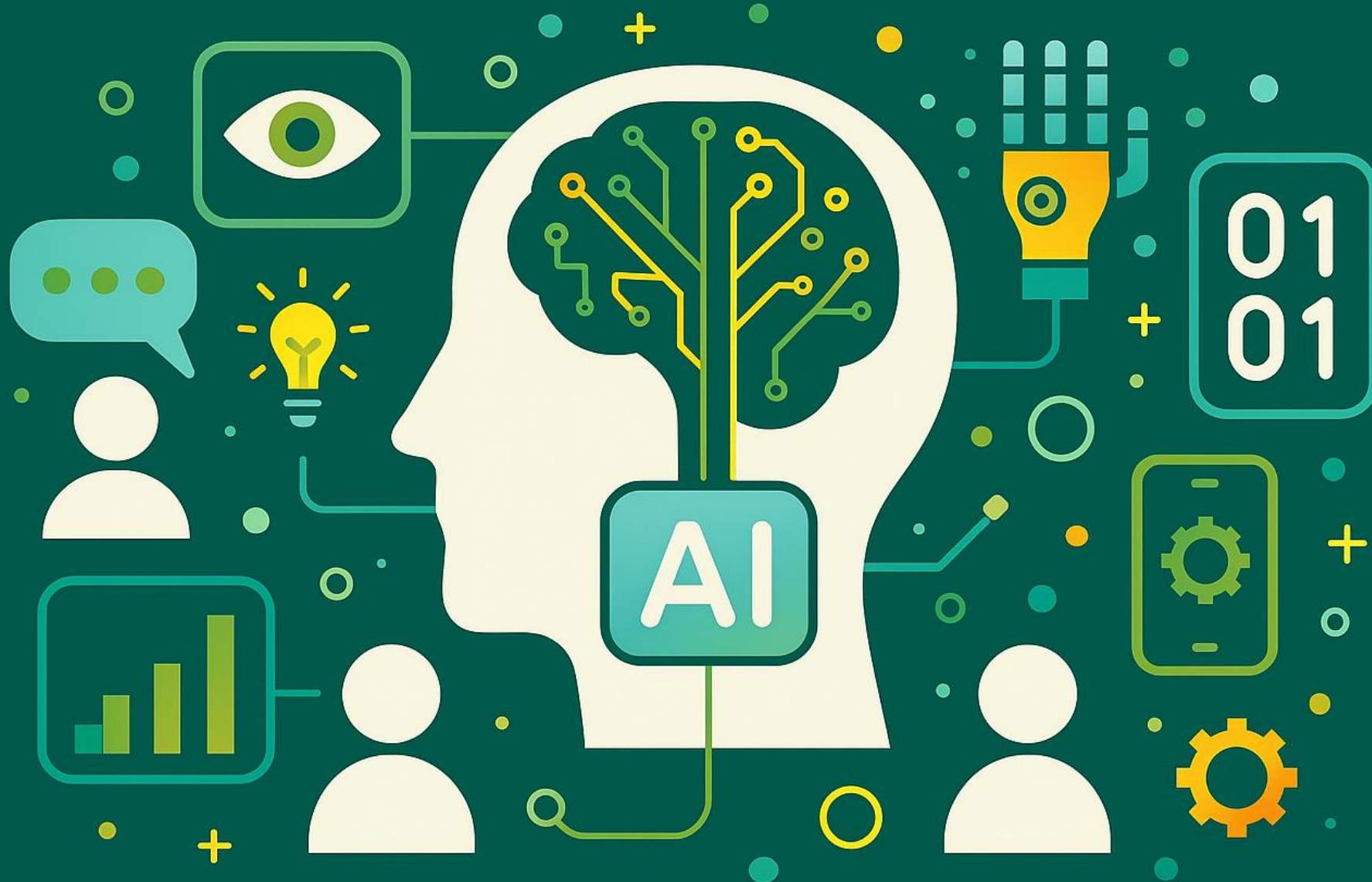


AI: An overview



Licence Disclaimer

This document is distributed under the terms of the GNU General Public Licence, Version 3, dated 29 June 2007. It is intended to promote freedom to use, study, modify, and share the content herein, in accordance with the principles of free and open-source documentation. By accessing, reproducing, or modifying this document, you agree to comply with the conditions set forth in the GNU GPL v3. A full copy of the licence is available at <https://www.gnu.org/licenses/gpl-3.0.html>.

This licence applies to the document as a whole, including any derivative works, unless otherwise stated. No warranties are provided, and the document is offered “as-is” without liability for its use or interpretation.

Attribution Requirement

When using, sharing, or adapting this document for any individual, group, or organisation, proper citation of the original author is required. Please cite as follows:

Alsaggaf, I. (2025) *Introduction to Artificial Intelligence*. Available at:
<https://github.com/ibrahimsaggaf/Introduction-to-Artificial-Intelligence> (Accessed: [insert date]).

Content

- A brief history of AI
- AI in practice
- AI ethics and limitations
- Q&A

Lab session: Setting up a Python environment

A brief history of AI

1943

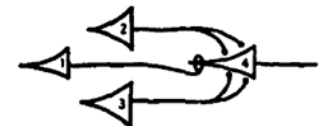
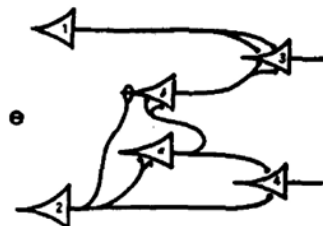
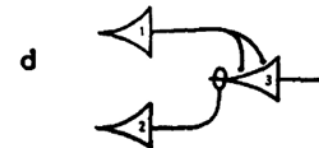
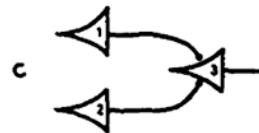
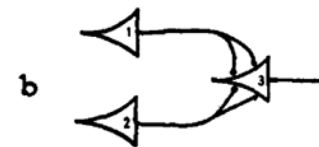
NN

A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY

WARREN S. MCCULLOCH and WALTER H. PITTS

Because of the “all-or-none” character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes. It is shown that many particular choices among possible neurophysiological assumptions are equivalent, in the sense that for every net behaving under one assumption, there exists another net which behaves under the other and gives the same results, although perhaps not in the same time. Various applications of the calculus are discussed.

A Logical Calculus of Ideas Immanent in Nervous Activity

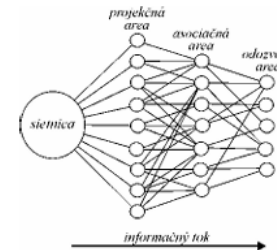


A brief history of AI



Frank Rosenblatt (1928 - 1969)

Rosenblattova predstava o perceptróne



Oblasť, kde sa premieta optickým systémom oka pozorovaný objekt, sa nazýva sietnica. Tá prenáša binárne hodnoty do vrstvy nazývanej projekčná area, kde sa binárne kódovaný obraz numericky predspracováva. Spoje medzi sietnicou a projekčnou oblasťou sú pevné a neadaptabilné. Spoje do druhej vrstvy (asociačnej arey) a tiež aj do tretej vrstvy (odozvová area) sú stochasticky generované. Základným cieľom adaptačného procesu perceptrónu je nastaviť váhové koeficienty spojov tak, aby aktivity neurónov z tretej vrstvy (odozvová oblasť) správne klasifikovali obraz dopadajúci na sietnicu.

- F. Rosenblat, The perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review* **65** (1958), 386-408.
- F. Rosenblatt, *Principles of Neurodynamics*. Spartan Books, Washington D.C., 1962.

A brief history of AI



Biol. Cybernetics 36, 193–202 (1980)

Biological
Cybernetics
© by Springer-Verlag 1980

Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position

Kunihiko Fukushima

NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Japan

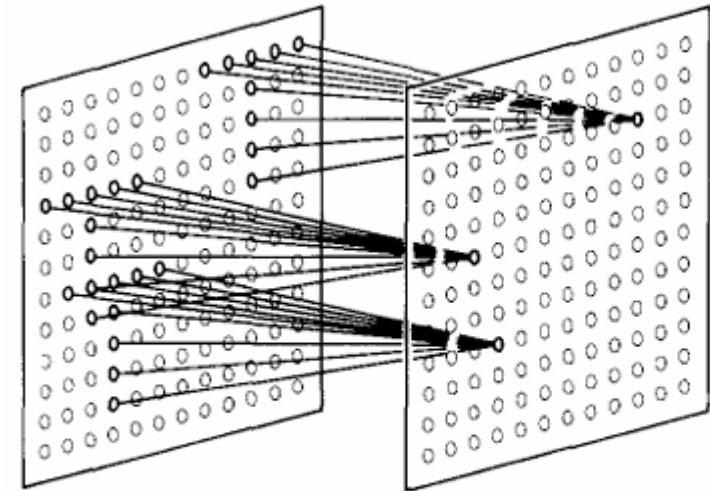
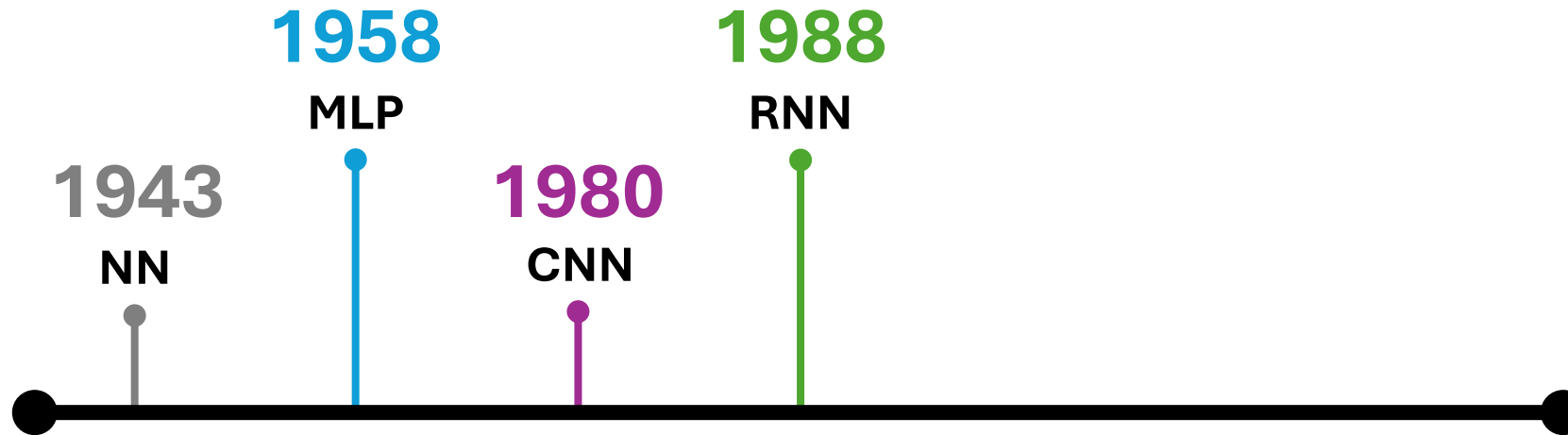


Fig. 3. Illustration showing the input interconnections to the cells within a single cell-plane

A brief history of AI



Generalization of Backpropagation with Application to a Recurrent Gas Market Model

PAUL J. WERBOS

U.S. Department of Energy

(Received August 1987; revised and accepted May 1988)

Abstract—Backpropagation is often viewed as a method for adapting artificial neural networks to classify patterns. Based on parts of the book by Rumelhart and colleagues, many authors equate backpropagation with the generalized delta rule applied to fully-connected feedforward networks. This paper will summarize a more general formulation of backpropagation, developed in 1974, which does more justice to the roots of the method in numerical analysis and statistics, and also does more justice to creative approaches expressed by neural modelers in the past year or two. It will discuss applications of backpropagation to forecasting over time (where errors have been halved by using methods other than least squares), to optimization, to sensitivity analysis, and to brain research.

This paper will go on to derive a generalization of backpropagation to recurrent systems (which input their own output), such as hybrids of perceptron-style networks and Grossberg/Hopfield networks. Unlike the proposal of Rumelhart, Hinton, and Williams, this generalization does not require the storage of intermediate iterations to deal with continuous recurrence. This generalization was applied in 1981 to a model of natural gas markets, where it located sources of forecast uncertainty related to the use of least squares to estimate the model parameters in the first place.

Keywords—Backpropagation, Recurrent, Continuous time, Reinforcement learning, Energy models, Prediction, Modelling, Cerebral cortex.

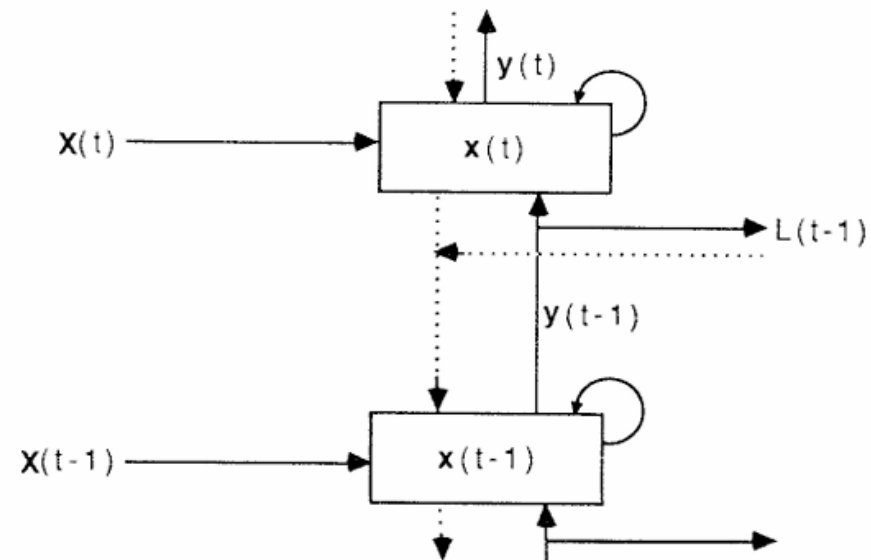
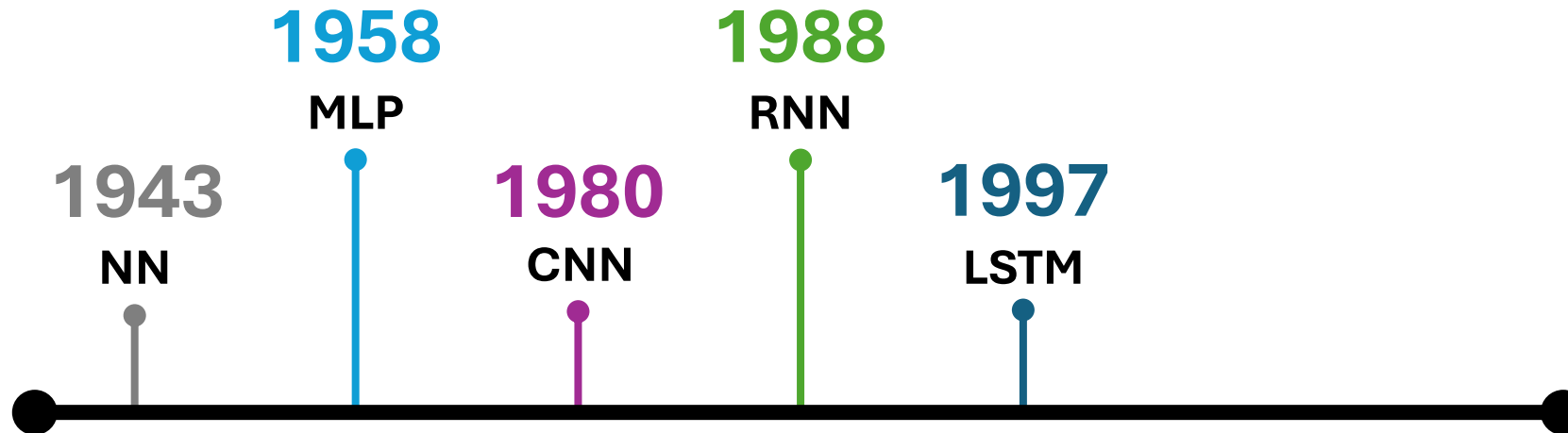


FIGURE 4. Proposed approach to recurrent networks.

A brief history of AI



Communicated by Ronald Williams

Long Short-Term Memory

Sepp Hochreiter

Fakultät für Informatik, Technische Universität München, 80290 München, Germany

Jürgen Schmidhuber

IDSIA, Corso Elvezia 36, 6900 Lugano, Switzerland

Learning to store information over extended time intervals by recurrent backpropagation takes a very long time, mostly because of insufficient, decaying error backflow. We briefly review Hochreiter's (1991) analysis of this problem, then address it by introducing a novel, efficient, gradient-based method called long short-term memory (LSTM). Truncating the gradient where this does not do harm, LSTM can learn to bridge minimal time lags in excess of 1000 discrete-time steps by enforcing constant error flow through constant error carousels within special units. Multiplicative gate units learn to open and close access to the constant error flow. LSTM is local in space and time; its computational complexity per time step and weight is $O(1)$. Our experiments with artificial data involve local, distributed, real-valued, and noisy pattern representations. In comparisons with real-time recurrent learning, back propagation through time, recurrent cascade correlation, Elman nets, and neural sequence chunking, LSTM leads to many more successful runs, and learns much faster. LSTM also solves complex, artificial long-time-lag tasks that have never been solved by previous recurrent network algorithms.

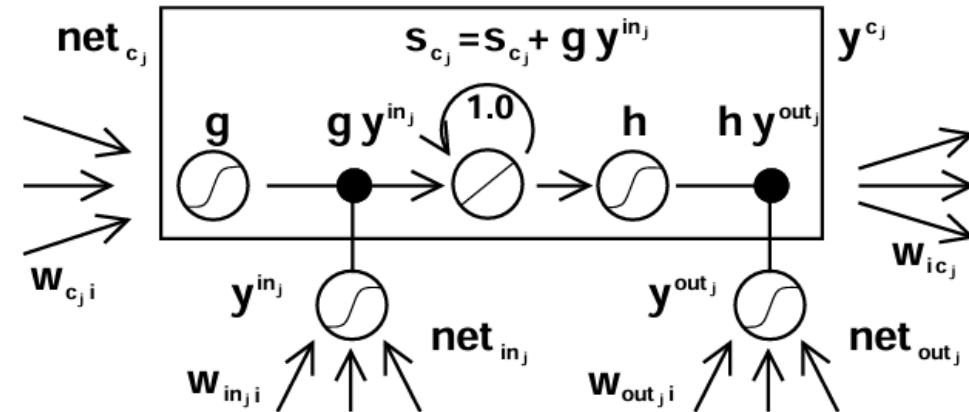
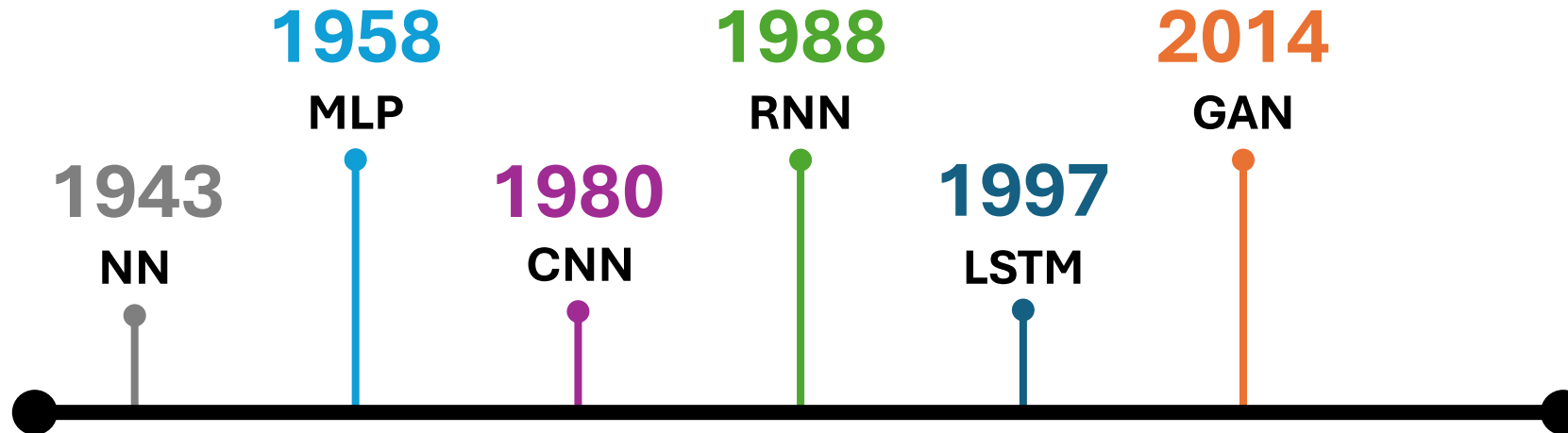


Figure 1: Architecture of memory cell c_j (the box) and its gate units in_j , out_j . The self-recurrent connection (with weight 1.0) indicates feedback with a delay of one time step. It builds the basis of the CEC. The gate units open and close access to CEC. See text and appendix A.1 for details.

A brief history of AI



Generative Adversarial Nets

Ian J. Goodfellow*, Jean Pouget-Abadie†, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair‡, Aaron Courville, Yoshua Bengio§
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

Abstract

We propose a new framework for estimating generative models via an adversarial process, in which we simultaneously train two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G . The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. In the space of arbitrary functions G and D , a unique solution exists, with G recovering the training data distribution and D equal to $\frac{1}{2}$ everywhere. In the case where G and D are defined by multilayer perceptrons, the entire system can be trained with backpropagation. There is no need for any Markov chains or unrolled approximate inference networks during either training or generation of samples. Experiments demonstrate the potential of the framework through qualitative and quantitative evaluation of the generated samples.

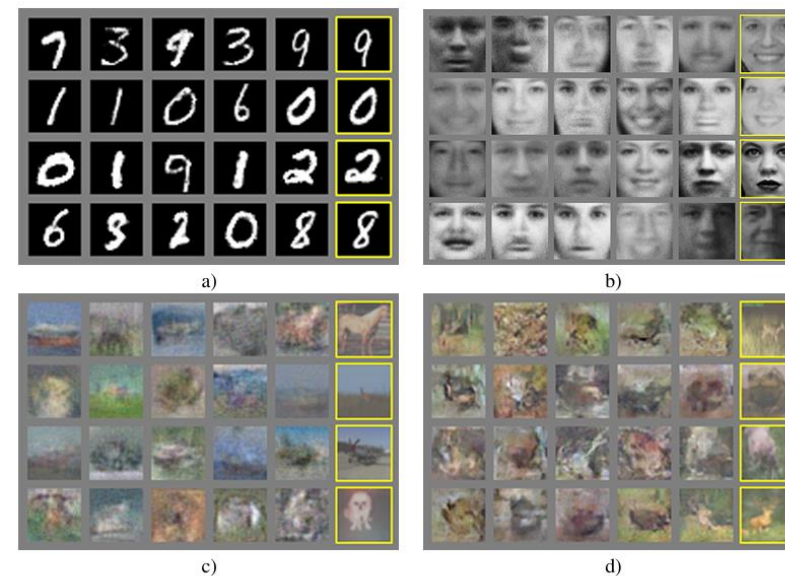
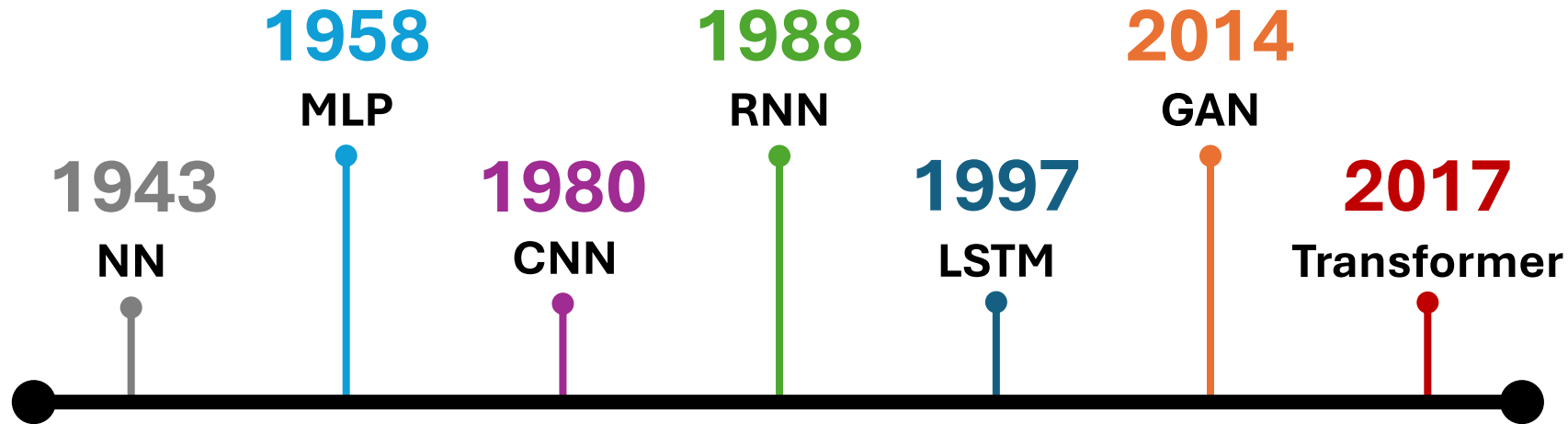


Figure 2: Visualization of samples from the model. Rightmost column shows the nearest training example of the neighboring sample, in order to demonstrate that the model has not memorized the training set. Samples are fair random draws, not cherry-picked. Unlike most other visualizations of deep generative models, these images show actual samples from the model distributions, not conditional means given samples of hidden units. Moreover, these samples are uncorrelated because the sampling process does not depend on Markov chain mixing. a) MNIST b) TFD c) CIFAR-10 (fully connected model) d) CIFAR-10 (convolutional discriminator and “deconvolutional” generator)

A brief history of AI



Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaier@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

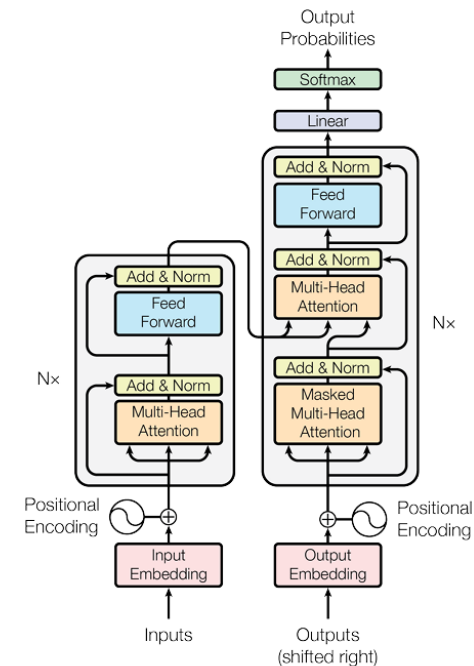


Figure 1: The Transformer - model architecture.

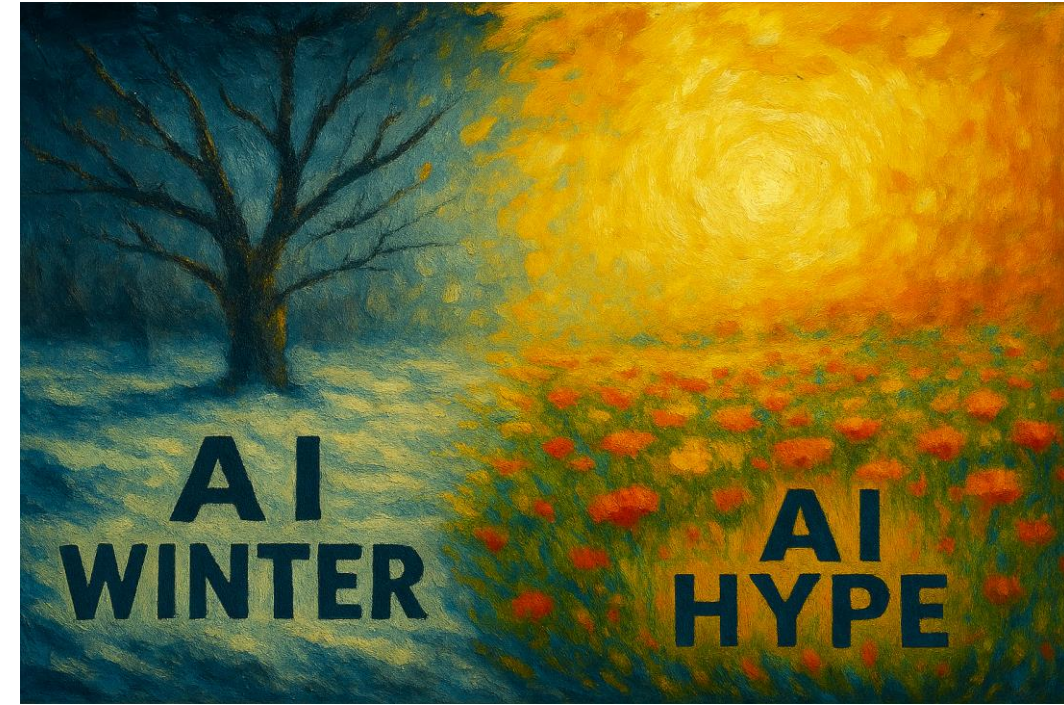
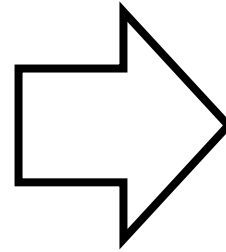


**AI
WINTER**

**AI
HYPE**

Text-to-image generative tasks

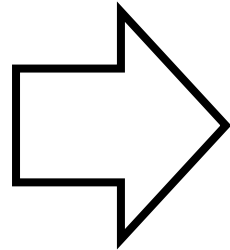
You are an artist. Paint a picture that resembles the following: An "AI winter" is a period of reduced funding, interest, and progress in artificial intelligence research, caused by a cycle of over-optimism, failed expectations, and criticism. These periods have occurred throughout AI's history when the technology failed to meet hyped promises, leading to a decrease in public and financial support. However, in 2010s the "AI hype" has happened, accelerating significantly in the 2020s. This was triggered by major breakthroughs in deep learning and generative AI. Your picture should capture the transition from "AI winter" to "AI hype".



Generated using <https://copilot.microsoft.com>

Text-to-text generative tasks

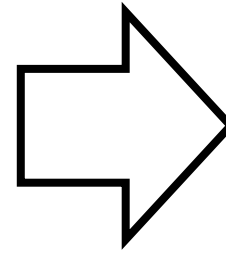
You are an AI subject matter expert with 20 years of experience. Write one paragraph that explains how the Kingdom of Saudi Arabia's vision 2030 aligns with artificial intelligence. Support your arguments with one example. Your answer shouldn't exceed 10 lines.



Saudi Arabia's Vision 2030 integrates artificial intelligence (AI) as a catalyst for economic diversification, digital transformation, and public sector modernization. AI enables smarter decision-making, predictive analytics, and automation across key sectors like healthcare, energy, and urban development. A notable example is the Saudi Data and Artificial Intelligence Authority (SDAIA), which launched the National Strategy for Data and AI to position the Kingdom among the top AI nations by 2030. SDAIA has embedded AI offices in 23 government entities and initiated large-scale training programs to upskill public employees. These efforts ensure ethical AI adoption while accelerating innovation. Through sovereign cloud infrastructure and high-performance computing, Saudi Arabia is building the digital backbone to realize Vision 2030's ambitions.

Text classification

Saudi Arabia's Vision 2030 integrates artificial intelligence (AI) as a catalyst for economic diversification, digital transformation, and public sector modernization. AI enables smarter decision-making, predictive analytics, and automation across key sectors like healthcare, energy, and urban development. A notable example is the Saudi Data and Artificial Intelligence Authority (SDAIA), which launched the National Strategy for Data and AI to position the Kingdom among the top AI nations by 2030. SDAIA has embedded AI offices in 23 government entities and initiated large-scale training programs to upskill public employees. These efforts ensure ethical AI adoption while accelerating innovation. Through sovereign cloud infrastructure and high-performance computing, Saudi Arabia is building the digital backbone to realize Vision 2030's ambitions.



Entertainment



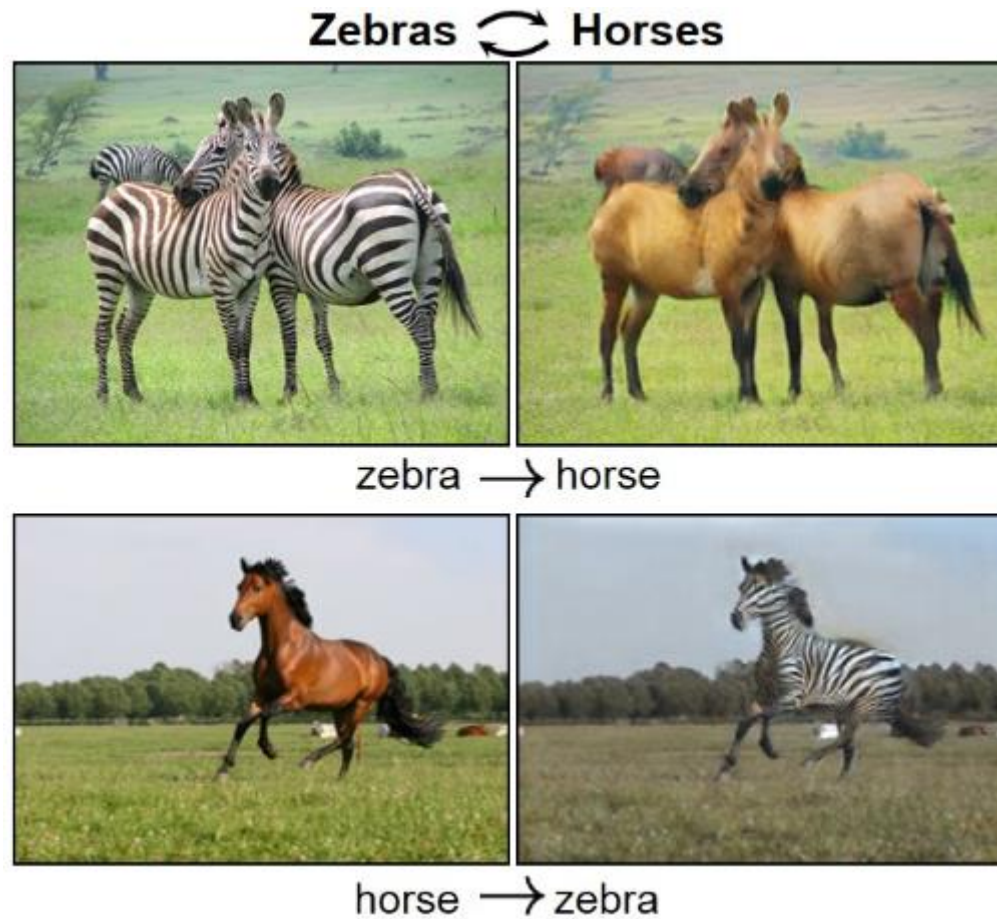
Healthcare



Business



Image-to-image translation



Can artificial intelligence contribute to:

Social discrimination

?

Unfair decisions

?

Biased outcomes

?

Can artificial intelligence contribute to:

Social discrimination

Yes

Unfair decisions

Yes

Biased outcomes

Yes

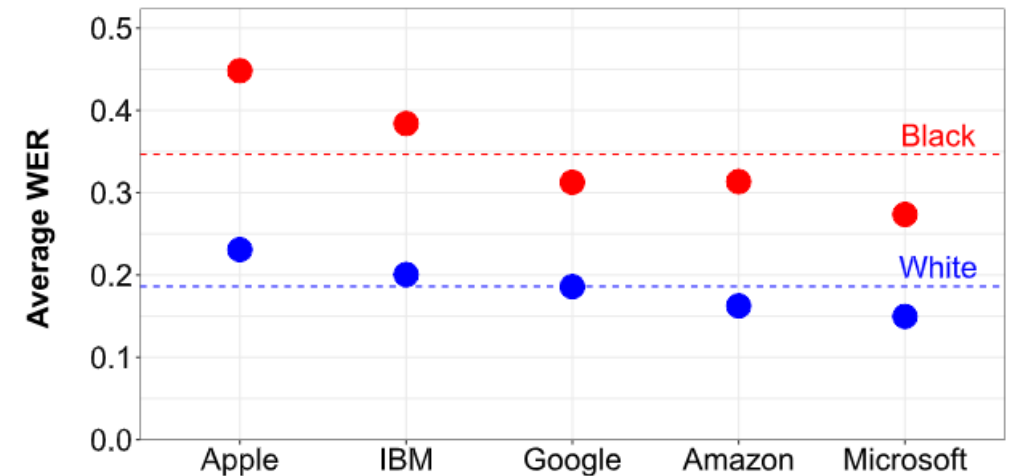
In 2020, a study reviewed Automated Speech Recognition (ASR) systems from **Amazon, Apple, Google, IBM, and Microsoft** found that they have **higher error** rates when transcribing black people's voices than white people's.

Racial disparities in automated speech recognition

Allison Koenecke^a, Andrew Nam^b, Emily Lake^c, Joe Nudell^d, Minnie Quartey^e, Zion Mengesha^c, Connor Toups^c, John R. Rickford^c, Dan Jurafsky^{c,f}, and Sharad Goel^{d,1}

^aInstitute for Computational & Mathematical Engineering, Stanford University, Stanford, CA 94305; ^bDepartment of Psychology, Stanford University, Stanford, CA 94305; ^cDepartment of Linguistics, Stanford University, Stanford, CA 94305; ^dDepartment of Management Science & Engineering, Stanford University, Stanford, CA 94305; ^eDepartment of Linguistics, Georgetown University, Washington, DC 20057; and ^fDepartment of Computer Science, Stanford University, Stanford, CA 94305

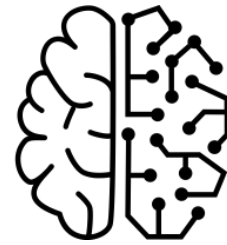
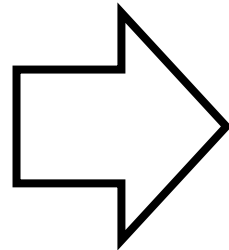
Edited by Judith T. Irvine, University of Michigan, Ann Arbor, MI, and approved February 12, 2020 (received for review October 5, 2019)



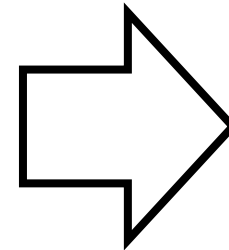
AI ethics

Biased training data

Vehicle	Colour	Points
car	red	95
car	blue	99
bicycle	green	15
scooter	yellow	32

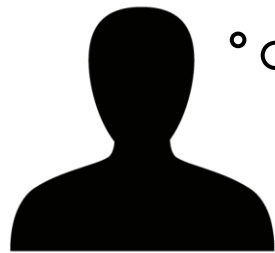


AI model

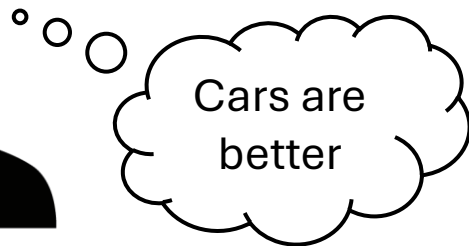


Vehicle	Colour	Points
car	yellow	?

Vehicle	Colour	Points
bicycle	red	?



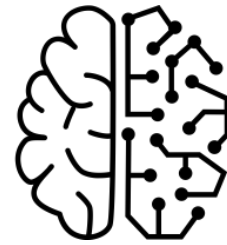
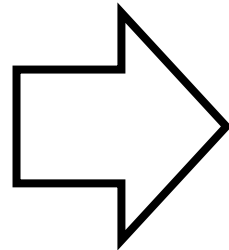
Creator



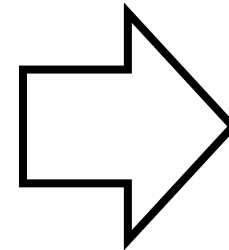
AI ethics

Biased training data

Vehicle	Colour	Points
car	red	95
car	blue	99
bicycle	green	15
scooter	yellow	32

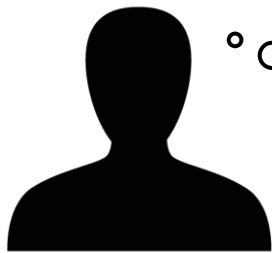


AI model

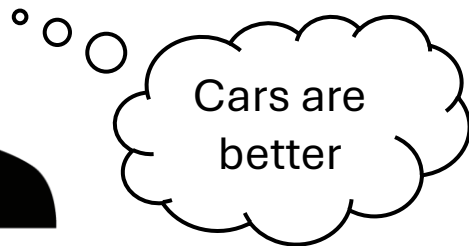


Vehicle	Colour	Points
car	yellow	High

Vehicle	Colour	Points
bicycle	red	Low



Creator



AI ethics

Between 2020 and 2025, around more than **6,000** research works related to AI ethics were published.

COGENT BUSINESS & MANAGEMENT
2025, VOL. 12, NO. 1, 2551809
<https://doi.org/10.1080/23311975.2025.2551809>



INFORMATION & TECHNOLOGY MANAGEMENT | RESEARCH ARTICLE | OPEN ACCESS | [Check for updates](#)

Artificial intelligence (AI) and ethical concerns: a review and research agenda

Munmun Ghosh

Symbiosis Institute of Media and Communication, Symbiosis International (Deemed University), Pune, India

SPRINGER NATURE Link

[Find a journal](#) [Publish with us](#) [Track your research](#) [Search](#)

[Home](#) > [Minds and Machines](#) > Article

The Ethics of AI Ethics: An Evaluation of Guidelines

[Open access](#) | Published: 01 February 2020

Volume 30, pages 99–120, (2020) [Cite this article](#)



[This journal](#) [Journals](#) [Publish](#) [News & events](#) [About Cell](#)

REVIEW · Volume 2, Issue 9, 100314, September 10, 2021 · [Open Access](#)

A high-level overview of AI ethics

[Emre Kazim](#) [✉](#) · [Adriano Soares Koshiyama](#) [✉](#)

Part of Springer Nature

BMC Medical Ethics

[Home](#) [About](#) [Articles](#) [Submission Guidelines](#) [Join The Board](#) [Collections](#) [Submit manuscript](#)

Systematic Review | [Open access](#) | Published: 27 May 2025

Ethical and social considerations of applying artificial intelligence in healthcare—a two-pronged scoping review

[Emanuele Ratti](#), [Michael Morrison](#) [✉](#) & [Ivett Jakab](#)

[BMC Medical Ethics](#) **26**, Article number: 68 (2025) | [Cite this article](#)

scientific reports

[Explore content](#) [About the journal](#) [Publish with us](#)

[nature](#) > [scientific reports](#) > [articles](#) > article

Article | [Open access](#) | Published: 30 October 2025

A framework for AI ethics literacy: development, validation, and its role in fostering students' self-rated learning competence

[Junfeng Yang](#) [✉](#), [Wenyan Xie](#) & [Junjie Ni](#)

[Scientific Reports](#) **15**, Article number: 38030 (2025) | [Cite this article](#)

SPRINGER NATURE Link

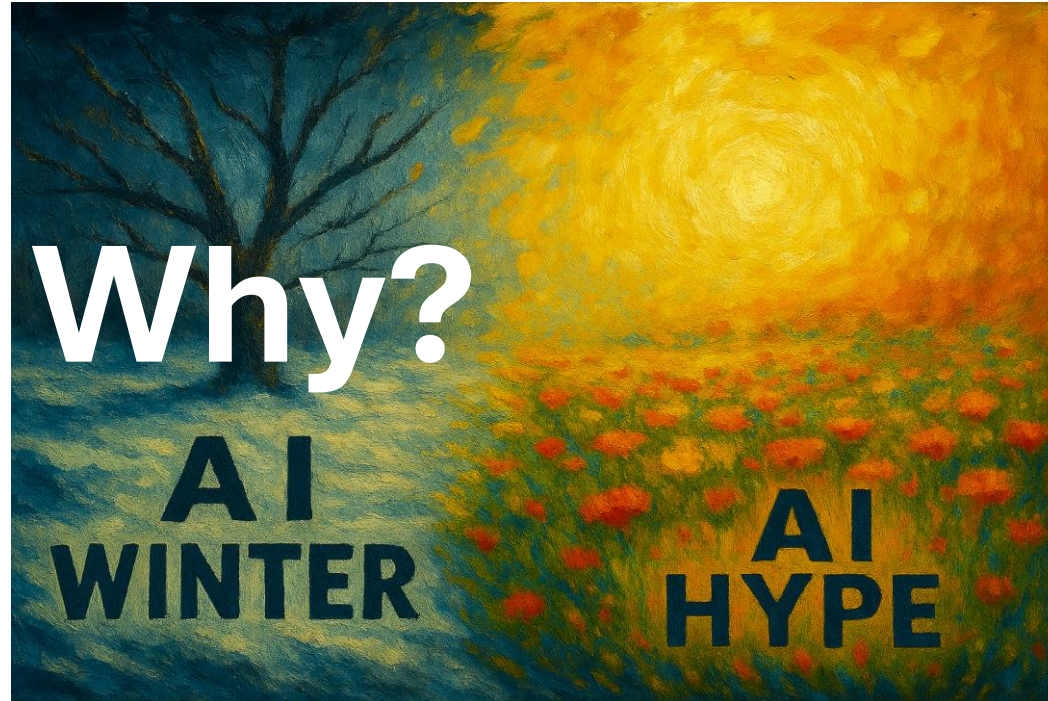
[Find a journal](#) [Publish with us](#) [Track your research](#) [Search](#)

[Home](#) > [Minds and Machines](#) > Article

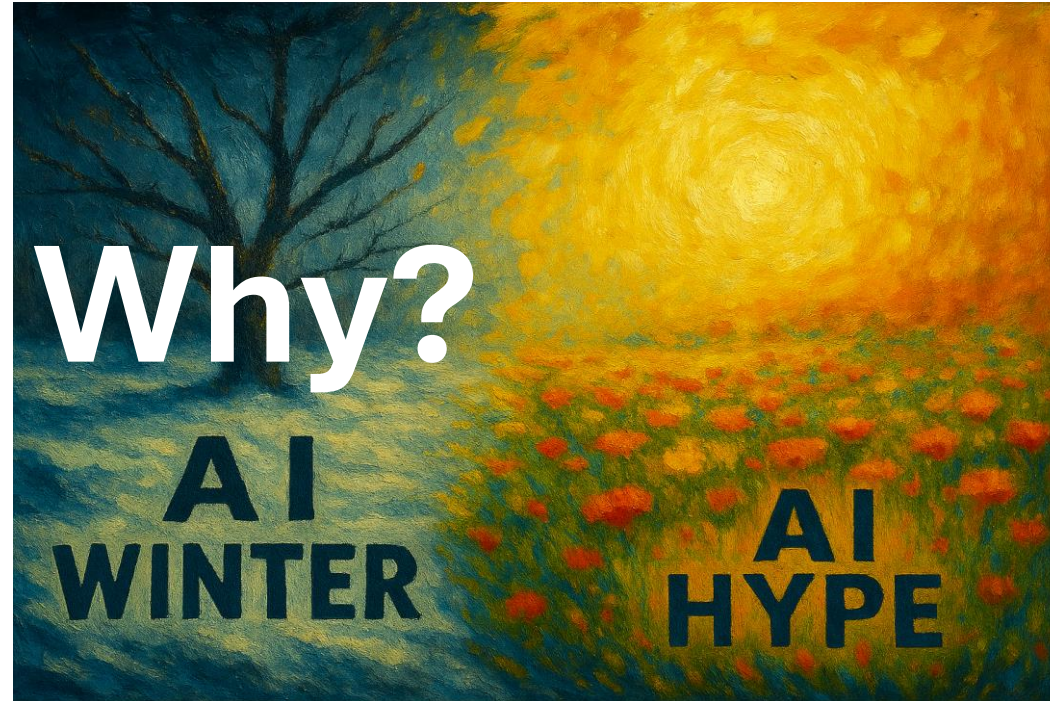
Mapping the Ethics of Generative AI: A Comprehensive Scoping Review

[Open access](#) | Published: 17 September 2024

Volume 34, article number 39, (2024) [Cite this article](#)



- Overly ambitious expectations
- Unmet promises



Data



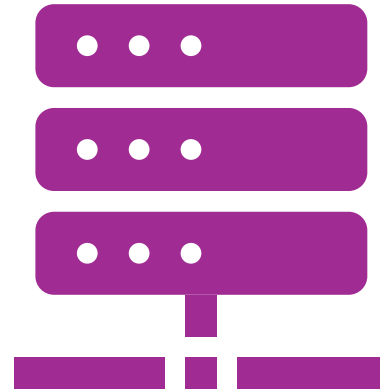
- Availability
- Correctness
- Completeness
- Consistency
- Freshness
- ...

Data



- Availability
- Correctness
- Completeness
- Consistency
- Freshness
- ...

Computing resources



- GPUs
- FLOPS
- Memory
- Parallel processing
- Distributed computing
- ...

Q&A

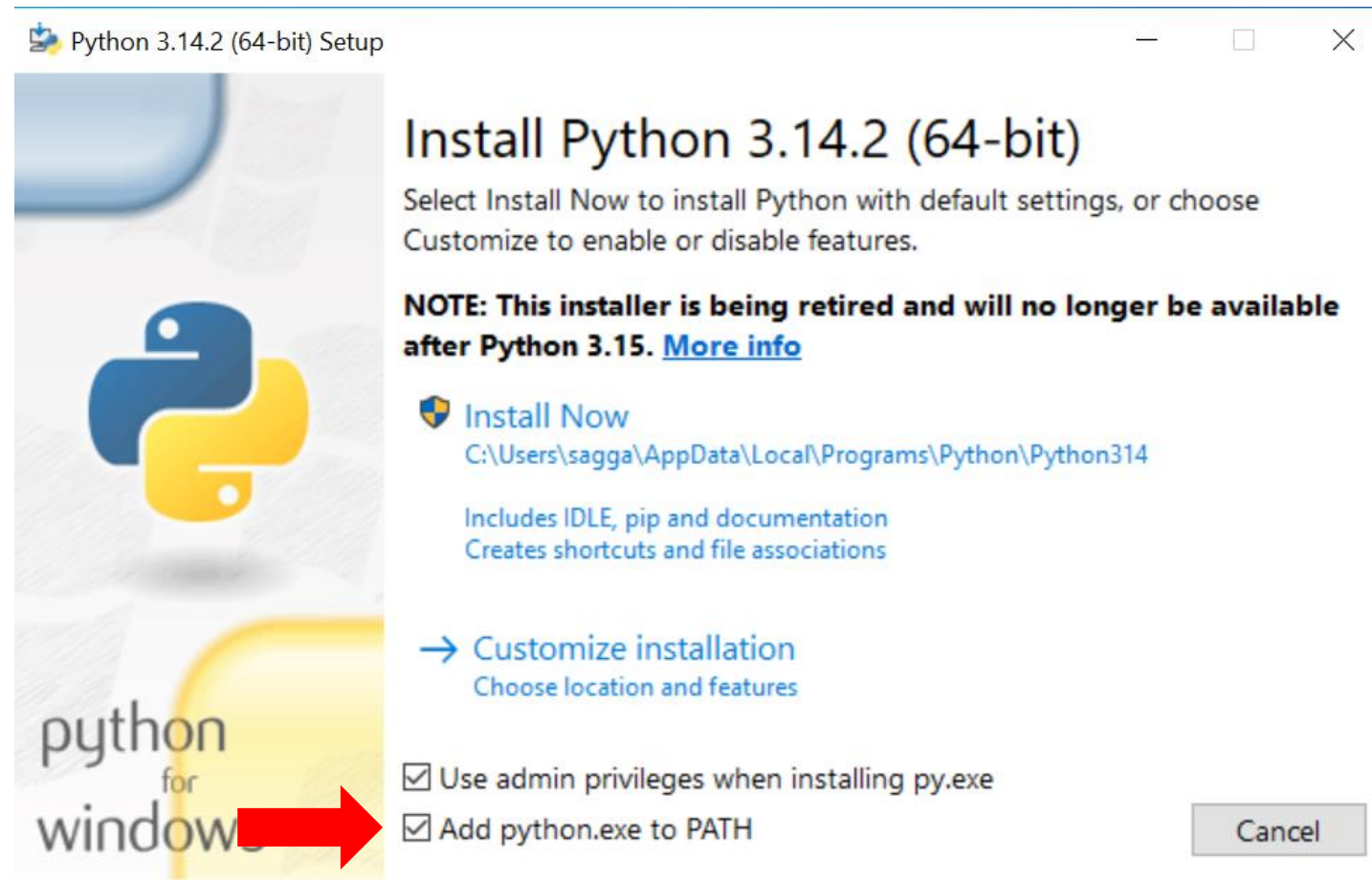


Lab Time

Lab 1: Setting up a Python environment


Step 1

- Go to <https://github.com/ibrahimsaggaf/Introduction-to-Artificial-Intelligence> and download Python 3.14.2 installer.
- Run the .exe file, check the two boxes to add to PATH, and press on install now.
- follow the instructions.



Step 2

- Open a command prompt to verify Python installation by typing:
 1. *python --version*
 2. *pip --version* (Python package manager)

 Command Prompt

```
Microsoft Windows [Version 10.0.19045.6466]  
(c) Microsoft Corporation. All rights reserved.
```

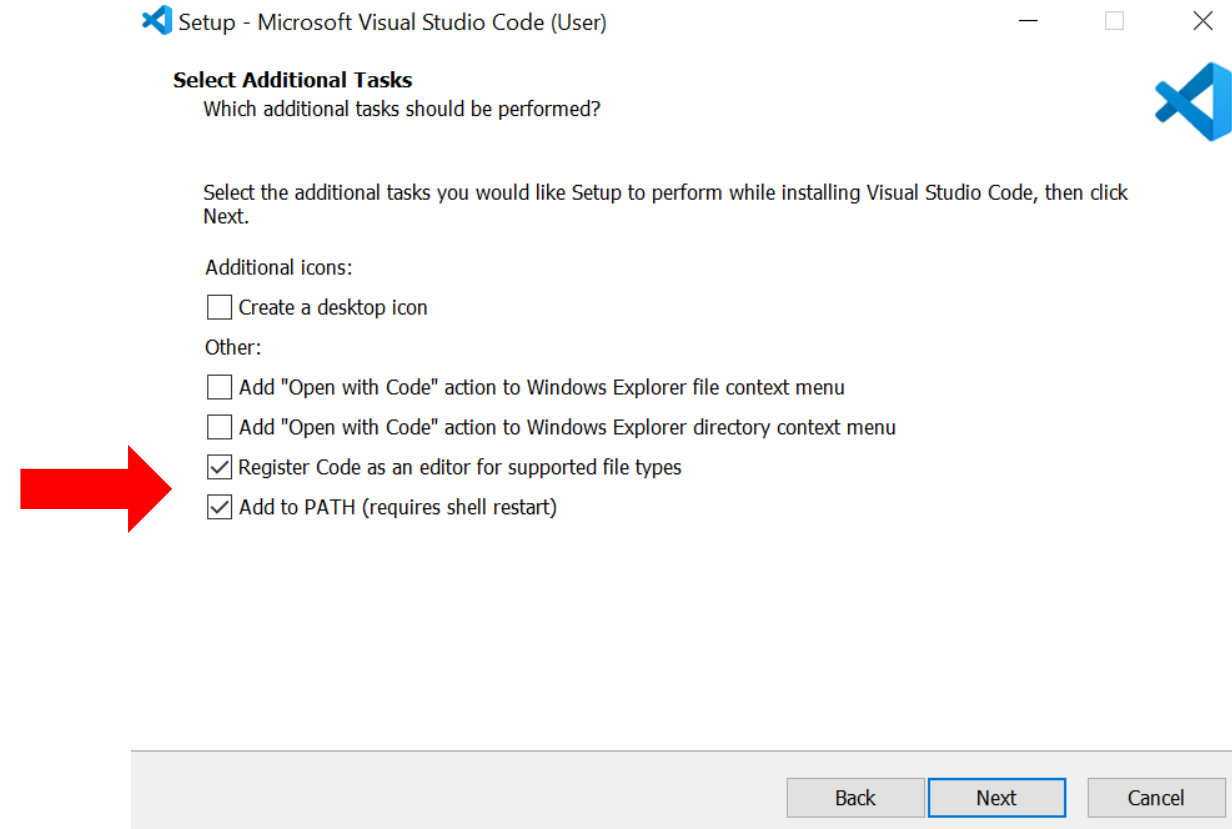
```
C:\Users\sagga>python --version  
Python 3.14.2
```

```
C:\Users\sagga>pip --version  
pip 25.3 from C:\Users\sagga\AppData\Local\Programs\Python\Python314\Lib\site-packages\pip (python 3.14)
```

```
C:\Users\sagga>_
```

Step 3

- Go to <https://code.visualstudio.com/> and download Visual Studio Code installer.
- Run the .exe file and follow the instructions. Make sure to add to PATH by checking the last two boxes as shown below.







Step 4

- Open Visual Studio Code.
- Open a new folder and create a new file named “helloworld.py”
- Type inside the new file the following print statement:
print('Hello world!')
- Open a new terminal window and type the following commands:
 1. *python --version*
 2. *pip --version*
 3. *python helloworld.py*

EXPLORER

▼ LAB1

 helloworld.py `helloworld.py`  helloworld.py

```
1 print('Hello world!')
```

PROBLEMS OUTPUT **TERMINAL** PORTS DEBUG CONSOLE

```
PS C:\Users\sagga\Downloads\lab1> python --version
```

Python 3.14.2

```
PS C:\Users\sagga\Downloads\lab1> pip --version
```

```
pip 25.3 from C:\Users\sagga\AppData\Local\Programs\Python\Python314\Lib\site-packages\pip (python 3.14)
```

```
PS C:\Users\sagga\Downloads\lab1> python helloworld.py
```

Hello world!

```
PS C:\Users\sagga\Downloads\lab1>
```

> OUTLINE

> TIMELINE

Step 5

- Create a virtual environment named “lab1_env” by running the following command in a terminal window:
python -m venv lab1_env
- Activate the virtual environment by running the command:
lab1_env/scripts/activate
- Install the Numpy library inside the virtual environment by running the command:
pip install numpy
- Check the installed libraries inside lab1_env by running the command:
pip freeze
- Deactivate lab1_env by running the command :
deactivate

Step 5

- Run the command *pip freeze* again but outside the virtual environment and examine the output.

A screenshot of the Visual Studio Explorer sidebar. The sidebar is dark-themed. At the top, it says 'EXPLORER'. Below that, there's a folder icon and the text 'LAB1' with a downward arrow. Under 'LAB1', there's a subfolder 'lab1_env' and a file 'helloworld.py' with a Python icon.

The image shows a VS Code interface with a terminal window open. The terminal displays the following commands and output:

```
PS C:\Users\sagga\Downloads\lab1> python -m venv lab1_env
PS C:\Users\sagga\Downloads\lab1> lab1_env/scripts/activate
(lab1_env) PS C:\Users\sagga\Downloads\lab1> pip install numpy
Collecting numpy
  Downloading numpy-2.4.1-cp314-cp314-win_amd64.whl.metadata (6.6 kB)
  Downloading numpy-2.4.1-cp314-cp314-win_amd64.whl (12.4 MB)
    12.4/12.4 MB 14.6 MB/s 0:00:01
Installing collected packages: numpy
Successfully installed numpy-2.4.1
(lab1_env) PS C:\Users\sagga\Downloads\lab1> pip freeze
numpy==2.4.1
(lab1_env) PS C:\Users\sagga\Downloads\lab1> Deactivate lab1_env
PS C:\Users\sagga\Downloads\lab1> pip freeze
PS C:\Users\sagga\Downloads\lab1>
```

The interface includes a sidebar on the left with icons for Explorer, Search, and Run and Debug. The top of the terminal window has tabs for PROBLEMS, OUTPUT, TERMINAL (selected), PORTS, and DEBUG CONSOLE. The top right of the terminal window shows a powershell icon and window controls. The bottom of the interface has a sidebar with icons for Outline and Timeline.

Lab 1: Setting up a Python environment

Congrats! 

[✓] Install Python 3.14.2

[✓] Install Visual Studio Code

[✓] Verify the installation

[✓] Create a virtual environment

Python full course for beginners

<https://www.youtube.com/watch?v=K5KVEU3aaeQ>