

Beseitigt Künstliche Intelligenz die Notwendigkeit Programmieren in Schulen zu unterrichten?

Um 2010 herum, als Douglas Rushkoff die Gesellschaft mit dem Buch «Program or be Programmed» provozierte, begann sich die ganze Welt zu fragen, ob und wie Programmieren als neue Kompetenz in Schulen konzipiert werden sollte. In der Schweiz fand Programmieren durch «Medien und Informatik» Kompetenzen Eingang in den Lehrplan 21.

Inzwischen hat die Künstliche Intelligenz (KI) begonnen, ihr volles Potenzial zu entfalten. Neue Ansätze wie Deep Learning, kombiniert mit enorm grossen Datensätzen und angetrieben von leistungsfähigerer Hardware, begannen sich auszuzahlen. Sie übertrafen den Menschen nicht nur in mechanischen Fähigkeiten wie mathematischen Berechnungen, sondern beispielsweise auch im Schachspiel. 1996 gewann IBMs «Deep Blue» im Schach gegen Grossmeister Gary Kasparov. Man nahm damals jedoch an, dass ähnliche Leistungen für Spiele mit grösseren Entscheidungsspielräumen niemals erreicht werden könnten. Doch 2016 besiegte Googles «AlphaGo» den weltbesten Go-Spieler. Die Erfolge von KI wurden nicht nur weiter vorangetrieben, sondern sogar beschleunigt. Mit der Veröffentlichung von GitHub «CoPilot» begann Microsoft, einen neuen Massstab für KI zu setzen. Mit «CoPilot» konnten Benutzer vollständige Programme mit nur wenigen Kommentaren erstellen, die vorschlugen, was der Code tun sollte.

«Explicative Programming» hat die Interdisziplinarität im Blick

Sollten wir angesichts der erstaunlichen Entwicklung der KI überhaupt in Betracht ziehen, in Schulen zu programmieren? Sind die ursprünglichen Begründungen für die Einführung der Informatik im Allgemeinen und der Programmierung im Besonderen noch haltbar? Es gibt zwei Denkrichtungen. Eine davon ist eine berufsorientierte Denkweise, in der es dringend erforderlich ist, eine Pipeline für die Informatikausbildung zu füttern. In der zweiten Denkrichtung, dem «Explicative Programming», geht es darum, die Welt durch das Schreiben von Programmen zu verstehen. Die Essenz des Explikativen Programmierens ist «Computational Thinking» (Denken mit dem Computer). Wichtig ist, dass sich diese erklärende Programmierung nicht auf das Verständnis der Welt der Informationstechnologie konzentriert, sondern auf die Welt im Allgemeinen.

Kann es sein, dass man durch Programmieren Mathematik, Musik, Kunst, Geometrie und sogar Sprachen besser verstehen könnte? Interessanterweise war Seymour Papert, der durch die Programmiersprache Logo als Vater der pädagogischen Programmierung gilt, kein Befürworter davon, Programmieren um des Programmierens willen zu lehren. Er unterschied ausdrücklich «Lernen zu Programmieren» von «Programmieren zu lernen».

Damit KI Sinn macht, muss die Gesellschaft das Rushkoff-Ultimatum ablehnen. Es geht nicht um «programmieren oder programmiert werden», sondern darum, einen Weg der Konvergenz zu finden, der menschliche Fähigkeiten mit den Möglichkeiten der Computer verbindet. Der Schlüssel dazu liegt darin, dass die Summe menschlicher Fähigkeiten und Computerleistungen mehr ist als die KI allein. Gary Kasparov etwa fand neue Strategien, um Schach zu spielen, indem er wiederholt gegen «Deep Blue» antrat. Menschen, die KI als computergestützte Denkwerkzeuge einsetzen, können die Dinge relativieren und den gesunden Menschenverstand anwenden. Die Konvergenz mag ein langsamer und sich ständig ändernder Prozess sein, aber sie könnte sich als der vielversprechendste Weg erweisen, um effektive Mittel zu finden, um menschliche Fähigkeiten und Computerleistungen miteinander gewinnbringend zu verbinden.

Programmieren in unterschiedlichen Kontexten einsetzen

Die Herausforderung für den Unterricht wird darin bestehen, interdisziplinäre Grenzen zu überschreiten, um Konvergenz zu finden. Während Bildungseinrichtungen theoretisch Interdisziplinarität annehmen wollen, tun sie sich schwer damit, innovative Lehrpraktiken zu entwickeln, die der Interdisziplinarität dienen. Schulen waren bis ins 21. Jahrhundert hinein fest unterteilt in Disziplinen wie Sprache, Mathematik und Naturwissenschaften. Selbst auf der Primarschulebene, wo die Lehrer nicht so stark spezialisiert sind, begünstigten Kursstrukturen und pädagogische Modelle keine Interdisziplinarität. Im Gegenteil, jüngere Disziplinen wie die Informatik wurden oft als Konkurrenz zu etablierten Disziplinen betrachtet. Die Ansicht, dass Interdisziplinarität eine fortgeschrittene Bildungspraxis ist, die zunächst ein ausreichendes individuelles disziplinäres Verständnis erfordert, bevor Verbindungen zwischen den Disziplinen hergestellt werden können, ist tief in der Lern- und Lehrpraxis verwurzelt.

Papert sah das ganz anders, als er das Machtprinzip („power principle“) vorschlug. Genau wie bei Sprachen, argumentierte er, sollten wir zuerst versuchen, etwas zu verwenden (wie etwa kleine Kinder, die versuchen, mit Worten mitzuteilen, was sie gerne essen), bevor wir es verstehen. In ähnlicher Weise schlug er vor, das Programmieren zu nutzen, um Mathematik zu verstehen.

Beseitigt KI also die Notwendigkeit, Programmieren in Schulen zu unterrichten? Wenn wir das Programmieren um seiner selbst willen lehren oder es «nur» als Instrument nutzen, um sich der Welt der Informationstechnologie zu nähern, dann vielleicht. Wir verstehen

nicht, wie selbstfahrende Autos funktionieren, nur weil wir einige Programmiergrundlagen gelernt haben. Mit «Explicative Programming» wird Programmieren jedoch zu einem konvergierenden Prozess, der menschliche Fähigkeiten und Computerleistungen mit dem gemeinsamen Ziel verbindet, komplexe Systeme meist ausserhalb, aber auch innerhalb des Bereichs der Technologie zu erklären. Erstellen Sie deshalb Simulationen, um die Mechanismen einer Ausbreitung des Covid-19-Virus in der Biologie zu erklären. Programmieren Sie Arcade-Spiele wie Pac-Man, um fesselnde Artefakte zu bauen, aber auch um schwierige mathematische Konzepte wie Wahrscheinlichkeit zu erklären. Entwickeln Sie ausführbare musikalische Darstellungen wie Rhythmusmaschinen, um wichtige Prinzipien wie Interpretation, Hierarchie und Zufall zu erklären, die Musik und Berechnung gemeinsam haben oder codieren Sie Meisterwerke der modernen Kunst wie Gemälde von Mondrian neu, um künstlerischen Ausdruck, rechnerisches Denken und Farbtheorie zu erklären.

Präsentation

- Umreisst wichtige Meilensteine relevanter KI-Entwicklungen.
- Veranschaulicht die Kompromisse zwischen verschiedenen Begründungen für das Unterrichten von Informatik an öffentlichen Schulen und diskutiert die Auswirkungen von KI auf diese Begründungen.
- Präsentiert den Begriff des “Explicative Programming”
- Beschreibt kurz die Wirksamkeit der Informatik-Unterrichtsstrategie, die an der PH FHNW mit über 2000 Grundschullehrern angewendet und getestet wurde.
- Schlägt Strategien für andere PHs und FHs vor, um Informatikkurse zu entwickeln, die menschliche Fähigkeiten mit Computerleistungen (insbesondere KI) synergieren.
- Diskutiert mit den Teilnehmern die wahrscheinlichen Auswirkungen von KI auf die Informatikausbildung an PHs und FHs.

Präsentator

Prof. Dr. Alexander Repenning war der erste Hasler-Professor in der Schweiz. Er entwickelte und empirisch evaluierte die Scalable Game Design-Strategie, die Computational Thinking durch Gamedesign-Aktivitäten in den USA, Mexiko, Brasilien und jetzt der Schweiz lehrt. Er ist ein Pionier der Computational Thinking Tools und der blockbasierten Programmierung. Repenning ist Inhaber des Lehrstuhls für Informatikdidaktik an der PH FHNW. Repenning ist einer der 100 Digital Shapers (Bilanz, Schweiz), ein Berater der US National Academy of Sciences, der European Commission, der Japanese Ministry of Education, der Organization for Economic Co-operation and Development (OECD), dem US White House Office of Science and Technology Policy (OSTP) und der US National Science Foundation.

Referenzen

Repenning, A., & Basawapatna, A. (2021). Explizite Programmierung. *Mitteilungen der ACM*, 64(11), 30-33.