

Definitionversuch 2

Künstliche Intelligenz ist ein Forschungsfeld, das **Methoden** entwickelt, um einzelne Fertigkeiten, die wir für intelligent halten, nachzuahmen.

- ▶ KI ist nicht universell/monolithisch; Problem \Rightarrow Teilprobleme \Rightarrow passende Einzelmethoden \Rightarrow KI-System
- ▶ Fähigkeiten und Grenzen von KI kann man nur verstehen, wenn man die **Methoden** versteht <https://ki-campus.org/podcasts/kikapiert>



KI-Campus

Die Lernplattform
für Künstliche Intelligenz

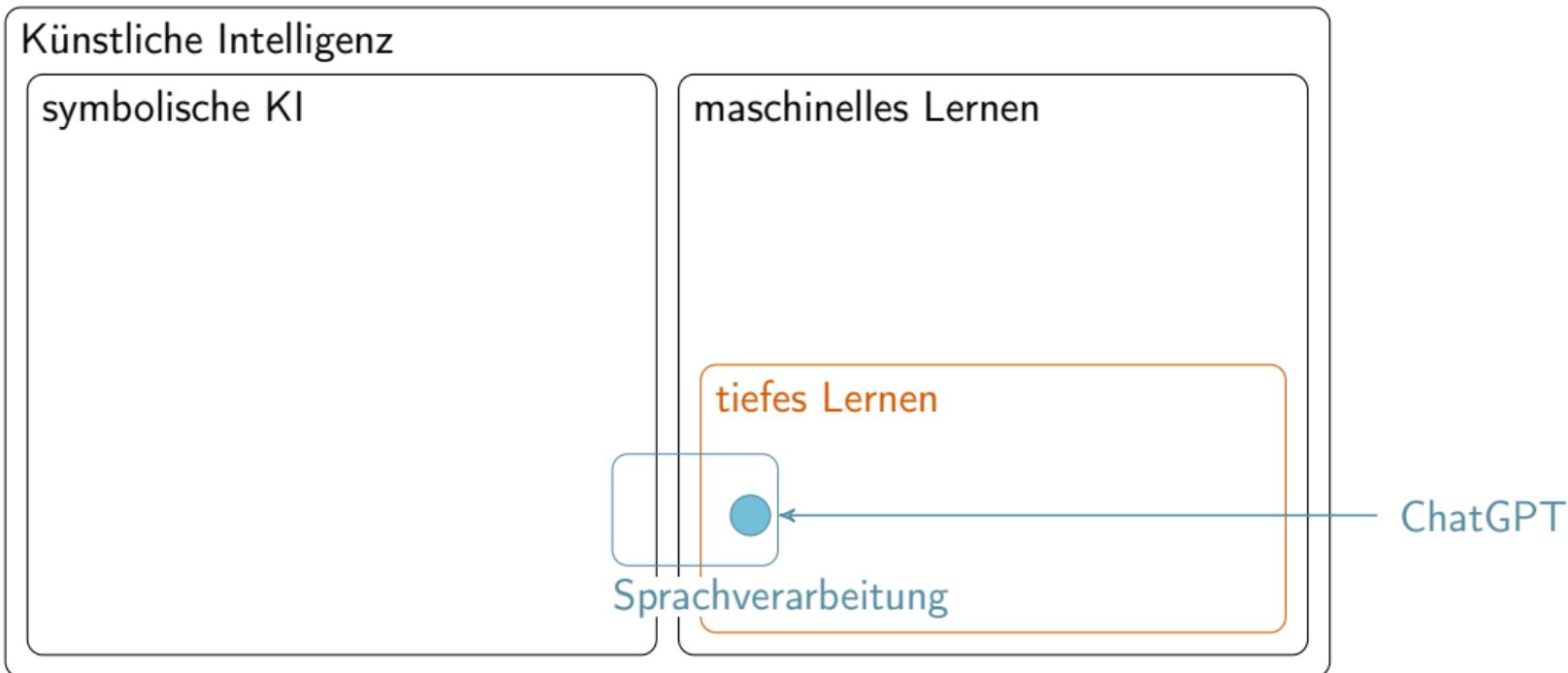
Künstliche Intelligenz

symbolische KI

maschinelles Lernen

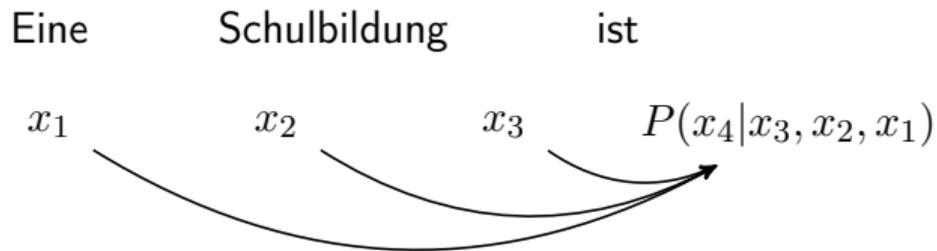
tiefes Lernen

Sprachverarbeitung



Was sind Sprachmodelle?

- ▶ Gegeben eine Reihe bisheriger Worte, ordne jedem möglichen Folgewort eine Wahrscheinlichkeit zu



- ▶ Gegeben eine Reihe bisheriger Worte, ordne jedem möglichen Folgewort eine Wahrscheinlichkeit zu

Eine	Schulbildung	ist	verpflichtend: 40%
x_1	x_2	x_3	wichtig: 60%
			$P(x_4 x_3, x_2, x_1)$

Was sind Sprachmodelle?

- ▶ Gegeben eine Reihe bisheriger Worte, ordne jedem möglichen Folgewort eine Wahrscheinlichkeit zu

Eine

Schulbildung

ist

wichtig

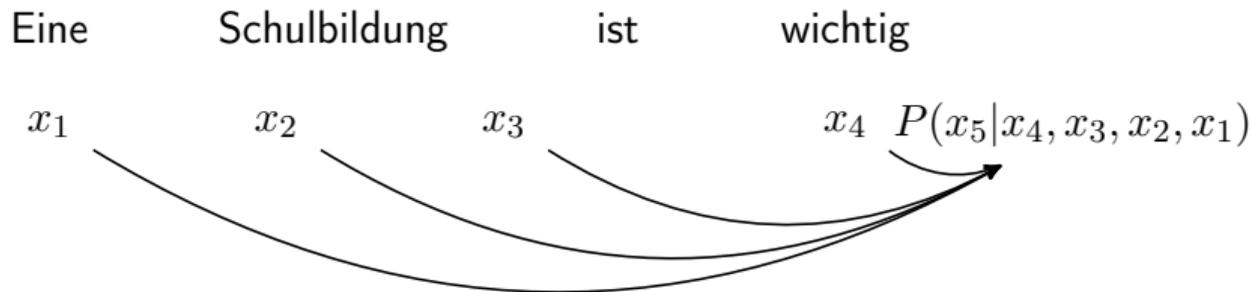
x_1

x_2

x_3

x_4

- ▶ Gegeben eine Reihe bisheriger Worte, ordne jedem möglichen Folgewort eine Wahrscheinlichkeit zu



- ▶ Gegeben eine Reihe bisheriger Worte, ordne jedem möglichen Folgewort eine Wahrscheinlichkeit zu

Eine

Schulbildung

ist

wichtig

zur

x_1

x_2

x_3

x_4

x_5

Was sind Sprachmodelle?

- ▶ Gegeben eine Reihe bisheriger Worte, ordne jedem möglichen Folgewort eine Wahrscheinlichkeit zu

Eine

Schulbildung

ist

wichtig

zur

Entfaltung

x_1

x_2

x_3

x_4

x_5

x_6

- ▶ Gegeben eine Reihe bisheriger Worte, ordne jedem möglichen Folgewort eine Wahrscheinlichkeit zu

Eine	Schulbildung	ist	wichtig	zur	Entfaltung
x_1	x_2	x_3	x_4	x_5	x_6

- ▶ Trainingsdaten: Großer Teil des Textes im Internet \Rightarrow „große“ Sprachmodelle
- ▶ Seit 2017: Transformer-Architektur & scaling (Vaswani u. a. 2017; Karpathy 2023)
- ▶ Wichtig: **Keine** explizite Berücksichtigung von Bedeutung/Wahrheitsgehalt; nur Wortkorrelationen (Bender u. a. 2021)

- ▶ Grundsätze lehren und leben:
 - ▶ Menschen haben weiterhin Verantwortung für die Texte (insb. Richtigkeits- und Plagiatsprüfung!)
 - ▶ Benutzung kennzeichnen, keine persönlichen Daten übergeben
- ▶ Kompetenzen schulen (fachlich + KI! fachübergreifend!), z.B.
 - ▶ Strategie 1: System nicht Text „erfinden“ lassen, sondern Kontext und Fakten vorgeben (Textverarbeitung statt -erzeugung)
 - ▶ Strategie 2: Umfassende Aufgaben in Teilaufgaben zerlegen („chaining“)
- ▶ Plagiatsdetektionssoftware meiden! (Paaßen, Burchardt und Pinkwart 2023)

Tutoring System for Programming

< greet > About Impressum Feedback Profile

Use the following signature to say (print) "Hello " to an input name:

```
def greet(name):  
    print()
```

Afterwards please greet Alice and Bob on your console.

Note: Python uses a fixed indentation for separating blocks. You may be already familiar to other programming languages like JAVA which usually use curly brackets for this purpose. However Python programs require you to always indent your blocks by either four spaces or a TAB.

Feedback for task greet

test_print_out

Test failure: You forgot to greet Alice and Bob on the console.

test_greet

Test success

```
1 def greet(name):  
2     print(f"Hello {name}")  
3  
4 greet("Alice")
```

Run Submit

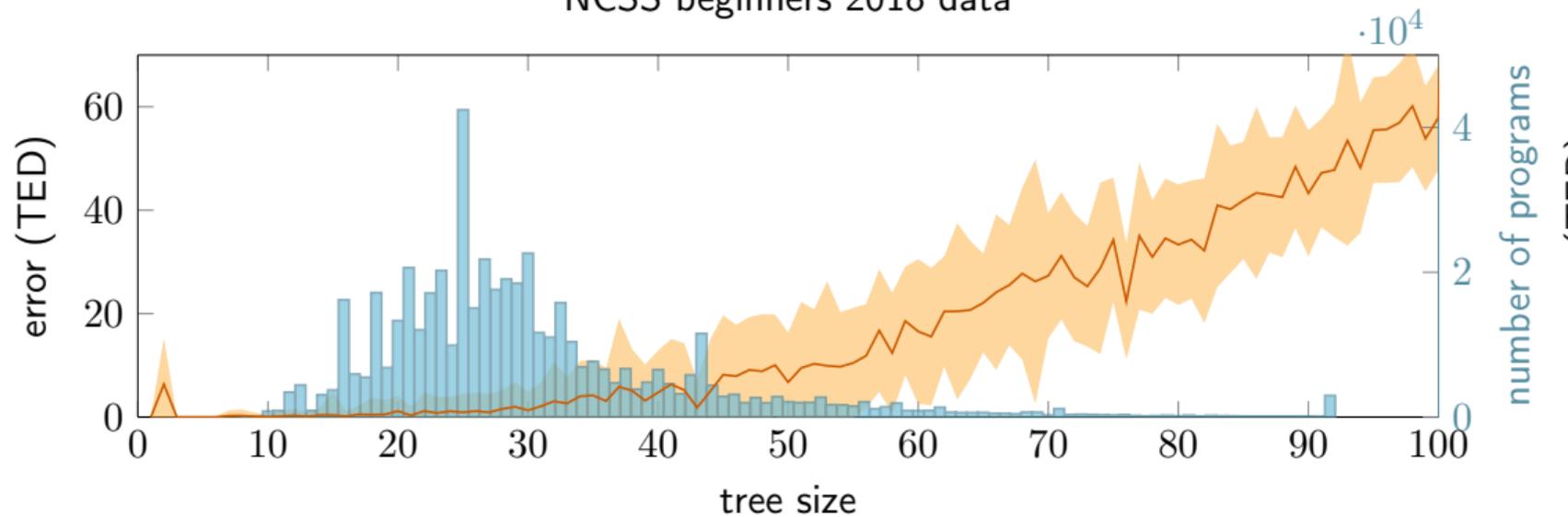
- Bender, Emily M. u. a. (2021). „On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?“ In: **Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency**. FAccT '21. Virtual Event, Canada, S. 610–623. DOI: 10.1145/3442188.3445922.
- Hattie, John und Helen Timperley (März 2007). „The Power of Feedback“. en. In: **Review of Educational Research** 77.1, S. 81–112. ISSN: 0034-6543, 1935-1046. DOI: 10.3102/003465430298487. URL: <http://journals.sagepub.com/doi/10.3102/003465430298487> (besucht am 20. 11. 2023).
- Karpathy, Andrej (2023). **Let's build GPT: from scratch, in code, spelled out**. URL: <https://www.youtube.com/watch?v=kCc8FmEb1nY>.

- Ouyang, Long u. a. (2022). „Training language models to follow instructions with human feedback“. In: **Advances in Neural Information Processing Systems**. Hrsg. von S. Koyejo u. a. Bd. 35. Curran Associates, Inc., S. 27730–27744. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.
- Paaßen, Benjamin, Aljoscha Burchardt und Niels Pinkwart (27. Feb. 2023). **Chancen, Potenziale und Grenzen von ChatGPT in der Bildung – Stellungnahme des DFKI Labor Berlin**. URL: <https://www.dfki.de/web/news/chancen-potenziale-und-grenzen-von-chatgpt-in-der-bildung-stellungnahme-des-dfki-labor-berlin>.
- Schleiss, Johannes u. a. (März 2023). **Künstliche Intelligenz in der Bildung. Drei Zukunftsszenarien und fünf Handlungsfelder**. German. Discussion Paper. KI-Campus. DOI: 10.5281/zenodo.7702620.

Vaswani, Ashish u. a. (2017). „Attention is All you Need“. In: **Proceedings of the 30th International Conference on Advances in Neural Information Processing Systems (NeurIPS 2017)**. Hrsg. von I. Guyon u. a. URL: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

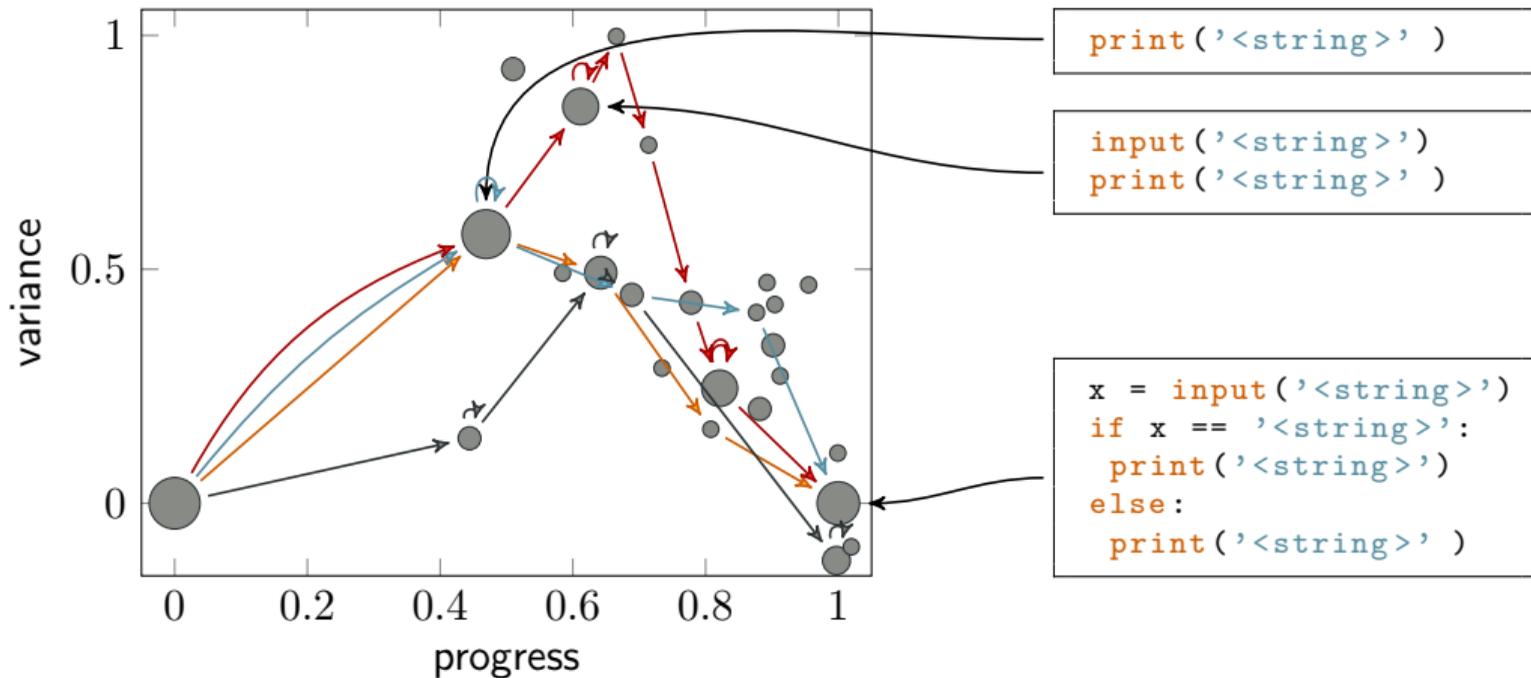
- ▶ 448,992 Python programs from the beginners challenge of the 2018 National Computer Science School
- ▶ encoding dimension 256, crossentropy loss, learning rate 10^{-3} , ADAM optimizer
- ▶ ca. 1 week of training time, 130k batches of 32 programs each
- ▶ **Result:** ast2vec, a **pre-trained** neural network

NCSS beginners 2018 data



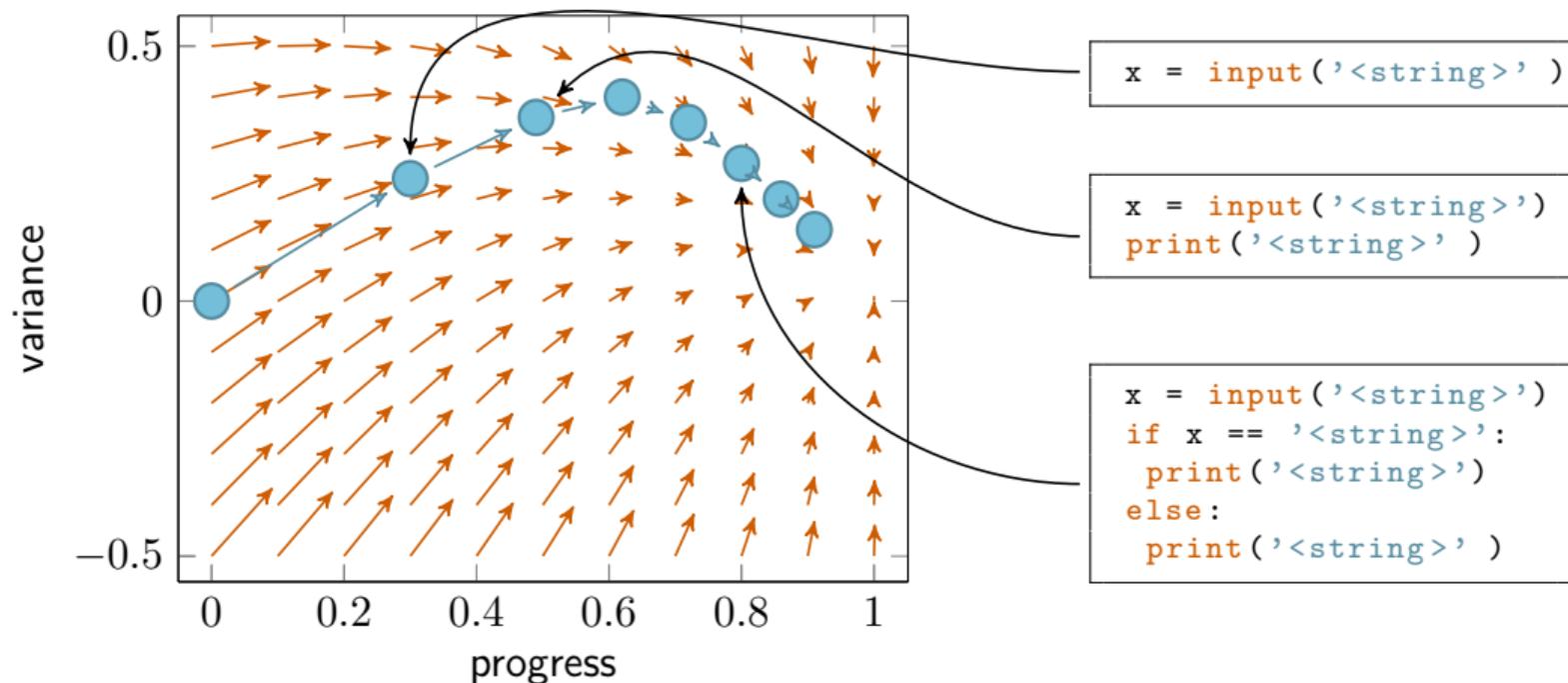
Progress-Variance plot

- ▶ x -axis: direction from empty solution to goal; y -axis: orthogonal direction with maximum variance



Prediction

- ▶ Predict a student's next program as $f(\vec{x}) = \vec{x} + \mathbf{W} \cdot (\vec{b} - \vec{x})$
- ▶ Learn \mathbf{W} via linear regression; set \vec{b} to closest correct solution
- ⇒ Provably converges to \vec{b} (for strong enough regularization)



- Linnainmaa, Seppo (1970). “The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors”. MA thesis. University of Helsinki.
- Minsky, Marvin and Seymour Papert (1969). **Perceptrons: An introduction to computational geometry**. Cambridge, MA, USA: MIT Press.
- Olazaran, Mikel (1996). “A Sociological Study of the Official History of the Perceptrons Controversy”. In: **Social Studies of Science** 26.3, pp. 611–659. doi: 10.1177/030631296026003005.
- Paaßen, Benjamin, Malwina Dywel, et al. (July 24, 2022). “Sparse Factor Autoencoders for Item Response Theory”. In: **Proceedings of the 15th International Conference on Educational Data Mining (EDM 2022)** (Durham, UK). Ed. by Alexandra I. Cristea et al., pp. 17–26. doi: 10.5281/zenodo.6853067.

- Paaßen, Benjamin, Irena Koprinska, and Kalina Yacef (2022). “Recursive Tree Grammar Autoencoders”. In: **Machine Learning** 111. Special Issue of the ECML PKDD 2022 Journal Track, pp. 3393–3423. doi: 10.1007/s10994-022-06223-7. url: <https://arxiv.org/abs/2012.02097>.
- Paaßen, Benjamin, Jessica McBroom, et al. (2021). “Mapping Python Programs to Vectors using Recursive Neural Encodings”. In: **Journal of Educational Datamining** 13.3, pp. 1–35. doi: 10.5281/zenodo.5634224. url: <https://jedm.educationaldatamining.org/index.php/JEDM/article/view/499>.
- Rosenblatt, Frank (1958). “The perceptron: A probabilistic model for information storage and organization in the brain”. In: **Psychological Review** 65.6, pp. 386–408. doi: 10.1037/h0042519.
- Rumelhart, David, Geoffrey Hinton, and Ronald Williams (1986). “Learning representations by back-propagating errors”. In: **nature** 323.6088, pp. 533–536. doi: 10.1038/323533a0.