viele einfache ALUs parallel arbeiten können, und das parallele Speicherinterface von GPUs eine sehr hohe RAM-Bandbreite aufweist. Der Kontrollfluss in allen ALUs wird aber von derselben Control Unit (CU) gesteuert, sodass in jeder ALU die gleiche Operation durchgeführt wird: Single Instruction Multiple Data (SIMD). Etwas flexibler sind Single-Instruction-Multiple-Thread-Architekturen (SIMT), bei denen einzelne ALUs wenn nötig auch Operationen auslassen können.

Das „Neue" bei DLI-Beschleunigern ist, dass viele einfache Processing Elements (PEs) mit eigenen ALUs und eigenen CUs eingesetzt werden, zum Beispiel zwölf Cores beim Myriad 2. Weiterhin sind die Datenfluss-Strukturen in DLI-Beschleu-

nigern speziell auf die sehr schnelle Heranführung von Input-Werten und Gewichten sowie deren Weitergabe an die anderen PEs optimiert. Später wird noch gezeigt, dass das Vorhandensein einer CU in jedem PE eine größere Flexibilität bei der Verarbeitung der Datenflüsse erlaubt. Dadurch werden die vorhandenen Strukturen in DNNs effizient ausgenutzt.

Deutschland hat Trend verschlafen

Während GPUs für Embedded-Anwendungen relativ teuer sind, stoßen DLI-Beschleuniger bereits in die Preiskategorie von etwa zehn Euro vor. Neben dedizierten Chips haben Anbieter wie Synopsys, Cadence und Xilinx inzwischen auch Intellectual Property-Cores (IP) im Angebot. Arm Limited hat unter dem Namen „Project Trillium" eigene Machine-Learning-Varianten seiner IP-Cores angekündigt (**Tabelle**). Manche IP-Cores können auch als Co-Prozessoren eines Hauptprozessors für DLI-Berechnungen eingesetzt werden. Seit zwei Jahren rollt eine regelrechte Flutwelle an Produktankündigungen sowohl für Chips als auch IP-Cores durch die Presse. In den USA und in China planen sowohl etablierte Firmen wie Qualcomm und Huawei als auch unbekannte Start-Ups weitere DLI-Beschleuniger. Eine Vorschau von Tractica schätzt den DL-Beschleuniger-Markt im Jahr 2025 auf über 30 Milliarden Dollar [2]. In Europa, speziell auch in Deutschland, wurde dieser Trend verschlafen. Erst jetzt werden deutsche Firmen verstärkt aktiv, um den Vorsprung der Konkurrenz wieder einzuholen. Jenseits des milliardenschweren EU-Flagship-Projekts „The Human Brain Project" (HBP) gab es bisher leider auch in Deutschlands akademischer Welt nur wenige Aktivitäten zur Hardware-Beschleunigung von KI-Berechnungen.

Anwendungsbeispiele für Embedded DLI

Die neuen KI-Chips und IP-Cores erlauben vielfältige neue Anwendungen jenseits von Desktop-PCs und Servern. In (teil-)autonomen Fahrzeugen der Stufen drei bis fünf werden zukünftig massiv DLI-Systeme zur Umfeldüberwachung und Entscheidungsfindung eingesetzt, um die großen Datenmengen der Radar-, LiDAR (Light Detection and Ranging)- und Kamerasensoren auswerten zu können. Daneben gibt es bereits Künstliche-Intelligenz-Chips (KI) in einigen Mobiltelefonen, Überwachungskameras und Drohnen. Bei letzteren werden Kamerabilder in der Drohne ausgewertet. Dies beweist, wie groß die Rechenperformance in Tera Operations Per Second (TOPS) bei vergleichsweise geringer Leistungsaufnahme ist (s. Tabelle). Die Alternative zu Embedded-DLI wäre eine Übertragung von Sensordaten wie etwa Kamerabildern zur Auswertung in die Cloud. Allerdings ist das Embedded-Gerät dann auf Gedeih und Verderb auf die Zuverlässigkeit und ausreichende Datenrate des Kommunikations-Links angewiesen: Fällt dieser zum Beispiel in einem Tunnel aus, so fällt auch das Gerät aus. Weitere potenzielle Probleme sind die Kosten und die Stromaufnahme der Datenübertragung, zu hohe Latenzen – wenn auf der Basis von Sensorwerten Aktoren geregelt werden müssen – und der Schutz der Daten vor Ausspähen und Manipulation (Privacy und Security).
Wegen dieser Gründe wird Embedded-DLI in vielen zukünftigen Anwendungen lokal eingesetzt werden. Bei der Qualitätskontrolle in der Produktion und der Verkehrsüberwachung, bei Anomalie-Detektions- oder Predictive-Maintenance-Systemen wie in Haushaltsgeräten oder aber in vermeintlich exotischen smarten Geräten wie Mülleimern, die anrufen, wenn sie geleert werden müs-

Bild 1. Embedded-Vision-Anwendung mit dem Intel Movidius Neural Compute Stick.

(Bild: Fraunhofer IIS)

DLI-Beschleuniger	Typ	Zielanwendung	Performance
NVIDIA Jetson Nano	GPU-Plattform	Embedded	472 GOPS @ 5 – 10 W
Nvidia Jetson TX2	GPU-Plattform	Edge	1,3 TOPS @ 7,5 W
NVIDIA Jetson AGX Xavier	GPU-Plattform	Edge	30 TOPS @ 30 W
NVIDIA Drive AGX Pegasus	GPU-Plattform	Automotive	320 TOPS
Intel Movidius Myriad 2 bzw. Myriad X	Chip	Embedded/Edge DL/Vision	4 TOPS @ 1 W (Myriad X)
MobilEye EyeQ4	Chip	Automotive	2.5 TOPS @ 3 W
CEVA NeuPRO (und XM6)	Soft IP-Core	Embedded DL/Vision	2 – 12,5 TOPS
GreenWaves GAP8	Chip	Battery powered AI	200 MOPS bis 8 GOPS @ < 100 mW
Gyrfalcon Lightspeeur 2801/2802/2803	Chip	Edge & Cloud	16,8 TOPS @ 700mW (Typ 2803)
Canaan Kendryte K210	Chip	Embedded Vision & Audio	250 GOPS @ 300mW (bis 500 GOPS)
Google Coral Edge TPU	Chip	Edge	4 TOPS @ <2,5W
Bitmain Sophon BM1682/1682/1880	Chip	Edge & Cloud	3 TOPS (Typ BM1682)
Cadence DNA 100	Soft IP-Core	Embedded DL	0,5 – 12 TOPS @ 3,4 TOPS/W *
Cadence Tensilica Vision P6 DSP	Soft IP-Core	Embedded Vision	< 12 TOPS @ 3,4 TOPS / W *
Synopsys DesignWare EV Processor	Soft IP-Core	Embedded Vision	4,5 TOPS @ 2 TOPS/W *
ARM ML (und OD)	Soft IP-Core	Embedded DL/Vision	4,6 TOPS @ 1,5 W *
Xilinx DeePhi	Soft IP-Core	Embedded DL/Vision	0,23 TOPS § @ 3 W #
Kortiq AISCale	Soft IP-Core	Embedded/Edge DL/Vision	0,32-2,56 TOPS § @ 1,2 – 2,5 W #
Lattice sensAI Stack	Soft IP-Core	Embedded	<1 mW – 1 W
Videantis v-MP6000UDX	Soft IP-Core	Embedded DL/Vision	<6,6 TOPS @ 400 MHz

*: für ASIC-Implementierung (7 bzw. 16 nm) // #: für Xilinx-Zynq FPGA // §: Wert gilt beim Einsatz von Deep Compression und ist äquivalent zu einem vielfach höheren Wert (ca. 10x) bei konventionellen DLI-Beschleunigern [1]

Tabelle. Übersicht über aktuell verfügbare digitale Deep-Learning-Inference-Beschleuniger (außer für Mobiltelefone).

sen. Wie beim Mikrocontroller wird auch in diesem Bereich wohl die Zahl der Anwendungsideen stetig mit der Leistungsfähigkeit der Chips wachsen. Das DL-Training wird zumindest in den nächsten Jahren noch überwiegend Off-Line und auf Servern stattfinden. Die dort gewonnenen DNN-Parameter werden dann während der Geräteherstellung oder später per Firmware-Update an das Embedded-System übertragen.
Wie erwähnt, nutzen DLI-Beschleuniger gewisse Strukturen in DNNs aus. So weisen die Berechnungsmuster in Convolutional Layers (CLs) einige Merkmale auf, für die effiziente CL-Rechenarchitekturen entwickelt worden sind. Beispielsweise gibt es eine signifikante Datenwiederverwendung: Jeder CL verwendet eine Reihe von 3D-Filtern, die über die Input-Werte – Input Feature Map (IFM) – bewegt werden. Dieses Bewegungsmuster ist konstant und vorhersehbar für jeden gegebenen CL. In Anbetracht dessen kann der RAM-Transfer zwischen CL-Beschleuniger und externem Speicher signifikant reduziert werden. Dazu werden Segmente der Input Feature Map in On-Chip-Cache-Speichern gespeichert und aktuelle 3D-Filterkoeffizienten in den lokalen On-Chip-Speicher vorgeladen.

Embedded-DNN-Einsatz schwierig

Dies ist auch nötig, da CLs in der Regel sowohl sehr rechen- als auch äußerst speicherintensiv sind. So verfügt beispielsweise ein VGG-16-DNN über mehr als 138 Millionen Gewichtswerte. Bei einer 16-bit-Zahlendarstellung müssen dafür etwa 276 MB Speicherplatz zur Verfügung stehen. Da die meisten CL-Beschleuniger die Input und Output Feature Maps nicht On-Chip speichern können, müssen sie extern gespeichert und während der CL-Verarbeitung zwischen CL-Beschleuniger und externem Speicher hin und her bewegt werden. Um ein Eingangsbild zu verarbeiten, erreicht die Datenmenge des VGG-16-DNN dadurch schnell bis zu 60 MB. Wenn man die Anzahl der erforderlichen Berechnungen analysiert, um ein Eingangsbild zu klassifizieren, erhält man im Falle des VGG-16 über 15 Milliarden MAC-Operationen (GMAC). Dies macht den Einsatz von solchen DNNs in Embedded-Anwendungen sehr schwierig, insbesondere da in den meisten Fällen strenge Anforderungen an Latenz, Durchsatz und/oder Leistungsaufnahme vorhanden sind. Insbesondere der benötigte massive Speichertransfer zum externen Off-Chip-RAM verschlingt eine sehr große Leistung – deutlich mehr als Speichertransfers, die on-Chip bleiben können.

Speicherreduzierung macht´s möglich

Es ist aber möglich, die für die Speicherung der Gewichte und vorläufigen Feature Maps erforderliche Speichergröße zu reduzieren, indem Deep Compression verwendet wird, das aus mehreren Kompressionsverfahren besteht [4]. Diese Verfahren entfernen einige der Gewichte, in der Regel aus Convolutional und Fully Connected Layern, basierend auf einer Wertung ihrer Bedeutung für den Output des DNN. Weniger wichtige Parameter können entfernt werden, das heißt auf den Wert Null gesetzt werden. Wird dies korrekt durchgeführt, kommt es zu keiner Beeinträchtigung der Klassifizierungsleistung und Genauigkeit des „beschnittenen" Netzwerks. Nachdem der Beschneidungsprozess (Pruning oder Synapseneliminierung) abgeschlossen ist, bleibt eine sogenannte „Sparse Representation" übrig, also eine ausgedünnte Darstellung des DNN **(Bild 3)**.

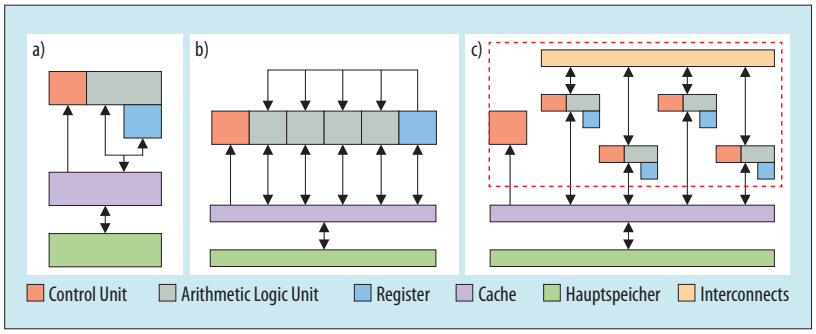


Bild 2. Vergleich von einer klassischen CPU-Architektur (a), einer GPU (b) und eines typischen digitalen Deep-Learning-Inference-Beschleunigers (c). (Quelle: Fraunhofer IIS)

Erstaunlicherweise besitzen DNNs scheinbar eine sehr hohe Redundanz, sodass extrem hohe Beschneidungsgrade auf DNNs angewendet werden können. Werte von mehr als 80 Prozent Pruning der Parameter sind keine Seltenheit – ohne die Genauigkeit des Netzwerks zu beeinträchtigen. Die Anzahl von Nullen kann mit Standard-Datenkompressionsalgorithmen komprimiert werden, um den Speicherbedarf deutlich zu reduzieren. So kann beispielsweise bei VGG-16 mit entsprechenden Beschneidungsalgorithmen die Speichergröße von 276 MB auf 5,5 MB – also um 98 Prozent – reduziert werden.

Verbessern der Effizienz

Um die Energie- und Recheneffizienz weiter zu verbessern, kann die Statistik der Input und Output Feature Maps, die im DNN verarbeitet werden, untersucht werden. Die Untersuchung erfolgt, um den erforderlichen Speicherzugriff durch Feature-Map-Kompression weiter zu reduzieren. Eine effiziente Feature-Map-Kompression ist bei Nutzung der Rectified-Linear-Unit-Aktivierungsfunktion (ReLU) in einem Layer möglich. Die ReLU-Aktivierungsfunktion führt zu vielen Nullen und damit einer Ausdünnung in den Output Feature Maps, weil sie alle negativen Werte an ihrem Eingang auf Null korrigiert. Obwohl die Anzahl der Nullen in den Feature Maps grundsätzlich von den Eingangsdaten des DNN abhängt, nimmt sie dennoch tendenziell in hinteren Layern zu. So sind beispielsweise im VGG-16-DNN im Durchschnitt fast 48 % der IFM-Werte des Conv1_2-Layers Null. Der Wert wächst auf bis zu 88 % für das Conv5_3-Layer.
Neben der Reduzierung der erforderlichen Speichergröße und der Datenmenge während der DNN-Berechnung kann die ausgedünnte Darstellung von DNNs und Feature Maps auch dazu verwendet werden, die Inferenz-Geschwindigkeit durch Optimierung des Berechnungsprozesses von Convolutional- und Fully Connected Layern zu beschleunigen. Da in diesen Layern sehr viele Multiplikationen zwischen Input-Werten und Gewichten vorkommen, wird die Berechnung der Produktsumme effizienter, wenn sie alle Gewichte mit Wert Null überspringt (Zero Skipping – siehe **Bild 4**). Durch diese Begrenzung auf die Durchführung der absolut notwendigen Berechnungen wird eine signifikante

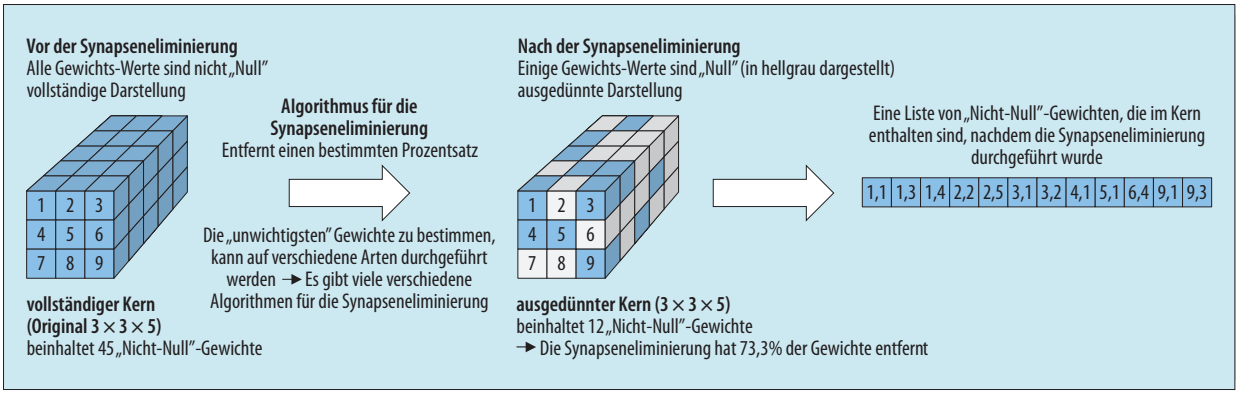


Bild 3. Anwendung von Pruning (Synapseneliminierung) zur Ausdünnung der Gewichte eines dreidimensionalen Filters. (Quelle: Kortiq)

Verkürzung der Inferenzdauer bei gleichzeitiger Reduzierung der Leistungsaufnahme erreicht.

Höhere Geschwindigkeit durch „AIScale"

Die „AIScale"-Architektur des deutschen Start-ups Kortiq wurde speziell entwickelt, um diese Optimierungsmöglichkeiten auszunutzen **(Bild 5)**. Sie ist in der Lage, alle Multiplikationen, die zu einem nullwertigen Ergebnis führen, sofort zu erkennen und deren Berechnung zu überspringen, wodurch die Bearbeitungszeit des Layers stark reduziert wird. Dies geschieht, wenn Convolutional-, Pooling- und Fully Connected Layer im AIScale IP-Core verarbeitet werden. Darüber hinaus sind die Compute Cores (CCs) des AIScale in der Lage, Feature- und Gewichts-Daten in komprimierter Form zu verarbeiten, wodurch der sonst nötige Dekompressionsschritt entfällt. Das verkürzt die Verarbeitungszeit weiter und reduziert die Größe der On-Chip-Speicher.

Durch die „Zero-Skipping"-Technik ist AIScale in der Lage, die DNN-Verarbeitungsgeschwindigkeit um das drei- bis zehnfache zu erhöhen, verglichen mit herkömmlichen GPU/CPU-Architekturen, die kein Zero-Skipping verwenden. AIScale wurde als Standard-Field-Programmable-Gate-Array (FPGA)-Soft-IP-Core für die Xilinx-FPGA-Familien entwickelt, kann aber leicht an die Bedürfnisse anderer FPGA-Hersteller wie Intel oder Lattice angepasst oder sogar mit Application-Specific-Integrated-Circuit-Technik (ASIC) implementiert werden. Dabei ist AIScale ein universeller DNN-Beschleuniger, der verschiedene DNN-Familien wie Inception, ResNet, MobileNet, NASNet oder SqueezeNet beschleunigen kann.

Nach der geschilderten Ausnutzung der DNN-spezifischen Strukturen erscheint das zukünftig noch mögliche Entwicklungspotenzial von DLI-Beschleunigern zunächst gering. Allerdings darf nicht vergessen werden, dass auch Deep Learning selbst noch bei weitem nicht „ausentwickelt" ist. Im Abstand weniger Jahre tauchen immer wieder komplett neue Konzepte wie Long-Short-Term-Memory, Inception Layers und Deep Compression auf, die auf bisherigen DLI-Architekturen meist nicht ausgeführt werden können. Außerdem sind die eingesetzten DNN-Modelle und deren Hardware-Anforderungen stark anwendungsabhängig, beispielsweise bei Ultra-low-power-Anwendungen für Zeitreihen-Verarbeitung. Somit ist auch zukünftig noch viel Luft nach oben für Anpassungen von vorhandenen Architekturen und Entwicklung grundlegend neuer Architekturen wie AIScale.

Spiking Neural Networks (SNNs)

Neben den bis hierher ausschließlich betrachteten digitalen DL-Beschleunigern beherbergt der KI-Chip-Zoo aber noch weitere Gattungen. Diese unterscheiden sich in der Durchführung der Berechnungen: Bei DNNs laufen alle Berechnungen Layer-weise wie eine Welle durch das neuronale Netz – das heißt, zuerst wird Layer 1 komplett berechnet, dann Layer 2 und so weiter.

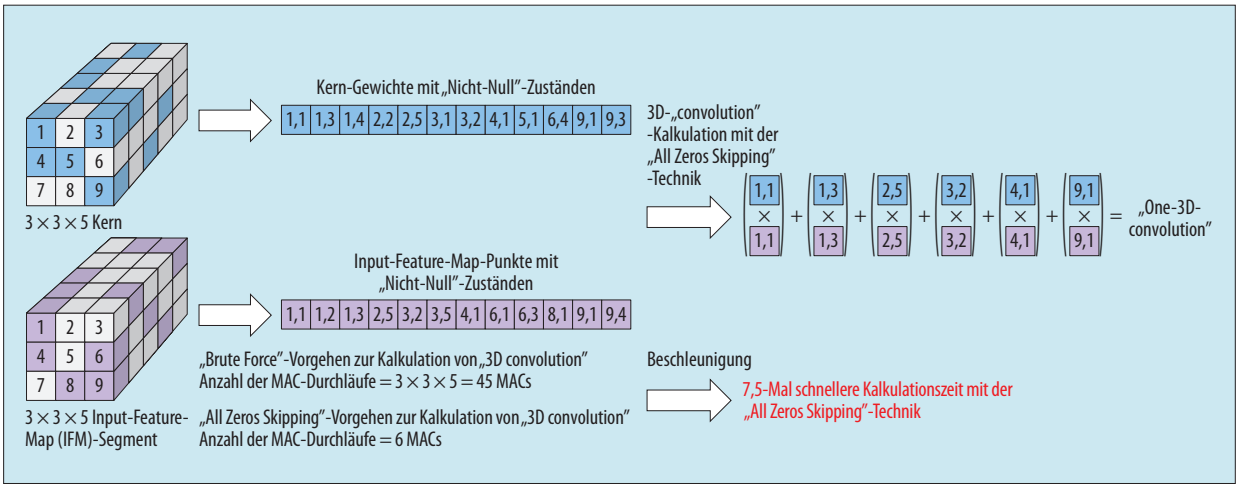


Bild 4. Reduzierung der Berechnungen durch Überspringen von Multiplikationen mit Null (Zero Skipping). (Quelle: Kortiq)

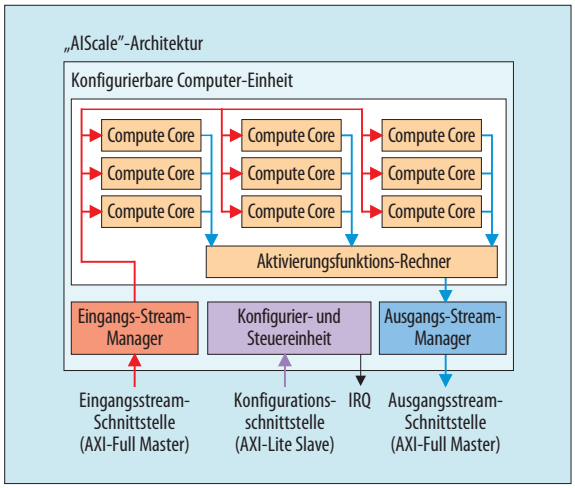


Bild 5. Blockdiagramm der AIScale-Architektur. (Quelle: Kortiq)

Insofern ist ein Neuron im Layer 1 erst einmal passiv, nachdem es in der aktuellen Inferenz berechnet wurde, und es wird erst wieder in der nächsten Inferenz aktiv. Eine Alternative dazu stellen Spiking Neural Networks (SNNs) dar. Die Netzwerk-Topologie von SNNs ähnelt prinzipiell der von DNNs. Allerdings können hier die Neuronen wie beim biologischen Gehirn prinzipiell zu jedem Zeitpunkt aktiv werden. Sie senden dann kurze Impulse, sogenannte Spikes, zu ihren Nachfolge-Neuronen. Je mehr ein Neuron von seinen Vorgängern angeregt wird, umso häufiger pulst es selbst. Die Pulsfrequenz ist also das Äquivalent zum Ausgangswert (Output Feature) eines DNN.

Da SNNs ans Hirn angelehnt sind, werden sie – wie die analogen Äquivalente, die weiter unten vorgestellt werden – auch als neuromorphe Schaltungen bezeichnet. Dieser Begriff ist aber schwammig, teilweise werden auch DL-Beschleuniger dazugezählt. An einfachen neuromorphen Schaltungen wird faktisch schon seit Jahrzehnten geforscht. Enormen Auftrieb hat das Thema aber erst in den letzten Jahren erhalten, zum einen durch die Erfolge von DL in der Praxis, zum anderen durch die akademische Forschung im HBP. In dem Projekt von der University of Manchester und der TU Dresden wurde die „SpiNNaker"-Chip-Architektur entwickelt. Darüber hinaus haben IBM mit seinem „TrueNorth" und Intel mit dem „Loihi" bereits digitale SNN-Chips produziert, die allerdings nicht kommerziell verfügbar sind. Man erwartet von diesen SNN-Chips bei der gleichen Aufgabe eine nochmals geringere Leistungsaufnahme und kürzere Latenzen gegenüber DLI-Beschleunigern. Weiterhin gibt es Ansätze, wie SNN-Chips während der Inferenz weiter trainiert werden können. Dadurch können die Chips für „Reinforcement Learning" eingesetzt werden. Auf der anderen Seite gibt es aber beim Thema Training und Hyper-Parameter-Auswahl für SNNs immer noch zahlreiche offene Fragen, während bei DNNs das Wissen deutlich fortgeschrittener ist. Mit dem „Akida" von BrainChip ist für 2019 bereits ein kommerzieller SNN-Chip angekündigt, eine größere Anzahl von Produkten und Anwendungen ist jedoch erst in einigen Jahren zu erwarten.

Analoge DLI-Beschleuniger

Bis hierher war die Rede rein von digitalen Schaltkreisen. Wie bereits erwähnt, gibt es die beiden Kategorien DLI-Beschleuniger und SNN-Chips auch als teilweise analoge Variante. Neben den bereits beschriebenen bedeutenden Fortschritten bei den digitalen Varianten wurde in letzter Zeit auch viel Aufwand für die Entwicklung von analogen DLI-Beschleunigern betrieben. Dieser Begriff bezeichnet analoge speicherbasierte Schaltungen, welche die MAC-Berechnungen mit Hilfe einer Speichermatrix mit Crossbar-Architektur vornehmen. Im Crossbar sind alle Gewichtsfaktoren der Synapsen als Leitwerte von memristiven Komponenten gespeichert. Memristoren sind elementare Komponenten, deren Widerstandswert durch Anlegen eines Stroms einprogrammiert werden kann und dann auch ohne diesen Strom gehalten wird.

Wie in **Bild 6** gezeigt, nutzt der Crossbar-Ansatz das Ohmsche und Kirchhoffsche Gesetz für eine analoge MAC-Berechnung. Jeder Input-Wert einer Synapse wird als Spannungswert U an den jeweiligen Memristor angelegt, dessen Leitwert G gleich dem Gewicht ist. Der dann fließende Strom I ist das Produkt aus $I = G \times U$. Die Ströme aller Synapsen eines Neurons treiben eine gemeinsame Leitung und addieren sich damit. Der Summenstrom des Neurons muss anschließend noch eine nicht-lineare Aktivierungsfunktion passieren und stellt dann den Ausgangswert des Neurons dar. Weil die MAC-Berechnung dort stattfindet, wo die Speicher sind (In-Memory-Computation), müssen die Gewichte nicht von anderen On- oder gar Off-Chip-Speichern zur CPU gestreamt werden. Außerdem läuft die Berechnung aller MAC-Operationen voll parallel. Deswegen verbessert die analoge Cross-Bar-Architektur sowohl die Inferenz-Geschwindigkeit als auch die Energieeffizienz [6]. Aus diesem Grund sind analoge Beschleuniger sehr gut für Fully Connected Layer geeignet, während Convolutional Layer auch von digitalen Beschleunigern berechnet werden können [6].

Crossbar-Architektur

Diese Architektur benötigt resistive Speicherzellen (Memristoren) wie RRAM (Resistive RAM), PCM (Phase-Change

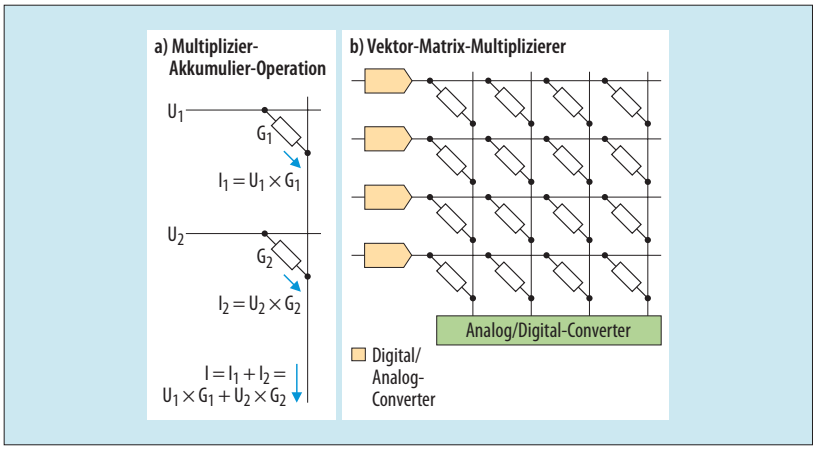


Bild 6. Berechung eines Neurons durch analoge MAC-Operationen (a), analoge Speichermatrix mit Crossbar-Architektur zur Berechnung eines fully-connected Layers (b). (Quelle: Fraunhofer IIS)

Memory), MRAM (Magnetic RAM), STT-MRAM (Spin Transfer Torque Magnetic RAM), CBRAM (Conductive Bridge RAM), OxRAM (Oxide Based RRAM) oder FeRAM (Ferroelectric RAM), die alle nichtflüchtig sind (Embedded Non-Volatile RAM). Sie unterscheiden sich voneinander durch die Anzahl der möglichen Schreibzyklen, die Speicherungszeit – um einen Leitwert zu programmieren – und die physikalische Dichte [3, 6]. Die Memristoren weisen einen Zustand mit hohem Widerstand und einen mit niedrigem Widerstand auf. Der Wechsel zwischen beiden Zuständen erfordert das Anlegen einer bestimmten Spannung über eine Mindestzeit.

Die Crossbar-Architektur benötigt sowohl Analog-Digital-Converter (ADCs) als auch Digital-Analog-Converter (DACs) an ihrer Peripherie. DACs sind für die Umwandlung der Input-Werte der Synapsen in Spannungen U_1, U_2 bis U_n nötig. Diese Werte liegen in der Regel digital gespeichert vor. Ein ADC ist für die Umsetzung des summierten Ausgangsstroms erforderlich. Der ADC hilft dabei, den Ausgangswert wieder digital abzuspeichern. Die Anforderungen dieser Umsetzer könnten die gesamte Fläche und Leistungsaufnahme der Schaltung extrem erhöhen. Deswegen muss die Wortbreite der ADCs und DACs begrenzt und dabei die Rechengenauigkeit des DNNs durch geeignete Techniken erhalten werden [5]. Der „errechnete" Ausgangsstrom des Crossbars kann statt als Digitalwert auch als Spike ausgegeben werden. Dafür wird eine Art Schmitt-Trigger-Schaltung benötigt, bei der die Frequenz der Ausgangsspikes dann proportional zu dem Ausgangstrom des Crossbars ist.

Die noch am weitesten in der Zukunft liegende Art von neuromorpher Hardware sind schließlich analoge SNN-Schaltungen, bei denen in analogen Neuronen Spikes erzeugt werden – fast so wie im menschlichen Gehirn. In Deutschland hat die Universität Heidelberg mit der „BrainScaleS"-Architektur im HBP eine sehr anspruchsvolle analog-digitale SNN-Schaltung entwickelt. Um zwischen den hier vorgestellten verschiedenen Ansätzen für neuromorphe Hardware vergleichen zu können, sollten als Metriken unter anderem Leistungsaufnahme, Latenz, Durchsatz und Kosten berücksichtigt werden [3]. Weitere Metriken können die Recheneffizienz (Anzahl von 16-bit-äquivalenten Operationen pro Sekunde und pro Quadratmillimeter), die Leistungseffizienz (Anzahl von 16-bit-äquivalenten Operationen pro Watt) und die Speichereffizienz (On-chip Kapazität für die Gewichtsfaktoren pro Quadratmillimeter) sein [5].

Aufholjagd

Die Entwicklung von KI-Chips, das heißt DL-Beschleunigern und neuromorpher Hardware, hat seit 2015 eine ungeahnte Dynamik entwickelt. Diese Dynamik wird auch in den nächsten Jahren anhalten, einige vorhandene Anwendungen stark umkrempeln und viele gänzlich neue Anwendungen hervorbringen. Eine weitere Reduktion der Leistungsaufnahme um mehr als zwei Größenordnungen ist dabei durchaus realistisch. Insofern stellt diese neue Halbleiter-Kategorie eine große Chance für neue Anwendungen in unterschiedlichsten Wirtschaftsbranchen dar. China hat das schon vor längerem begriffen, Europa und Deutschland sollten schnell nachziehen. *ts*

Literatur:

[1] **https://www.iis.fraunhofer.de/de/ff/kom/iot/neuromorphic.html**

[2] **https://www.tractica.com/newsroom/press-releases/deep-learning-chipset-market-to-reach-66-3-billion-by-2025/**

[3] ***Sze V., Chen Y., Yang T. and Emer J.S.*: Efficient Processing of Deep Neural Networks: A Tutorial and Survey. in Proceedings of the IEEE. vol. 105. no. 12. pp. 2295-2329. Dec. 2017.**

[4] ***Han S., Mao H., Dally W.J.*: Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. International Conference on Learning Representations. 2015.**

[5] ***Shafiee, A.*: ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars. International Symposium on Computer Architecture.2016.**

[6] ***Tsai H.*: Recent progress in analog memory-based accelerator for deep learning. Journal of PhysicsD.2018.**

Dr. Marco Breiling studierte Elektrotechnik in Karlsruhe, Trondheim, Paris und Southampton und promovierte in Erlangen. Er ist seit 2001 am Fraunhofer Institut für Integrierte Schaltungen IIS in Erlangen tätig. Dort koordiniert er als Chief Scientist die neuromorphen Hardware-Aktivitäten.
marco.breiling@iis.fraunhofer.de

Prof. Dr. Rastislav Struharik ist CTO von Kortiq und außerordentlicher Professor am Institut für Energie, Elektronik und Telekommunikation an der Universität Novi Sad. Er erhielt den B.Sc., M.Sc. und Ph.D. in Elektronik und Telekommunikation an der Universität von Novi Sad in den Jahren 1999, 2005 und 2009. Dort ist er auch Leiter der Abteilung für Embedded Systems and Algorithms. Seine Forschungen und Interessen umfassen rekonfigurierbares Computing, Hardwarebeschleunigung von Algorithmen, Machine Learning, DSP-Algorithmen und Design und Verifikation von komplexen digitalen Systemen.
rastislav.struharik@kortiq.com

Dr. Loreto Mateu erwarb ihren Bachelor in Industrial Engineering an der Universitat Autònoma de Barcelona und schloss das Studium 1997 ab. Von 1999 bis 2002 absolvierte sie ihren Master in Elektrotechnik an der Universitat Politècnica de Catalunya, Spanien. 2009 beendete sie ihre Dissertation zum Thema „Energy Harvesting from Human Passive Power". Seit 2007 arbeitet sie am Fraunhofer IIS, und seit 2018 ist sie Gruppenleiterin der Gruppe „Advanced Analog Circuits".
loreto.mateu@iis.fraunhofer.de