Der KI-gesteuerte Workflow

Ingenieur*innen, die Machine Learning und Deep Learning einsetzen, erwarten oft, dass sie einen großen Teil ihrer Zeit mit der Entwicklung und Feinabstimmung von KI-Modellen verbringen. Die Modellierung ist zwar ein wichtiger Schritt im Workflow, aber das Modell ist nicht alleiniges Ziel. Der Schlüssel zum Erfolg bei der praktischen KI-Implementierung ist das frühzeitige Aufdecken von Problemen.

Vier Schrittelassen sich in einem KI-gesteuerten Workflow unterscheiden

SCHRITT 1: DATENAUFBEREITUNG

Die Datenaufbereitung ist wohl der wichtigste Schritt im KI-Workflow: Ohne robuste und genaue Daten zum Trainieren eines Modells sind Projekte rasch zum Scheitern verurteilt. Wenn Ingenieur*innen das Modell mit „schlechten" Daten füttern, werden sie keine aufschlussreichen Ergebnisse erhalten – und wahrscheinlich viele Stunden damit verbringen, herauszufinden, warum das Modell nicht funktioniert.

Um ein Modell zu trainieren, sollten Ingenieur*innen mit sauberen, gelabelten Daten beginnen, und zwar mit so vielen wie möglich. Dies kann einer der zeitaufwendigsten Schritte des Workflows sein. Wenn Deep Learning-Modelle nicht wie erwartet funktionieren, konzentrieren sich viele darauf, wie man das Modell verbessern kann – durch das Optimieren von Parametern, die Feinabstimmung des Modells und mehrere Trainingsiterationen. Doch noch viel wichtiger ist die Aufbereitung und das korrekte Labeln der Eingabedaten. Das darf nicht vernachlässigt werden, um sicherzustellen, dass Daten korrekt vom Modell verstanden werden können.

SCHRITT 2: KI-MODELLIERUNG

Sobald die Daten sauber und richtig gelabelt sind, kann zur Modellierungsphase des Workflows übergegangen werden. Hierbei werden die Daten als Input verwendet und das Modell lernt aus diesen Daten. Das Ziel einer erfolgreichen Modellierungsphase ist die Erstellung eines robusten, genauen Modells, das intelligente Entscheidungen auf Basis der Daten treffen kann. Dies ist auch der Punkt, an dem Deep Learning, Machine Learning oder eine Kombination davon in den Arbeitsablauf einfließt. Hier entscheiden die Ingenieur*innen, welche Methoden das präziseste und robusteste Ergebnis hervorbringt.

Die KI-Modellierung ist ein iterativer Schritt innerhalb des gesamten Workflows, und Ingenieur*innen müssen die Änderungen, die sie während dieses Schritts am Modell vornehmen, nachverfolgen können. Die Nachverfolgung von Änderungen und die Aufzeichnung von Trainingsiterationen mit Tools wie dem Experiment Manager von MathWorks sind entscheidend, da sie helfen die Parameter zu erklären, die zum genauesten Modell führen und reproduzierbare Ergebnisse liefern.

SCHRITT 3: SIMULATION UND TESTS

Ingenieur*innen müssen beachten, dass KI-Elemente meistens nur ein kleiner Teil eines größeren Systems sind. Sie müssen in allen Szenarien im Zusammenspiel mit anderen Teilen des Endprodukts korrekt funktionieren, einschließlich anderer Sensoren und Algorithmen wie Steuerung, Signalverarbeitung und Sensorfusion. Ein Beispiel ist hier ein Szenario für automatisiertes Fahren: Dabei handelt es sich nicht nur um ein System zur Erkennung von Objekten (Fußgänger*innen, Autos, Stoppschilder), sondern dieses System muss mit anderen Systemen zur Lokalisierung, Wegplanung, Steuerung und weiteren integriert werden. Simulationen und Genauigkeitstests sind der Schlüssel, um sicherzustellen, dass das KI-Modell richtig funktioniert und alles gut mit anderen Systemen harmoniert, bevor ein Modell in der realen Welt eingesetzt wird.

Um diesen Grad an Genauigkeit und Robustheit vor dem Einsatz zu erreichen, müssen Ingenieur*innen validieren, dass das Modell in jeder Situation so reagiert, wie es soll. Sie sollten sich auch mit den Fragen befassen, wie exakt das Modell insgesamt ist und ob alle Randfälle abgedeckt sind. Durch den Einsatz von Werkzeugen wie Simulink können Ingenieur*innen überprüfen, ob das Modell für alle erwarteten Anwendungsfälle wie gewünscht funktioniert, und so kosten- und zeitintensive Überarbeitungen vermeiden.

SCHRITT 4: EINSATZ

Ist das Modell reif für die Bereitstellung, folgt als nächster Schritt der Einsatz auf der Zielhardware – mit anderen Worten, die Bereitstellung des Modells in der endgültigen Sprache, in der es implementiert werden soll. Das erfordert in der Regel, dass die Entwicklungsingenieur*innen ein implementierungsbereites Modell nutzen, um es in die vorgesehene Hardwareumgebung einzupassen.

Die vorgesehene Hardwareumgebung kann vom Desktop über die Cloud bis hin zu FPGAs reichen. Mithilfe von flexiblen Werkzeugen wie MATLAB kann der endgültige Code für alle Szenarien generiert werden. Das bietet Ingenieur*innen den Spielraum, ihr Modell in einer Vielzahl von Umgebungen einzusetzen, ohne den ursprünglichen Code neu schreiben zu müssen. Das Deployment eines Modells direkt auf einer GPU kann hier als Beispiel dienen: Die automatische Codegenerierung eliminiert Codierungsfehler, die durch eine manuelle Übersetzung entstehen könnten, und liefert hochoptimierten CUDA-Code, der effizient auf der GPU läuft. ■

MathWorks
www.mathworks.com