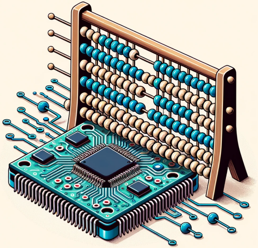
The Role of Machine Learning for Mathematics

What impact can we expect ML to have on theoretical math research?

Christoph Spiegel

Thursday, 11th of October 2023

Zuse Institute Berlin



Three results and three perspectives

1. **Approximation**
Constructions in combinatorics via neural networks [12]

2. **Generalization**
Advancing mathematics by guiding human intuition with AI [5]

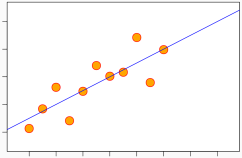
3. **Emergence**
Autoformalization with large language models [13]

4. Pointers for your own research

Approximation

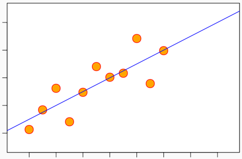
Approximation

Given $A \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$, the map $f : \mathbb{R}^n \to \mathbb{R}^m$, $\mathbf{x} \mapsto A\mathbf{x} + \mathbf{b}$ is an affine linear map.

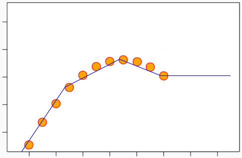


Approximation

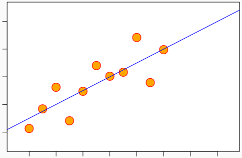
Given $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$, the map $f : \mathbb{R}^n \to \mathbb{R}^m$, $\mathbf{x} \mapsto A\mathbf{x} + \mathbf{b}$ is an affine linear map.



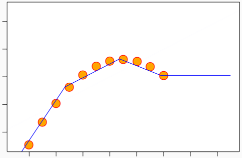
An ℓ-layer perceptron is a map $\mathbf{x} \mapsto f_\ell \circ g \ldots \circ f_2 \circ g \circ f_1(\mathbf{x})$ where $f_1, \ldots, f_\ell$ are affine linear maps and g a *non-linear* activation function.



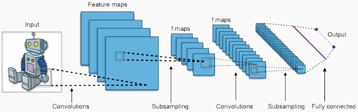
Given $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$, the map $f : \mathbb{R}^n \to \mathbb{R}^m$, $\mathbf{x} \mapsto A\mathbf{x} + \mathbf{b}$ is an affine linear map.



An ℓ-layer perceptron is a map $\mathbf{x} \mapsto f_\ell \circ g \ldots \circ f_2 \circ g \circ f_1(\mathbf{x})$ where $f_1, \ldots, f_\ell$ are affine linear maps and g a *non-linear* activation function.

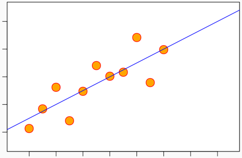


A neural network is a multilayer perceptron that accounts for some structure in the input.

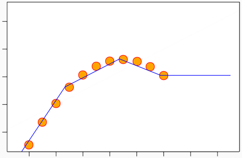


Approximation

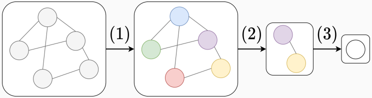
Given $A \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$, the map $f : \mathbb{R}^n \to \mathbb{R}^m$, $\mathbf{x} \mapsto A\mathbf{x} + \mathbf{b}$ is an affine linear map.



An ℓ-layer perceptron is a map $\mathbf{x} \mapsto f_\ell \circ g \ldots \circ f_2 \circ g \circ f_1(\mathbf{x})$ where $f_1, \ldots, f_\ell$ are affine linear maps and g a *non-linear* activation function.

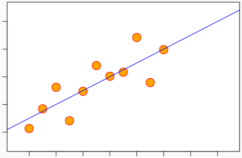


A neural network is a multilayer perceptron that accounts for some structure in the input.

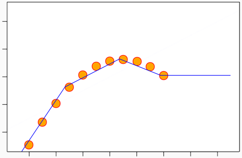


Approximation

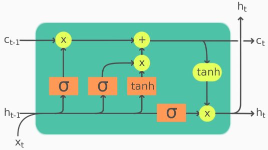
Given $A \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$, the map $f : \mathbb{R}^n \to \mathbb{R}^m$, $\mathbf{x} \mapsto A\mathbf{x} + \mathbf{b}$ is an affine linear map.



An ℓ-layer perceptron is a map $\mathbf{x} \mapsto f_\ell \circ g \ldots \circ f_2 \circ g \circ f_1(\mathbf{x})$ where $f_1, \ldots, f_\ell$ are affine linear maps and g a *non-linear* activation function.



A neural network is a multilayer perceptron that accounts for some structure in the input.



Approximation

Let us assume we are using ReLU activation, that is $g(x) = \max(x, 0)$.

Universal Approximation Theorem

Any continuous function with compact support can be approximated arbitrarily closely by a 2-layer perceptron.

Let us assume we are using ReLU activation, that is $g(x) = \max(x, 0)$.

Universal Approximation Theorem

Any continuous function with compact support can be approximated arbitrarily closely by a 2-layer perceptron.

Under additional assumptions it can also be approximated by a (not necessarily 2-layer) perceptron with width at most dimension of the input.

See Shen et al. [11] for a study of the optimal depth–width–tradeoff.

Approximation

Let us assume we are using ReLU activation, that is $g(x) = \max(x, 0)$.

Universal Approximation Theorem

Any continuous function with compact support can be approximated arbitrarily closely by a 2-layer perceptron.

Under additional assumptions it can also be approximated by a (not necessarily 2-layer) perceptron with width at most dimension of the input.

See Shen et al. [11] for a study of the optimal depth–width–tradeoff.

Question

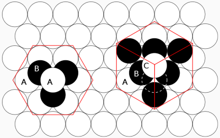
How few layers can a multilayer perceptron have that *exactly* models a specific function, e.g., the maximum of its input? See Hertrich et al. [8]

Approximation

The answer to many mathematical questions is given by one object...

Sphere packing

What is the densest arrangement of congruent spheres in Euclidean space?

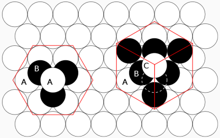


Approximation

The answer to many mathematical questions is given by one object...

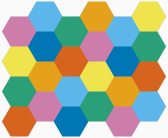
Sphere packing

What is the densest arrangement of congruent spheres in Euclidean space?



Hadwiger Nelson

How many colors do you need such that no two points at unit distance have the same color?

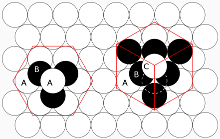


Approximation

The answer to many mathematical questions is given by one object...

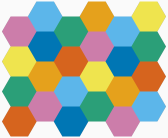
Sphere packing

What is the densest arrangement of congruent spheres in Euclidean space?



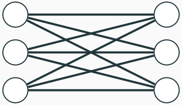
Hadwiger Nelson

How many colors do you need such that no two points at unit distance have the same color?



Mantel's Theorem

What is the maximum edge density of a graph without triangles?

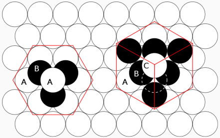


Approximation

The answer to many mathematical questions is given by one object...

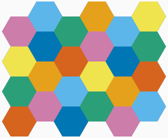
Sphere packing

What is the densest arrangement of congruent spheres in Euclidean space?



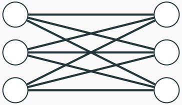
Hadwiger Nelson

How many colors do you need such that no two points at unit distance have the same color?



Mantel's Theorem

What is the maximum edge density of a graph without triangles?



Idea. *Somehow* get a neural network to represent these objects.

Approximation

arXiv > math > arXiv:2104.14516

Mathematics > Combinatorics

[Submitted on 29 Apr 2021]

Constructions in combinatorics via neural networks

Adam Zsolt Wagner

We demonstrate how by using a reinforcement learning algorithm, the deep cross-entropy method, one can find explicit constructions and counterexamples to several open conjectures in extremal combinatorics and graph theory. Amongst the conjectures we refute are a question of Brualdi and Cao about maximizing permanents of pattern avoiding matrices, and several problems related to the adjacency and distance eigenvalues of graphs.

arXiv:2104.14516

Timothy Gowers @wtgowers@mathstodon.xyz
@wtgowers

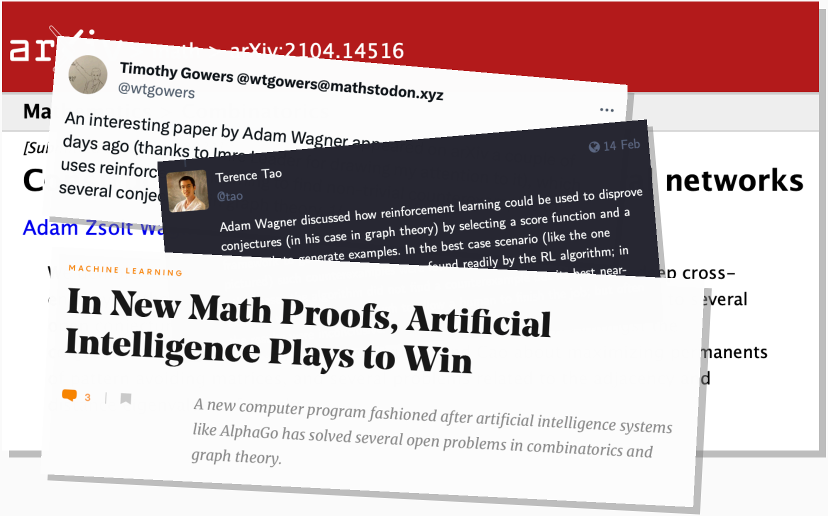
An interesting paper by Adam Wagner appeared on arXiv a couple of days ago (thanks to Imre Leader for drawing my attention to it), which uses reinforcement learning to find non-trivial counterexamples to several conjectures in graph theory. 1/

...ral networks

Adam Zsolt Wagner

We demonstrate how by using a reinforcement learning algorithm, the deep cross-entropy method, one can find explicit constructions and counterexamples to several open conjectures in extremal combinatorics and graph theory. Amongst the conjectures we refute are a question of Brualdi and Cao about maximizing permanents of pattern avoiding matrices, and several problems related to the adjacency and distance eigenvalues of graphs.

Approximation


arXiv:2104.14516
Timothy Gowers @wtgowers@mathstodon.xyz
@wtgowers
An interesting paper by Adam Wagner app
days ago (thanks to Imr
uses reinforc
several conje
Adam Zsolt Wa
networks
14 Feb
Terence Tao
@tao
Adam Wagner discussed how reinforcement learning could be used to disprove
conjectures (in his case in graph theory) by selecting a score function and a
scenario (like the one
found readily by the RL algorithm; in
MACHINE LEARNING
In New Math Proofs, Artificial Intelligence Plays to Win
3
A new computer program fashioned after artificial intelligence systems like AlphaGo has solved several open problems in combinatorics and graph theory.

Approximation

1. Represent discrete objects such as graphs or 0-1-matrices through **binary vectors** and define a **cost function**.

Approximation

1. Represent discrete objects such as graphs or 0-1-matrices through **binary vectors** and define a **cost function**.

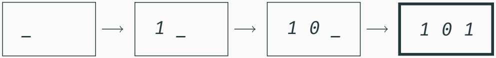
2. Model a distribution over these vectors through a **neural network evaluated in a round-based fashion**.

`_` $\longrightarrow$ `1 _` $\longrightarrow$ `1 0 _` $\longrightarrow$ `1 0 1`

Approximation

1. Represent discrete objects such as graphs or 0-1-matrices through **binary vectors** and define a **cost function**.

2. Model a distribution over these vectors through a **neural network evaluated in a round-based fashion**.

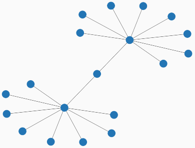


3. Update the parameters of the neural network through **Reinforcement Learning** to incentivise lower cost.

Approximation

Any connected graph on n vertices with matching number μ and largest eigenvalue λ_1 satisfies $\lambda_1 + \mu \geq \sqrt{n-1} + 1$.

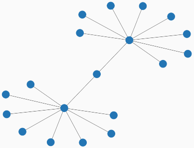
Aouchiche and Hansen [1]



Approximation

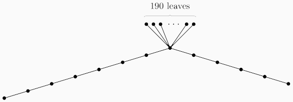
Any connected graph on n vertices with matching number μ and largest eigenvalue λ_1 satisfies $\lambda_1 + \mu \geq \sqrt{n-1} + 1$.

Aouchiche and Hansen [1]



Any connected graph with diameter D, proximity π and distance spectrum $\partial_1 \geq \ldots \geq \partial_n$ satisfies $\pi + \partial_{\lfloor 2D/3 \rfloor} > 0$.

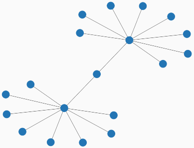
Aouchiche and Hansen [2]



Approximation

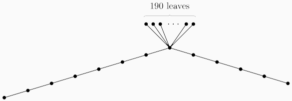
Any connected graph on n vertices with matching number μ and largest eigenvalue λ_1 satisfies $\lambda_1 + \mu \geq \sqrt{n-1} + 1$.

Aouchiche and Hansen [1]



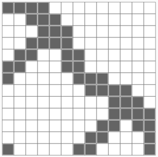
Any connected graph with diameter D, proximity π and distance spectrum $\partial_1 \geq \ldots \geq \partial_n$ satisfies $\pi + \partial_{\lfloor 2D/3 \rfloor} > 0$.

Aouchiche and Hansen [2]



What is the largest permanent of an $n \times n$ 0-1 matrix $A = (a_{i,j})$ that avoids the pattern $a_{i_1,i_3} = a_{i_2,i_1} = a_{i_3,i_2} = 1$?

Brualdi and Cao [3]

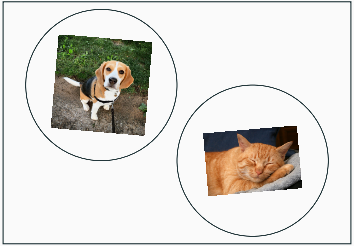


Generalization

Generalization

More commonly, machine learning applications are (or at least used to be) supervised **regression or classification tasks** on datasets:

$\mathbb{R}^{262\,144}$

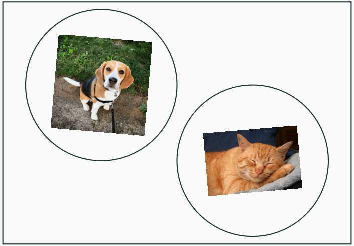


$$\longrightarrow \qquad \{0,1\}$$

Generalization

More commonly, machine learning applications are (or at least used to be) supervised **regression or classification tasks** on datasets:

$\mathbb{R}^{262\,144}$



$\longrightarrow \quad \{0,1\}$

Universal approximation tells us that we can fit *any* dataset, even random noise! Why would this generalize to any actual application?

Generalization

Classical statistics emphasizes simple models, guarantees, assumptions about the data, and interpretability of the output.

Generalization

Classical statistics emphasizes simple models, guarantees, assumptions about the data, and interpretability of the output.

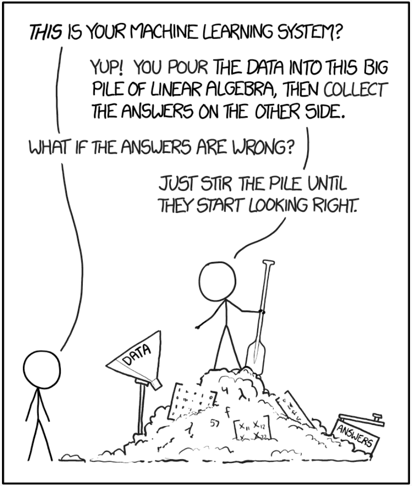
Big Data relies on overparameterization, offers little formal guarantees or interpretability, and accepts behavior like adversarial examples.

Generalization

Classical statistics emphasizes simple models, guarantees, assumptions about the data, and interpretability of the output.

Big Data relies on overparameterization, offers little formal guarantees or interpretability, and accepts behavior like adversarial examples.

Generalization is achieved through many practical tools, but the current state of machine learning has been likened to **alchemy**. Ali Rahimi at NeurIPS 2017



xkcd.com

Generalization

nature

Advancing mathematics by guiding human intuition with AI

Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, Marc Lackenby, Geordie Williamson, Demis Hassabis & Pushmeet Kohli

Nature **600**, 70–74 (2021) | Cite this article

256k Accesses | **102** Citations | **1607** Altmetric | Metrics

Two collaborations from Google DeepMind with András Juhász and Marc Lackenby (knot theory) and Geordie Williamson (representation theory).

nature

Quanta magazine

MACHINE LEARNING

Machine Learning Becomes a Mathematical Collaborator

By KELSEY HOUSTON-EDWARDS

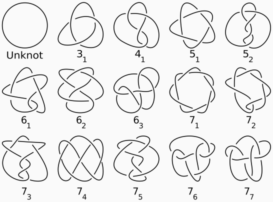
February 15, 2022

Two recent collaborations between mathematicians and DeepMind demonstrate the potential of machine learning to help researchers generate new mathematical conjectures.

Two collaborations from Google DeepMind with András Juhász and Marc Lackenby (knot theory) and Geordie Williamson (representation theory).

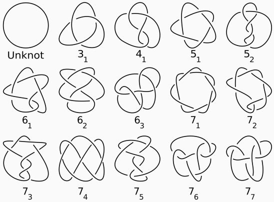
Generalization

A knot is an embedding of the S^1 into $\mathbb{R}^3$. Two knots equivalent if they can continuously be deformed into each other. An invariant is a function on equivalence classes of knots.



Generalization

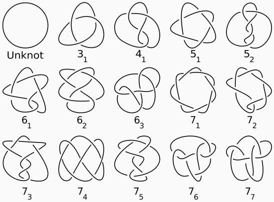
A knot is an embedding of the S^1 into $\mathbb{R}^3$. Two knots equivalent if they can continuously be deformed into each other. An invariant is a function on equivalence classes of knots.



Juhász and Lackenby believed in an undiscovered relation between **geometric** and **algebraic** varieties.

Generalization

A knot is an embedding of the S^1 into $\mathbb{R}^3$. Two knots equivalent if they can continuously be deformed into each other. An invariant is a function on equivalence classes of knots.

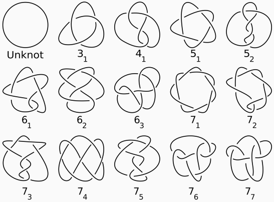


Juhász and Lackenby believed in an undiscovered relation between **geometric** and **algebraic** varieties.

geometric varieties		**algebraic varieties**
Meridional translation *Longitudinal translation* *Short geodesic* *Injectivity radius* ...	$\longrightarrow$	*Signature* *Jones polynomial* ...

Generalization

A knot is an embedding of the S^1 into $\mathbb{R}^3$. Two knots equivalent if they can continuously be deformed into each other. An invariant is a function on equivalence classes of knots.



Juhász and Lackenby believed in an undiscovered relation between **geometric** and **algebraic** varieties.

geometric varieties

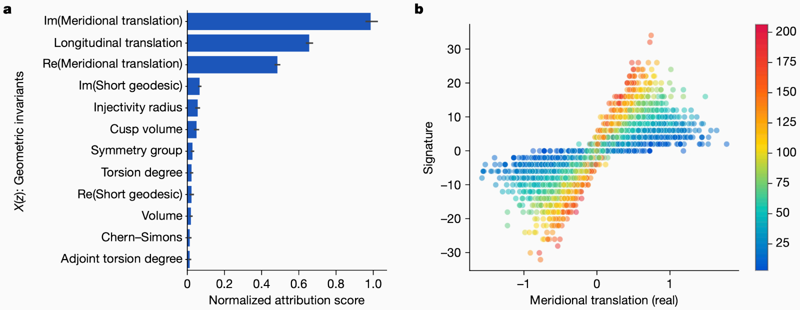
Meridional translation
Longitudinal translation
Short geodesic
Injectivity radius
...

Neural Network $\longrightarrow$

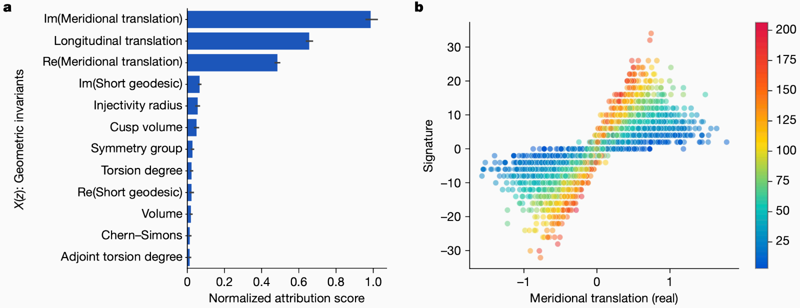
algebraic varieties

Signature
Jones polynomial
...

Generalization

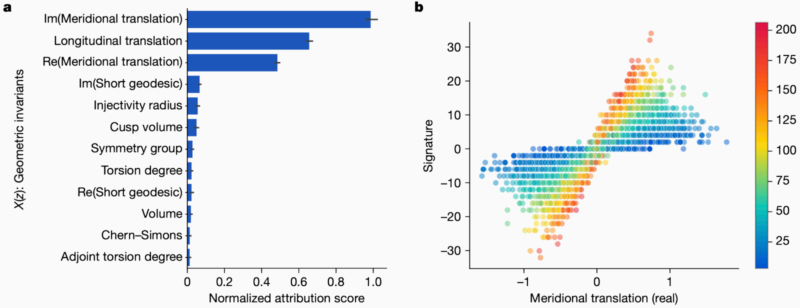


Generalization



Based on this, it was conjectured that

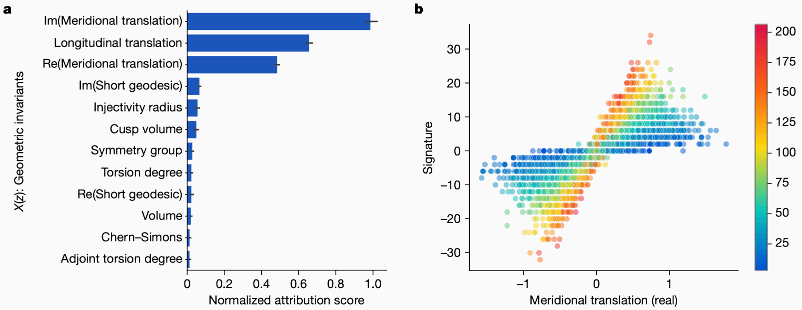
$$\left|\, 2\sigma(K) - \mathrm{Re}\left(\lambda(K)/\mu(K)\right)\right| < c_1\, \mathrm{vol}(K) + c_2$$



Based on this, it was conjectured that

$$\cancel{\left|2\sigma(K) - \mathrm{Re}\left(\lambda(K)/\mu(K)\right)\right| < c_1 \,\mathrm{vol}(K) + c_2}$$

Generalization



Based on this, it was conjectured that

$$\cancel{\left|2\sigma(K) - \mathrm{Re}\left(\lambda(K)/\mu(K)\right)\right| < c_1 \operatorname{vol}(K) + c_2}$$

$$\left|2\sigma(K) - \mathrm{Re}\left(\lambda(K)/\mu(K)\right)\right| < c \operatorname{vol}(K) \operatorname{inj}(K)^{-3}$$

which was turned into a theorem in a separate publication [4]!

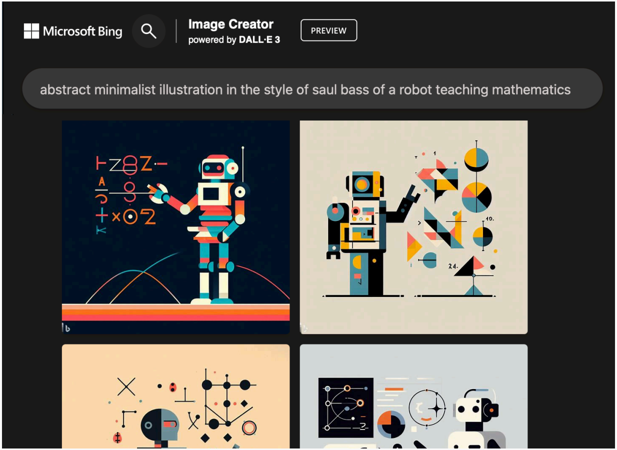
Emergence

Emergence

Over the last few years we have increasingly seen systems transcending big data with capabilities that verge on emergence.

Emergence

Over the last few years we have increasingly seen systems transcending big data with capabilities that verge on emergence.



Emergence

Over the last few years we have increasingly seen systems transcending big data with capabilities that verge on emergence.

CH

What would be an easy way of having a gitlab (privately hosted at git.zib.de) CI/CD workflow send messages to a particular channel on Slack?

You can use Slack's Incoming Webhooks to send messages from your GitLab CI/CD pipeline to a Slack channel. Here's a quick outline:

1. Create an Incoming Webhook in Slack and get the Webhook URL.
2. Add the Webhook URL as a secret variable in your GitLab project settings.
3. Update your `.gitlab-ci.yml` to send a message using `curl`.

Example `.gitlab-ci.yml` snippet:

yaml　Copy code

```yaml
notify_slack:
  script:
    - curl -X POST -H 'Content-type: application/json' --data '{"text":"Hell
```

Replace `$SLACK_WEBHOOK_URL` with the secret variable you added in GitLab.

Emergence

Why has theoretical math not been affected by this (yet)?

Emergence

Why has theoretical math not been affected by this (yet)?

Large Language Models (LLMs) have been called stochastic parrots that do not transcend the quality of their input.

Emergence

Why has theoretical math not been affected by this (yet)?

Large Language Models (LLMs) have been called stochastic parrots that do not transcend the quality of their input. Unlike (some) humans, they do not internally reason before answering.

Emergence

Why has theoretical math not been affected by this (yet)?

Large Language Models (LLMs) have been called stochastic parrots that do not transcend the quality of their input. Unlike (some) humans, they do not internally reason before answering.

Computation has seemingly gone from *precise output when given precise input* to accepting *fuzzy input at the cost of fuzzy output*.

Emergence

Why has theoretical math not been affected by this (yet)?

Large Language Models (LLMs) have been called stochastic parrots that do not transcend the quality of their input. Unlike (some) humans, they do not internally reason before answering.

Computation has seemingly gone from *precise output when given precise input* to accepting *fuzzy input at the cost of fuzzy output*.

But this story is not entirely true ...

Emergence

NeurIPS Proceedings

Autoformalization with Large Language Models

Part of Advances in Neural Information Processing Systems 35 (NeurIPS 2022) Main Conference Track

Bibtex Paper Supplemental

Authors

Yuhuai Wu, Albert Qiaochu Jiang, Wenda Li, Markus Rabe, Charles Staats, Mateja Jamnik, Christian Szegedy

Emergence

Case Study 2 Question:
Natural Language version: "*When all the girls at Madeline's school line up in rows of eight, there are seven left over. If instead they line up in rows of four, how many are left over? The final answer is 3.*" Translate the natural language version to an Isabelle version:

PaLM output:
```
theorem
    fixes n::nat
    assumes "n mod 8 = 7"
    shows "n mod 4 = 3"
```

Case Study 3 Question:
Natural language version: *"Let f be a linear function for which $f(6) - f(2) = 12$. What is $f(12) - f(2)$? The final answer is 30."* Translate the natural language version to an Isabelle version:

Codex output:
```
theorem
    fixes f :: "real \<Rightarrow> real"
    assumes "linear f"
            "f 6 - f 2 = 12"
    shows "f 12 - f 2 = 30"
```

Figure 3: Autoformalizations from natural language to Isabelle code. **Left:** Case study 2 – perfect formalization by PaLM. **Right:** Case study 3 – incorrect formalization by Codex.

Machine learning approaches to directly create proofs in formal language exist [10] and are improving along with LLM advances …

Pointers for your own research

Pointers for your own research

- Implementing a machine learning approach is often **hard** for entirely different reasons than mathematics is hard and can require a lot of resources and time.

Pointers for your own research

- Implementing a machine learning approach is often **hard** for entirely different reasons than mathematics is hard and can require a lot of resources and time.
- You should know in advance if you care about the application or the method and if you want a general purpose or a problem-specific approach.

Pointers for your own research

- Implementing a machine learning approach is often **hard** for entirely different reasons than mathematics is hard and can require a lot of resources and time.
- You should know in advance if you care about the application or the method and if you want a general purpose or a problem-specific approach.
- If you end up publicizing your method, be aware of existing approaches beyond Machine Learning:

Pointers for your own research

- Implementing a machine learning approach is often **hard** for entirely different reasons than mathematics is hard and can require a lot of resources and time.
- You should know in advance if you care about the application or the method and if you want a general purpose or a problem-specific approach.
- If you end up publicizing your method, be aware of existing approaches beyond Machine Learning:
  - Discrete black-box optimization has been studied since the 60s with effective approaches like Simmulated Annealing [9]. They have been extensively used to find objects like Ramsey colorings.

Pointers for your own research

- Implementing a machine learning approach is often **hard** for entirely different reasons than mathematics is hard and can require a lot of resources and time.
- You should know in advance if you care about the application or the method and if you want a general purpose or a problem-specific approach.
- If you end up publicizing your method, be aware of existing approaches beyond Machine Learning:
    - Discrete black-box optimization has been studied since the 60s with effective approaches like Simmulated Annealing [9]. They have been extensively used to find objects like Ramsey colorings.
    - Graffiti [6] or AutoGraphiX [7] have been formulating and refuting conjectures in extremal graph theory since the 80s.

References

[1] M. Aouchiche and P. Hansen.
A survey of automated conjectures in spectral graph theory.
Linear algebra and its applications, 432(9):2293–2322, 2010.

[2] M. Aouchiche and P. Hansen.
Proximity, remoteness and distance eigenvalues of a graph.
Discrete Applied Mathematics, 213:17–25, 2016.

[3] R. A. Brualdi and L. Cao.
Pattern-avoiding (0, 1)-matrices.
arXiv preprint arXiv:2005.00379, 2020.

[4] A. Davies, A. Juhász, M. Lackenby, and N. Tomasev.
The signature and cusp geometry of hyperbolic knots.
arXiv preprint arXiv:2111.15323, 2021.

References

[5] A. Davies, P. Veličković, L. Buesing, S. Blackwell, D. Zheng, N. Tomašev, R. Tanburn, P. Battaglia, C. Blundell, A. Juhász, et al.
Advancing mathematics by guiding human intuition with ai.
Nature, 600(7887):70–74, 2021.

[6] S. Fajtlowicz.
On conjectures of graffiti.
In *Annals of Discrete Mathematics*, volume 38, pages 113–118. Elsevier, 1988.

[7] P. Hansen and G. Caporossi.
Autographix: An automated system for finding conjectures in graph theory.
Electronic Notes in Discrete Mathematics, 5:158–161, 2000.

References

[8] C. Hertrich, A. Basu, M. Di Summa, and M. Skutella.
Towards lower bounds on the depth of relu neural networks.
Advances in Neural Information Processing Systems, 34:3336–3348, 2021.

[9] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi.
Optimization by simulated annealing.
science, 220(4598):671–680, 1983.

[10] S. Polu and I. Sutskever.
Generative language modeling for automated theorem proving.
arXiv preprint arXiv:2009.03393, 2020.

References

[11] Z. Shen, H. Yang, and S. Zhang.
Optimal approximation rate of relu networks in terms of width and depth.
Journal de Mathématiques Pures et Appliquées, 157:101–135, 2022.

[12] A. Z. Wagner.
Constructions in combinatorics via neural networks.
arXiv preprint arXiv:2104.14516, 2021.

[13] Y. Wu, A. Q. Jiang, W. Li, M. Rabe, C. Staats, M. Jamnik, and C. Szegedy.
Autoformalization with large language models.
Advances in Neural Information Processing Systems, 35:32353–32368, 2022.