Security für Deep Learning

Management von Safety und Security bei Programmierbaren SoCs

Dr. James Hunt, Aicas GmbH, Dr. Giulio Corradi, Xilinx GmbH

Maschinelles Lernen und künstliche Intelligenz sind wichtige Trends, die eine zentrale Bedeutung als Unterstützung autonomer Entscheidungsfindung haben. Gefordert sind dafür ein sicheres, dynamisches Update von Neukonfigurationen und deren Überwachung sowie eine perfekt verwaltete und optimierte Sprachplattform. Diese bietet die notwendige Sicherheit für diese Systeme als auch eine einfache und sichere Verwendung für den Zugriff und die Aktualisierung von GPU- und FPGA-codierten Funktionen.

Industrieller Trend zum Machine Learning

Es wird erwartet, dass industrielle Anwendungen immer effizienter und intelligenter arbeiten werden. Maschinelles Lernen (ML) und künstliche Intelligenz (AI) sind Schlüsseltechnologien um diese Ziele zu erreichen. Für den industriellen Bereich übernehmen Aspekte wie Echtzeit und niedrige Latenz eine essentielle Bedeutung, und betonen weitere Besonderheiten des IIoT (Industrial Internet of Things), zumal wenn ML und AI physische Größen kontrollieren und Echtzeit und niedrige Latenz von allergrößter Bedeutung werden.

Die Edge-Plattform für IIoT

Erstellen einer Architektur einer IIoT-Plattform ist nicht trivial; vieles hängt dabei vom beabsichtigten Zweck ab. Zum Beispiel erfordert die gesamte Kette – von der Physical World bis zur Cloud – die Bewältigung multipler Technologien.
Experten definieren eine solche Plattform mit drei Hauptmerkmalen:

- eine Sammlung von Assets, hier als Kombination von Komponenten, Prozesse, Kenntnisse, Personen und Beziehungen gemeint;
- eine Sammlung technischer Elemente, insbesondere der zugrundeliegenden Kerntechnologie, die über die gesamte Produktpalette hinweg implementiert werden; und
- eine Reihe von Subsystemen und Schnittstellen, die eine gut funktionierende, homogene Struktur bilden

Da der Bedarf nach Aktualisierung ständig wächst, wird eine veränderbare Plattform empfohlen, bei der die feste Struktur trotzdem die Flexibilität anbietet, um sich bei Bedarf verschiedenen Anwendungen optimal anzupassen. Diese erfolgt mithilfe von programmierbarer Logik, durch benutzerdefinierte Elemente, die die Standardressourcen des Systems dabei unterstützen, sich optimal zu erweitern oder ergänzen. Diese Fähigkeit wird All Programmable System on Chip (APSoC) genannt.

Machine Learning (ML)

Maschinelle Lern-Modelle haben die Fähigkeit, auf einer Reihe von Eingabedaten basierend auf Erfahrung Schlussfolgerungen zu ziehen. Diese Erfahrung wird durch die Analyse einer Vielzahl von aus unterschiedlichen Systemen extrahierten Daten, die Beweise bilden, erreicht. Der Prozess, Schlussfolgerungen aus Beweismitteln zu ziehen, wird als Inferenz bezeichnet.

Das Extrahieren von Beweisen aus einer Menge von Daten erfolgt in diesen Modellen durch Trainings- Lernprozesse. Sobald das Training durch menschliche Kriterien (“supervised”) oder eine Beurteilungsfunktion (“unsupervised”) abgeschlossen ist, ist das ML-Modell bereit, den Schlussfolgerungsprozess durchzuführen, für den es trainiert wurde. Als Ergebnis kann das System neue und unbekannte Eingaben ableiten.
Viele Modelle werden heutzutage verwendet, um Folgerungen mit einem unterschiedlichen Grad an Komplexität, Genauigkeit und Präzision in ihrem Ergebnis zu treffen. Bei allen werden eine spezielle Umgebung und Ressourcen für ML erforderlich.

Die ML-Umwelt
Die zentrale Funktion in dieser ML-Umwelt übernimmt der Data Scientist. Er sucht die beste Abbildung und das Format für das vorliegende Problem. Es gibt grundsätzlich für ungelöste Probleme eine unbestimmbare Zeit für die Modellexploration, eine Aktivität, die Datenerfassung, Normalisierung und Informationsreduktion, sowie mögliche Anpassungen des maschinellen Lern-Algorithmus, Transformation, Erweiterungen und andere Datenmanipulationen umfasst. Mit einem Wort ist die Modelluntersuchung eine experimentelle Phase, in der die verborgene Essenz destilliert wird. Wenn das Modell den Zweck erfüllt, ist es bereit für die nächste Optimierung, die das Modell auf der endgültigen Computerplattform bereitstellt.

Der Prozess, um herausragende Eigenschaften zu entdecken und diese zu nutzen, um sie in das eingebettete Computerszenario einzupassen besteht aus folgenden Schritten:

- Problemidentifikation (beinhaltet das Verständnis der benötigten oder verfügbaren Informationen),
- Model Exploration (beinhaltet die Auswahl vernünftiger Algorithmen für maschinelles Lernen),
- Modellverifizierung (beinhaltet das Testen der Eignung für den ausgewählten Algorithmus)
- Modelleinbettung (beinhaltet mögliche Optimierungen für die Ausführung, den Footprint oder andere Parameter),
- Modellausführung in der Embedded-Umgebung (beinhalten, wie das maschinelle Lernen mit der Umgebung interagiert).

Problemerkennung, Modellexploration, Verifikation und Einbettung, sind die Themen, in denen der Designer am meisten betroffen ist. Leider fordert das reale Leben auch mit einem vollständigen Testsatz für die vorliegende Aufgabe, der alle Unsicherheiten umfasst, dennoch immer eine Anpassung, nachdem der ML-Algorithmus implementiert wurde. Bild 1 berücksichtigt dieses Umlernen als Feedback einer eingebetteter Ausführung.

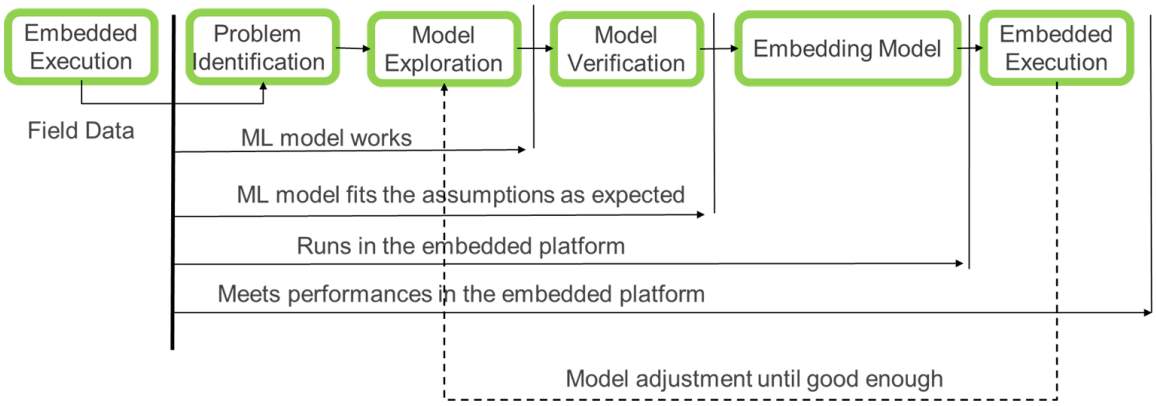

Bild 1: Umlernen als Feedback einer eingebetteten Ausführung

Realisiert werden kann die Modellbildung in leistungsfähigen FPGAs, in denen, entsprechend der Problemlösung, Modelle als Konfigurationen von FPGAs erzeugt werden, die anschließend zur Ausführung gelangen.
Daraus wird ein iteratives Vorgehen, da immer neue Konfigurationen erstellt, heruntergeladen und Inbetrieb genommen werden müssen. Jeder dieser Updates sollte schnell, sicher und automatisch erfolgen, ohne das laufende System unterbrechen zu müssen oder gar die komplette HW auszutauschen.

Sobald das ML-Modell geformt ist, ist das Schlüsselproblem Robustheit und Leistungen und für die Übertragung Sicherheit.

Bedarf an Sicherheit
Ein wie hier beschriebenes Modell benötigt eine initiale Inbetriebnahme und eine anschließend möglichst permanente Überarbeitung aufgrund neuer Informationen. Die generierten Resultate (also FPGA-Konfigurierung) müssen auf das Zielsystem aufgespielt und dort in Betrieb genommen werden. Aktualisierungen müssen unmittelbar, direkt und sicher eingespielt werden können, z.B. als Software over the Air (SOTA).

Dazu wird für das Gesamtsystem des ML ein Framework benötigt, das die Ausführung von Aktionen, die aus dem ML heraus initiiert werden, die Kommunikation mit Fernsystemen (z.B. in der Cloud) sowie die Sicherheit in der Übertragung ermöglicht bzw. sicherstellt.

Dieses Framework muss bidirektional arbeiten können, d.h. Performance-Informationen aus dem entsprechenden Device auslesen und sicher unmanipuliert herausübertragen wie auch sichere Updates des ML-Modells (downloads) erlauben. Applikationen wie das ML-Modell oder FPGAs für die geforderte schnelle Signalverarbeitung und flexible Änderung der Schaltung, um z.B. nachträgliche Verbesserungen an den implementierten Funktionen vornehmen zu können, ohne dabei direkt die Hardware ändern zu müssen, werden im beschriebenen Framework ebenso wie Aktualisierungen in einem Portal remote zum Download zur Verfügung gestellt. Von dort können Sie initial inbetrieb genommen werden sowie können notwendige Updates automatisch auf das Zielsystem gesandt werden.

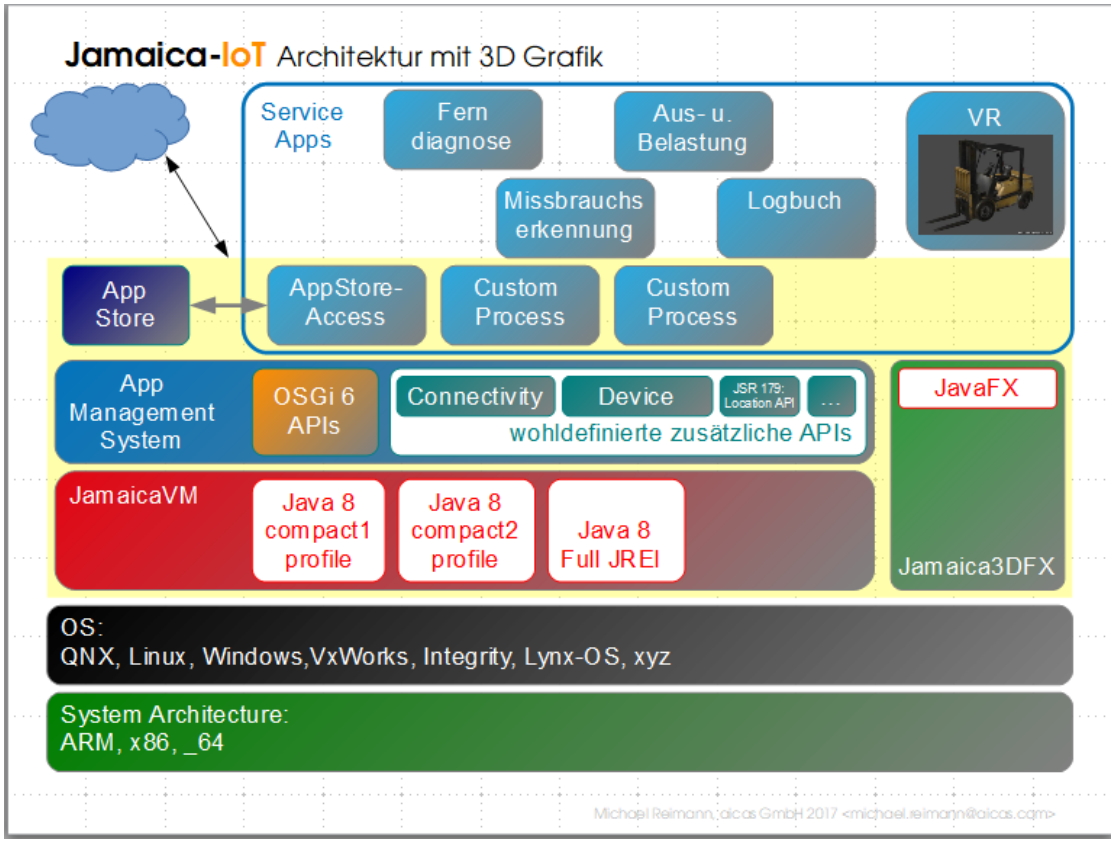

Bild 2: Framework zur sicheren Aktualisierung, Ausführung und Kommunikation

Die Security- Vertrauenskette

Eine nahtlose Kette des Vertrauens wird von der Erstellung eines ML- Modells oder eines FPGAs und deren Updates bis zur sicheren Ausführung auf dem Device benötigt, hier folgt eine auswahlhafte Beschreibung solcher Sicherheitsmechanismen. Ein Trusted Platform Module (TPM) bietet kryptografische Funktionen in einem Computer, die eine effektive Ergänzung zu seinen Sicherheitsfunktionen (z.B. für das Zynq®-7000 APSoC) darstellt

.

Die sichere Startfunktion bietet die Fähigkeit, alle Partitionen zu authentifizieren, die beim Booten geladen wurden. Es unterstützt auch die AES-Verschlüsselung (Advanced Encryption Standard) von Partitionen, die vertraulich behandelt werden müssen.

Der Secure Framework Loader (SFL) ist eine im Start-Image enthaltene Binärdatei, durch die das sichere Booten bestätigt wird. Aufgabe dieses Ladevorgangs ist, das Jamaica-IoT-Framework zu authentifizieren und starten. Da der SFL durch das sichere Booten authentifiziert wird, wird ihm vertraut. Wenn der SFL das Framework authentifiziert und lädt, überträgt er deshalb das Vertrauen auf das Framework.

Der SFL überträgt das Vertrauen auf das Jamaica-Iot-Framework, das wiederum das entsprechende Root-Zertifikat enthält. Mit diesem werden alle Konfigurationsressourcen authentifiziert und damit wird das Vertrauen auf alle Konfigurationsdaten übertragen. Wenn die Authentifizierung der

Konfigurationsressourcen fehlschlägt, wird das Framework gestoppt und ein Fehler protokolliert.
Nach Verifikation der Signatur wurde das Vertrauen auf die Konfigurationsdaten übertragen. Die Konfigurationsdaten enthalten das OEM Certification Authority (CA) Root-Zertifikat. Dieses ermöglicht es dem OEM, zu steuern, welche Ressourcen und welche Anwendungen auf dem Gerät installiert und ausgeführt werden dürfen.
Alle Anwendungen, Komponenten und Ressourcen müssen zur Installationszeit signaturauthentifiziert werden, und zwar durch ein Zertifikat, das direkt oder indirekt mit dem OEM CA Root-Zertifikat verkettet ist.

Die Installation kann über einen Over-the-Air-Download oder ein lokales Speichermedium erfolgen. Es wird keine Anwendung, Komponente oder Ressource installiert, wenn die Authentifizierung ihrer Signaturen fehlschlägt oder wenn die Authentifizierung ihrer zugehörigen Zertifikate fehlschlägt.

Ressourcenmanagement
Der Betreiber des Jamaica-IoT Frameworks oder eine von ihm autorisierte Instanz vergibt Definitionen für eine Begrenzung der Ressourcennutzung, z.B. Größe des RAM oder Anzahl erlaubter Threads. Diese werden während der Ausführung des Frameworks bzw. einer Applikation darin angewandt. So kann verhindert werden, dass geladene Programme oder auch nur Programmkomponenten zum Update sich nach dem Installieren durch übermäßige Ausbreitung des Devices bemächtigen.

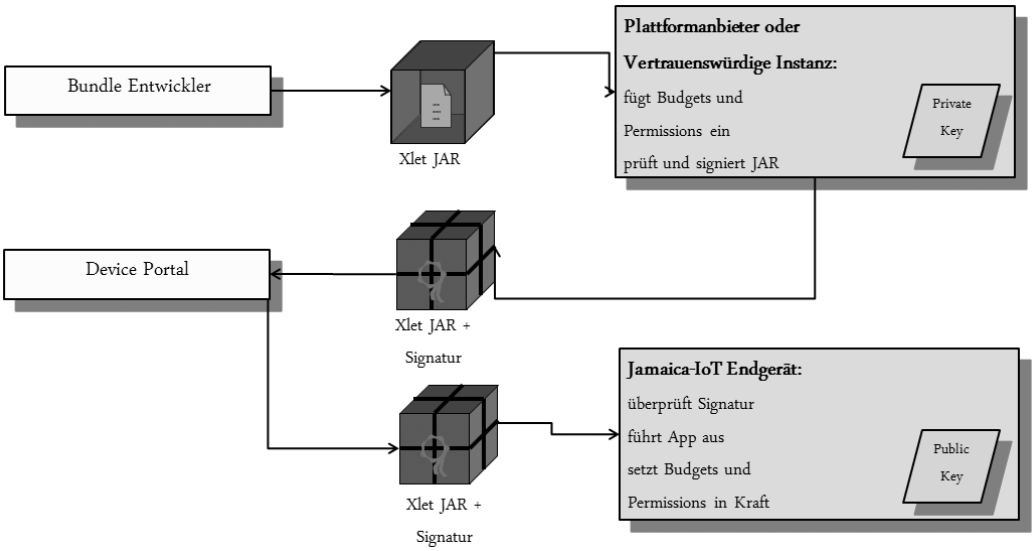

Bild 3: Security Vertrauenskette in Jamaica-IoT

Fazit:
Machine Learning ist eine angebrachte Technologie für eingebettete Systeme, aber um das volle Potential ausnutzen zu können braucht man mehr als ein neurales Netzwerk. Erst durch Anwendung weiterer Technologien wie FPGA für eine effiziente Ausführung des Lernens bis hin zu einem Framework für das sichere Herunterladen, Installieren und Ausführen dynamischen Codes können die großen Fähigkeiten des Machine Learning in eingebetteten Systemen verbreitet werden.

Autoren

Dr. James J. Hunt ist Mitbegründer und GF der Firma aicas. Er hat einen B.Sc.Yale und einen Master von der Boston Univ., sowie an der Universität Karlsruhe promoviert. Er hat diverse Erfahrungen in der Wafer-Scale-Integration, parallelen Systemen für Signalverarbeitung und formalen Methoden. Er hat außerdem den Avionik-Standard ED217 (DO-332) mitgestaltet. Aktuell leitet er die Expertengruppe für echtzeitfähiges Java (JSR-282).

Co-Referent Dr. Giulio Corradi ist Sr. Systemarchitekt und bringt 25 Jahre Erfahrung in den Bereichen Management, Software Engineering Embedded Systems und Entwicklung von ASICs und FPGAs mit. Machine Learning, Echtzeitkommunikation und funktionale Sicherheit stehen bei ihm im Mittelpunkt. Seit 2006 arbeitet Giulio bei Xilinx in München und leistete einen wichtigen Beitrag bei der Xilinx Functional Safety Zertifizierung von Tools und Compilern.