HOCHSCHULE DER MEDIEN

WHITEPAPER

KI & CODING

Werden künstliche Intelligenzen Programmierer:innen ersetzen?

Aktuelle Themen, SoSe 2022

Jana Cuntz | UK3 jc038@hdm-stuttgart.de Mart.Nr.: 42972

Werden künstliche Intelligenzen Programmierer:innen ersetzen?

Die Digitalisierung schreitet voran und das Internet of Things wächst und wächst. Noch vor 20 Jahren gab es kaum mobile Geräte die Internet fähig waren. Nun gibt es Apps für Waschmaschinen und Mikrowellen und selbst Elektroautos bewegen sich ohne Software kein Stück vom Fleck. In fast jedem Produkt steckt ein bisschen Code. Und wenn nicht dort, dann in den Maschinen, die es produzieren oder im Logistikzentrum, das dafür sorgt, dass das Produkt zu uns kommt. Und egal ob große App oder kleine API, Software braucht Menschen, die diese Schreiben können. Mit zunehmendem Bedarf an Software steigt somit auch der Bedarf an Softwareentwicklern und Personen mit diesen Qualifikationen. Doch gerade in der IT-Branche ist der in Deutschland herrschende Fachkräftemangel extrem zu spüren (Specht, 2022).

Die Branche versucht auf verschiedene Weisen diesem Mangel entgegenzuwirken. Eine Option sind No-Code oder Low-Code Programme. Sie ermöglichen es schnell, ohne viel Know-how Anwendungen zu programmieren, ohne selbst Code zu schreiben. Daher werden auf Seite der Anwender:innen weniger Kenntnisse benötigt. Doch auch diese Programme, Tools und Oberflächen brauchen Programmierer:innen und Softwareingenieur:innen, die diese wiederum programmieren, reparieren und ständig weiterentwickeln. Deshalb erleichtern diese Anwendungen gerade den Nutzer:innen die Arbeit, aber ein:e Programmierer:inn kann dadurch nicht ersetzt werden.

Code folgt immer gewissen Mustern. Es gibt eine intrinsische Logik, bestimmte Vokabeln und eine festgelegte Grammatik. Vergleichbar mit einer semantischen Sprache. Diese Programmiersprachen unterscheiden sich untereinander. Aber im besten Fall kann ein Stück Code von jeder Person gelesen werden, die sie Sprache des Codes beherrscht.
Wenn es klare Muster und Regeln gibt, liegt es daher nicht fern, Deep Learning und künstliche Intelligenzen (im Folgenden KI oder AI abgekürzt) anzuwenden, um das Leben der Programmierer:innen zu vereinfachen – oder um sie komplett zu ersetzten?


[1]

NO CODE & LOW CODE
Diese Art von Programmieren funktioniert, wie der Name vermuten lässt, auf wenig bis gar keinem Code. Der User kann meist mithilfe von visuellen Drag-and-drop-Elementen, Text-Befehle oder Zeichnungen gekoppelt durch Bilderkennung Apps, Website oder Plattformen gestalten. Im Hintergrund wird dann der entsprechende Code generiert. Bei Low-Code Anwendungen kann der Code oft noch eingesehen und manuell angepasst werden. Bei No-Code Anwendungen ist dies meist nicht mehr möglich. Es gibt mittlerweile auch einige No- und Low-Code Anwendungen, die durch KI unterstützt werden, um die Ergebnisse und die Handhabung zu vereinfachen. Beispiele für bekannte No- und Low-Code Anwendungen sind z. B. Wordpress, Google Analytics und Microsoft PowerApps (02100 Digital, 2020).

Übersetzen

Jede:r kennt Google Übersetzer, Deep L oder andere Übersetzungstool. Hier werden Algorithmen und neuronale Netze genutzt, um Wörter, Sätze und sogar ganze Texte von einer in eine andere Sprache zu übersetzten. Nach demselben Prinzip gibt es mittlerweile auch einige Code-Übersetzer.

[2]

Manchmal ist es notwendig, bestehenden Code in eine andere Sprache zu übersetzen. Entweder weil der Code in der die Software geschrieben wurde veraltet ist, oder weil sich die Anforderungen über die Entwicklung eines Produktes ändern und dann eine andere Programmiersprache doch die bessere gewesen wäre. Es kommt auch vor, dass man zwei Programme in verschiedenen Sprachen zu einer zusammenfügen möchte. Dies ist nicht ganz so einfach wie semantisches übersetzen, da meist auch die Softwarearchitektur angepasst werden muss. Auf dem Markt gibt es Programme, die das zumindest bedingt können, sogenannte Transcomplier. Trotzdem braucht es hier noch die massive Unterstützung von Softwareentwickler:innen. Außerdem braucht es Personen, die mit zwei Sprachen sehr gut auskennen (Computer Hope, 2017).

TransCoder

Meta AI, das Forschungsinstitut des amerikanischen Großkonzerns Meta, veröffentlicht im Juli 2020 eine Neuheit. Sie entwickelten einen Open Source TransCoder der mithilfe von Deep Leraning Mechanismen dazu in der Lage ist Code zwischen den Programmiersprachen C++, JAVA und Python 3 zu übersetzen. Auf der Seite können einfach wie bei den gängigen semantischen Übersetzern Code eingegeben oder eingefügt werden und in wenigen Sekunden wird der Code übersetzt. Normalerweise bräuchte man für das Trainieren einer solchen KI sehr viele Teile desselben Codes, der in zwei Sprachen geschrieben wurde. Dies ist bei TransCode nicht der Fall. Die KI lernt durch Code der einzelnen Sprachen und kann so viel schneller trainiert werde.

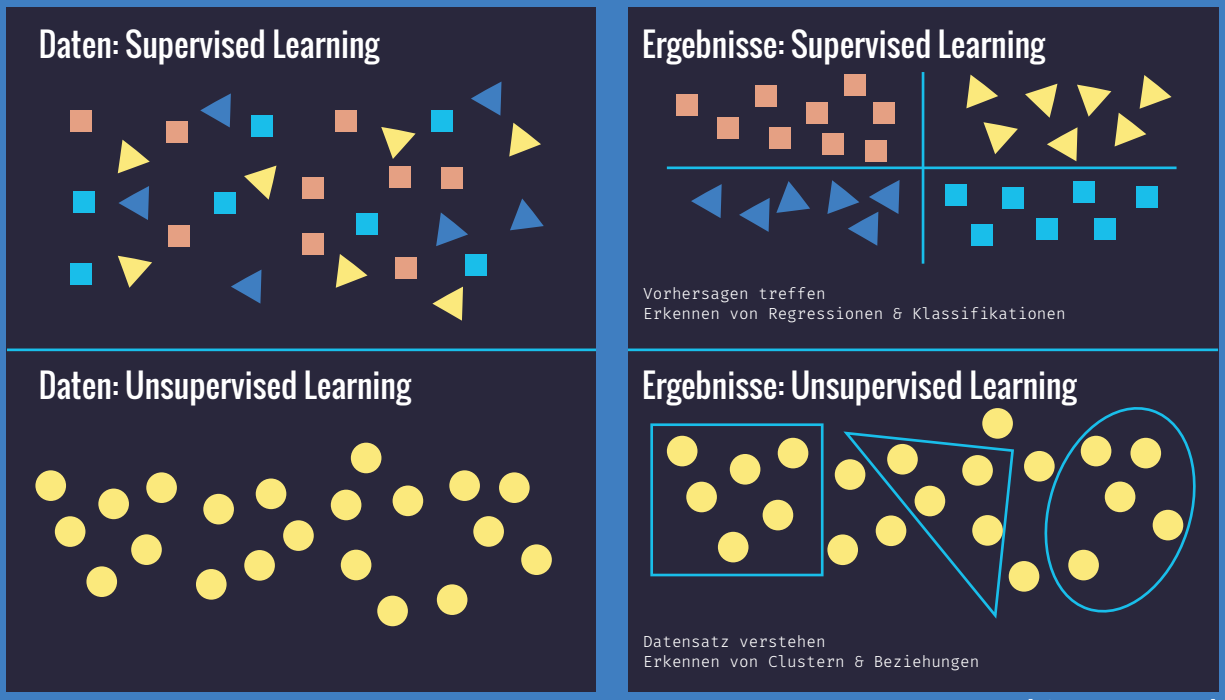
Wie das funktioniert:
Ganz vereinfacht gesagt, bringt sich die KI selbst die Programmiersprache bei. Die Trainingsdaten werden nach Mustern durchsucht und dann geclustert. Z. B. Welche Befehle was tun. Das macht die KI dann mit allen drei Sprachen. Wird ein Code zum Übersetzten eingegeben, analysiert die KI, was der Code macht und findet dazu die passenden Funktionen in den anderen Sprachen. Dabei wird auf sogenanntes unsupervised machine learning zurückgegriffen (Meta, 2020).

Code zu übersetzen, wird in der Praxis recht wenig gebraucht. Praktisch ist es vor allem für Anfänger:innen oder Personen, die eine neue Programmiersprache lernen wollen. Sie können den Code der bekannten Sprache übersetzen und sehen so, wie dasselbe Problem in einer anderen Sprache gelöst wird.
Der TransCoder ist Open Source und kann kostenlos von jedem User genutzt werden. Somit werden Unmengen an Trainingsdaten generiert. Daher besteht die Möglichkeit, dass Meta unter anderem diese Daten nutzt, um ihre eigene KI zu trainieren, damit sie später dann evntl. Selbst Code schreiben kann.

SUPERVISED & UNSUPERVISED MACHINE LEARNING

Es gibt mehrere Möglichkeiten, eine KI zu trainieren. Zwei davon sind supervised und unsupervised machine leraning. Der Name ist etwas irreführend. In beiden Fällen wird die KI nicht die ganze Zeit während des Lernprozesses beobachtet. Aber es gibt einen Unterschied in den Trainingsdaten und einen Unterschied welche Arten von Ergebnissen so erzielt werden können. Beim supervised learning bekommt die KI gelabelte Trainingsdaten. Diese haben, wie der Name sagt, Labels.

Das bedeutet, die KI hat zu Beginn schon Informationen. Wie in dem Bild links weiß die KI z. B. dass es sich um Dreiecke oder Quadrate handelt und, dass die unterschiedlichen Formen auch noch unterschiedliche Farben haben. Beim unsupervised learning hat die KI keinerlei Anhaltspunkt. Sie erhält einfach einen oder mehrere Datensätze. Doch nicht nur die Datengrundlage unterscheiden sich. Es gibt auch Unterschiede, welches Ziel damit verfolgt wird, und was die KI dann schlussendlich mit dem Datensatz erreicht (Cognitive Class, 2017).



[Eigene Darstellung]

Vervollständigen

Bleiben wir bei der Analogie zu semantischen Sprachen. Fast jedes Smartphoen und Tablet hat mittlerweile eine Autokorrektur oder Autovervollständigungsoptionen. Dies gibt es auch für Code. Entweder sind es Programme, die installiert werden müssen oder Integrationen, Plug-ins oder Ad-ons für bestehende Programmiersoftware und Plattformen. Es gibt kostenpflichtige und kostenfreie open source Lösungen sowie Unterstüttzungen für Einzel-Personen oder sogar ganze Teams. Manche dieser Programme basieren auf einer KI. Diese lernt mit dem Projekt mit. Zu Beginn sind die Vorschläge weniger hilfreich aber mit der Zeit lernt sie die Sprache und Anforderungen und ist somit eine gute Unterstützung. Bei Lösungen für mehrere Teammitglieder gibt es sogar die Möglichkeit, dass eine KI durch den Code von allen lernt. Das steigert die Effizienz der Teams. Je nach Programm gibt es verschiedene Feautures. Von einfacher Autokorrektur, über Vervollständigung bis hin zu Vorschlägen wie der code cleaner und effizienter gestaltet werden kann.

Die Trainingsdaten für die KIs werden aus den Projekten gezogen. Aber auch aus diversen Foren und Online-Plattformen auf denen Code offen und für jeden zugänglich zur Verfügung steht.

GitHub
In Softwareprogrammen ist Versionskontrolle extrem wichtig. Version 4 ist vielleicht gut gelaufen, dann wurde eine Kleinigkeit verändert und plötzlich geht nichts mehr. Daher ist es gut die alte Version noch zu haben um sie wiederherstellen zu können. Auch beim Zusammenarbeiten macht es Sinn den Code irgendwo abgesichert zuhaben. Jedes Mal wenn jemand aus dem Team eine neue Version hochlädt kann diese Person noch Kommentare hinzufügen, was sich genau geändert, hat. So kann das zusammenarbeiten vereinfacht werden. Eine der größten Plattformen, die dies ermöglicht, ist GitHub. Sie ist Kostenlos und gehört mittlerweile zu Microsoft. Auf dieser Plattform liegen somit Millionen von Softwareprojekten, die sich perfekt als Trainingsdaten eignen. (Wikipedia, 2022)
GitHub hat auch ein eigenes Tool zur Autovervollständigung und Korrektur von Code. Es nennt sich GitHub Copilot. Die KI, die hinter diesem Programm steht, ist dieselbe, die Open AI Codex benutzt. (OpenAI Codex, 2021)

[3]

Clean Coding
Beim Programmieren von Code ist es wichtig, dass dieser Clean ist. Die beinhaltet viele Unterpunkte. Die wichtigsten sind, dass ein code verständlich sein soll, gut veränderbar, und effizient.
Verständlichkeit ist wichtig, wenn mehrere Menschen an einem Code arbeiten. Es ist wichtig Kommentare zu machen, dass jeder genau weiß, welche Aufgabe dieser Teil des Codes erfüllt.
Technik verändert sich immer weiter. Ein guter Code ist darauf vorbereitet. Er ist so programmiert, dass er in Zukunft einfach erweitert oder verändert werden kann, ohne komplett neu geschrieben werden zu müssen.
Es gibt bekanntlich viele Wege, die nach Rom führen. So ist das auch beim Programmieren. Cleaner Code ist so kurz wie möglich, aber so lang wie nötig. Es sind keine Umwege oder andere Umgehungen enthalten, die dann dazu führen, dass der Code länger läuft und damit ineffizienter wird. (Martin, 2009)

Code generieren

Die Königsdisziplin im Bereich KI und Coding ist allerdings eine KI, die selbst Code schreiben kann. Dies wäre eine große Hilfe für Softwareentwickler und eine Lösung für das Problem des Fachkräftemangels in diesem Bereich.
Bis zum heutigen Zeitpunkt gibt es mehrere KIs, die dies schon in Teilen erreicht haben.

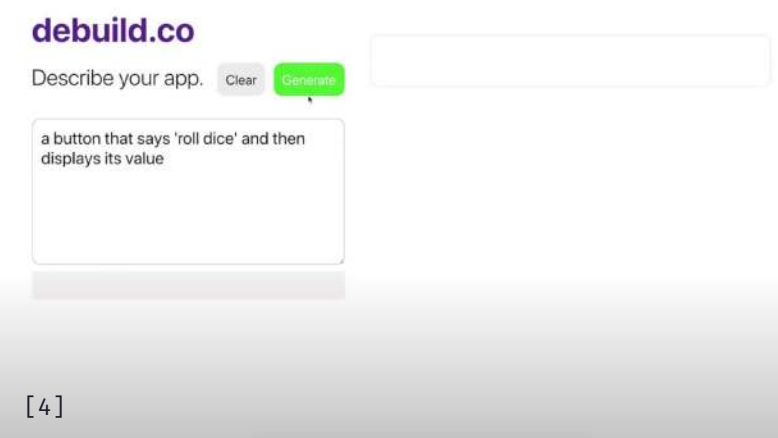
Eine KI wandelt semantische Befehle in Code um. So können Apps, Web-Applikationen und Websites gebaut und designt werden. Zwei andere lösen Probleme ähnlich zu Coding-Competitions. Alle KIs lernen mit frei zugänglichen Trainingsdaten aus dem Internet. Sie lernen Programmiersprachen und sind in der Lage aus semantische Sprachen Befehle und Aufgaben zu entnehmen.

Open AI Codex
Das Unternehmen Open AI hat sich komplett dem Erforschen von KI verschrieben. Mit Open AI Codex haben sie eine KI erschaffen, die Text Input in Code übersetzen kann. Besonders gut ist sie mit Python, aber sie kann auch andere Programmiersprachen wie JavaScript, Go und TypeScript. Außerdem kann sie auch mit Shell-Befehlen umgehen.

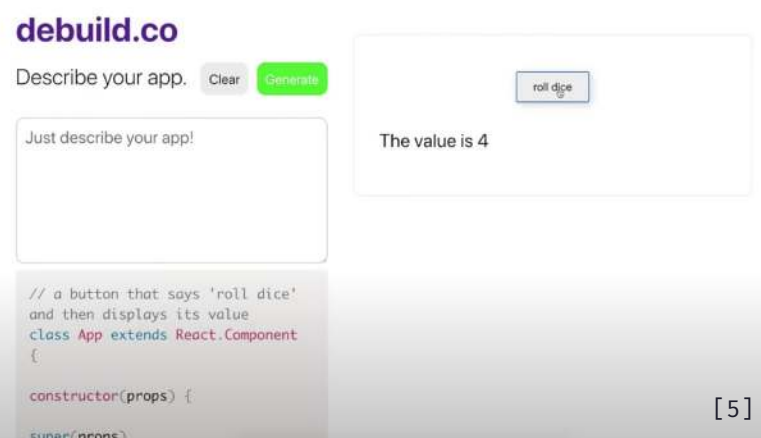
Es gibt mehrere Möglichkeiten, wie diese genutzt werden kann. Zum einen kann die KI als Lexikon benutzt werden. Über semantischen Text kann eine Frage zu einer Funktion gestellt werden und Codex gibt gesuchten Code-Schnipsel aus.

Zudem kann die KI in Kombination mit anderen Programmen beim Designen von Websites und Apps helfen. Und das durch einfache Text-Befehle wie z. B. „Header mit Mock-up, blauer fetter Call-to-Action-Button auf dem ‚Abonnieren' steht". Die KI baut dann genau das nach. Dabei macht sie das nicht nur auf einer Pixel-Ebene, sondern schreibt dazu den passenden Code. Die KI kann auch den Stil andere Seiten kopieren. „Baue mir eine Seite, die aussieht wie XY, aber als Online-Shop" ist ein Befehl, der ebenfalls akzeptiert wird. So können innerhalb von Minuten ganze Websites gebaut werden, und das ohne jegliche Vorkenntnisse.

Eine dritte Möglichkeit ist nicht nur statische Seiten zu bauen, sondern (Web-)Apps. Ein Text-Befehl kann lauten: „Ein Button, auf dem ‚Werfe einen Würfel' steht und dann eine Zahl anzeigt". Die KI versteht, was es bedeutet einen Würfel zu werfen und weiß, dass dieser die Werte zwischen 1 und 6 hat. Dann wird der dazugehörige Code programmiert (siehe Abbildung 4 und 5). (Disruption Theory, 2020). Auf diese Weise wurde sogar ein kleines Web-basiertes Computer-Spiel entwickelt (OpenAI Codex, 2021).



[4]



[5]

Momentan befindet sich OpenAI Codex noch in der Beta-Phase, aber die Entwicklungen sind vielversprechend. Die KI ist jetzt schon sehr vielseitig einsetzbar und es kann erwartet werden, dass noch mehr folgt. Sie ist aber jetzt schon eine große Hilfe, vor allem für Webentwickler und Webdesigner.

Die KI von Open AI wandelt Text und Befehle, die von Menschen gegeben werden, in Code um. Dabei sind die Anweisungen klar. Die anderen beiden KIs, die ebenfalls Code programmieren können, gehen einen anderen Weg. Sie sind weniger praktisch orientiert und setzen nicht direkt visuelle Ergebnisse um, sondern gehen einen Schritt weiter. Sie bekommen nur eine Aufgabe, finden dafür selbst die Lösung und generieren dafür den passenden, funktionierenden Code. Dabei bekommen Sie Aufgaben ähnlich, zu denen die in Coding-Competitions gestellt werden und schneiden beim Lösen dieser sogar noch gut ab.

Coding Competitions

Coding Competitions sind vergleichbar mit Denksport Aufgaben. Es gibt diverse Competitions, die offen für alle sind. Manchmal stellen Unternehmen Aufgaben, die ihnen in der Entwicklung ihres Produktes weiterhelfen. Dann wird häufig vom Unternehmen ein Preisgeld gesetzt. So profitiert das Unternehmen von der Schwarmintelligenz des Internets und die Teilnehmenden können nicht nur ihre Kenntnisse unter Beweis stellen, sondern evtl. noch etwas gewinnen. Die Competitions und Aufgaben, bei denen es bei den KIs geht, die Code generieren können handelt es sich um einfachere Aufgaben wie z.B. Programmiere einen Sitzplan. Dabei sind dann verschiedene Anforderungen gegeben wie z.B. „Das Kino hat 8 Reihen á 10 Sitzen. A möchte im hinteren Drittel sitzen, B neben C, D nicht neben A“ usw. Dabei werden die Aufgaben und Anforderungen häufig in Fließtexten gestellt und müssen dann in Form eines kleinen Programmes gelöst werden.

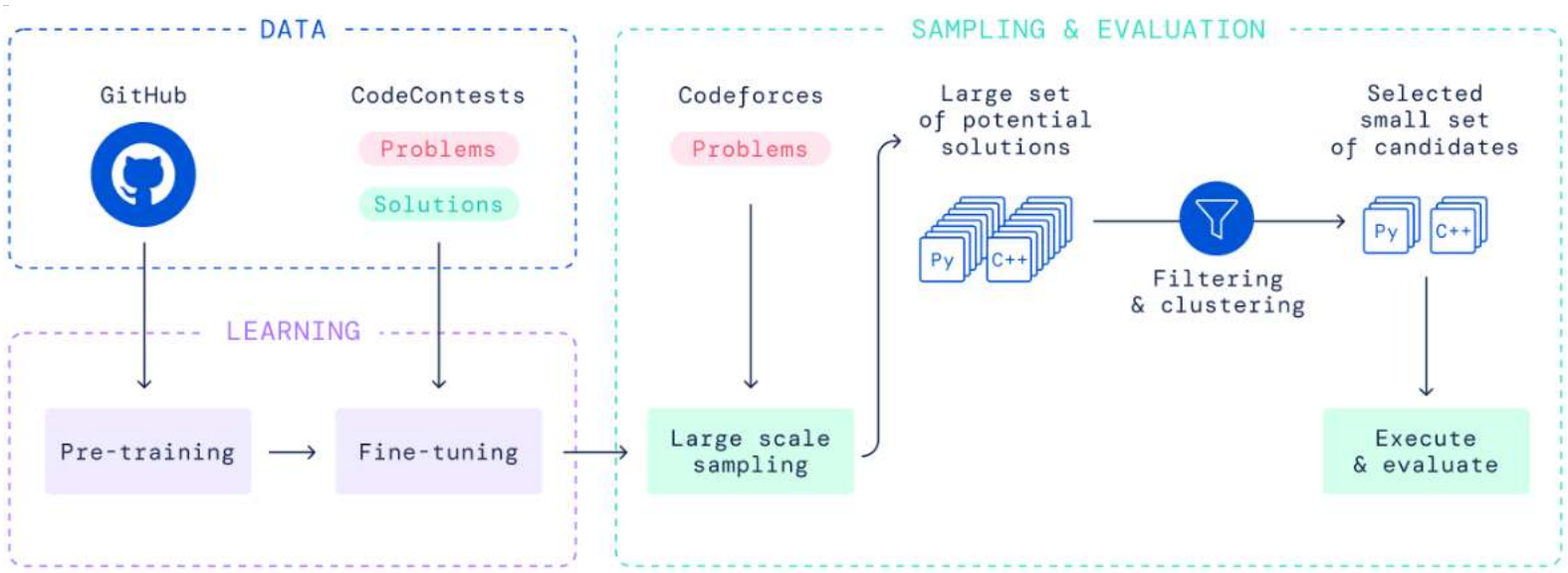
Deepcoder

Eine dieser KIs wurde in der Zusammenarbeit zwischen der Cambridge University und Microsoft entwickelt. Sie kann Aufgaben ähnlich zu denen aus Coding-Competition lösen. 2017 wurde ein Bericht zu Deepocoder veröffentlicht. Zu diesem Zeitpunkt war die KI noch limitiert. Sie konnte maximal fünf Zeilen Code schreiben. Viel mehr Informationen zur KI wurden nicht bekannt gegeben. Seitdem gab es auch keine weiteren Informationen zu dieser KI.

[6]

AlphaCode
Die zweite KI, die in der Lage ist, Code zu programmieren, ist eine KI des Google Tochterunternehmens DeepMind. Die KI wurde AlphaCode getauft. Im Februar 2022 gab DeepMind bekannt, dass AlphaCode an mehreren Coding-Competitions teilgenommen hat und dabei im Vergleich zu den menschlichen Mitstreiter:innen gut abgeschnitten hat. AlphaCode war insgesamt immer unter den Top 54 %. Auf der Website von Alpha Code können die Lösungen eingesehene werden. Zu jeder Aufgabe hat die KI mehrere Lösungen in den Sprachen Python und C++ generiert. Von allen Lösungen hat die KI die herausgefiltert, die funktionieren und diese dann eingereicht. (Deepmind, 2022) Wie diese KI lernt zeigt die Abbildung 8.

[7]



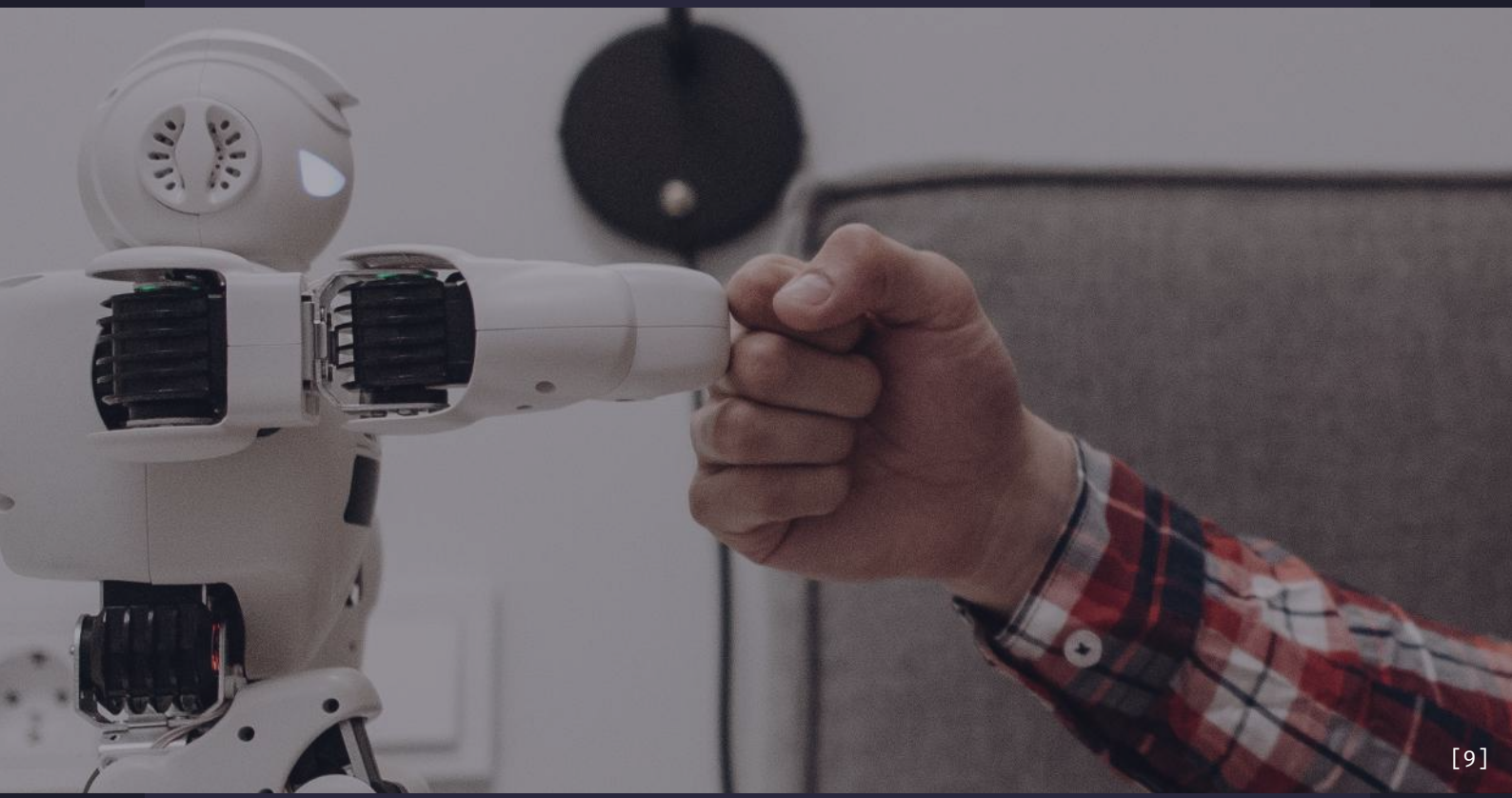
[8]

Fazit

> “You should think of it as something that could be an assistant to a programmer in the way that a calculator might once have helped an accountant.”
>
> Gary Marcus, an AI professor at New York University, told CNBC

Das war die Antwort von Gary Marcus auf die Frage, wie AI die Welt der Programmierer verändern wird. Es gibt viel Entwicklungen in diesem Bereich. Gerade in den letzten zwei bis drei Jahren und es ist zu erwarten, dass die Entwicklung noch schneller und weiter vorn gebracht wird. Dabei spielend die großen KIs von Unternehmen wie Open AI, Meta und Google eine große Rolle. Diese sind auch diejenigen, die alle ihre Ergebnisse publik machen und den Open Source Weg gehen. Was im Hintergrund entwickelt und geforscht wird, ist unklar.

Was aber sicher ist, dass Programmierer, egal ob Web, Frontend oder Backend, so schnell nicht ersetzt werden. Sie können zunehmend durch AI unterstützt werden. Dies bringt viele Vorteile. Es kann schneller und effizienter gearbeitet werden, außerdem braucht es zukünftig nicht mehr die tiefgreifenden Kenntnisse, um einfache Software zu erstellen. Solange alles funktioniert. Wie viele andere Jobs wird sich dieses Berufsfeld auch mit zunehmender Digitalisierung und Nutzung durch AI verändern. Aber keine Branche ist so schnell sich an neue Technologien anzupassen wie die IT-Branche. Daher kann gesagt werden, dass AI Programmierer nicht so bald ersetzen wird.



[9]

Quellen

02100 Digital. (2020, Oktober). Low-Code vs. No-Code—02100 Digital. https://www.02100.io/low-code-vs-no-code

Cognitive Class (Regisseur). (2017, März 14). Machine Learning—Supervised VS Unsupervised Learning. https://www.youtube.com/watch?v=cfj6yaYE86U

Computer Hope. (2017, Juli 12). What is a Transcompiler? https://www.computerhope.com/jargon/t/transcompiler.htm

Deepmind. (2022, Februar 2). Competitive programming with AlphaCode. https://www.deepmind.com/blog/competitive-programming-with-alphacode

Disruption Theory (Regisseur). (2020, August 29). GPT-3 Demo: New AI Algorithm Changes How We Interact With Technology. https://www.youtube.com/watch?v=8V20HkoiNtc

Martin, R. C. (Hrsg.). (2009). Clean code: A handbook of agile software craftsmanship. Prentice Hall.

Meta. (2020, Juli 21). Deep learning to translate between programming languages. https://ai.facebook.com/blog/deep-learning-to-translate-between-programming-languages/

OpenAI Codex. (2021, August 10). OpenAI. https://openai.com/blog/openai-codex/

Specht, F. (2022, Februar 7). Fachkräftelücke in den IT-Berufen so groß wie nie. Handelsblatt. https://www.handelsblatt.com/politik/deutschland/arbeitsmarkt-fachkraefteluecke-in-den-it-berufen-so-gross-wie-nie/28046062.html

Wikipedia. (2022). GitHub. In Wikipedia. https://de.wikipedia.org/w/index.php?title=GitHub&oldid=22447279

Bild-Quellen

Titelbild: Bild von Pexels

[1]: Photo by Pawel Nolbert on Unsplash

[2]: Bild von Thorsten Frenzel auf Pixabay

[3]: Git Hub Logo von https://logos-world.net/github-logo/

[4]: Screenshot aus dem Video https://www.youtube.com/watch?v=8V20HkoiNtc&ab_channel=DisruptionTheory

[5]: ebd.

[6]: Bild von StockSnap auf Pixabay

[7]: Bild von Steve Buissinne auf Pixabay

[8]: Abbildung von Deepmind, 2020

[9]: Foto von Pavel Danilyuk