

AI Generated Classes

Iddo Drori

Associate Professor of Computer Science

BU (practice), MIT (visiting), Columbia University (adj)

AI Generated

Homeworks
2021

Exams
Spring 2022

Curriculum
2022

Books
2023

Lectures
Spring 2023

Classes
Fall 2023



A neural network solves, explains, and generates university math problems by program synthesis and few-shot learning at human level

[Iddo Drori](#)  , [Sarah Zhang](#), [Reece Shuttlesworth](#) , [Leonard Tang](#), [Albert Lu](#) , [Elizabeth Ke](#), [Kevin Liu](#), [Linda Chen](#), [Sunny Tran](#) , [Newman Cheng](#) , [Roman Wang](#) , [Nikhil Singh](#) , [Taylor L. Patti](#), [Jayson Lynch](#), [Avi Shporer](#) , [Nakul Verma](#), [Eugene Wu](#), and [Gilbert Strang](#)   [-14](#) [Authors Info & Affiliations](#)

Edited by Jeffrey Ullman, Stanford University (Retired), Stanford, CA; received January 3, 2022; accepted June 13, 2022

August 2, 2022 | 119 (32) e2123433119 | <https://doi.org/10.1073/pnas.2123433119>

 24,859



PNAS

Vol. 119 | No. 32

Significance

Abstract

Data Availability

Supporting Information

References

Significance

We demonstrate that a neural network automatically solves, explains, and generates university-level problems from the largest Massachusetts Institute of Technology (MIT) mathematics courses at a human level. Our methods combine three innovations: 1) using recent neural networks pretrained on text and fine-tuned on code rather than pretrained on text; 2) few-shot learning synthesizing programs that correctly solve course problems automatically; and 3) a pipeline to solve questions, explain solutions, and generate new questions indistinguishable by students from course questions. Our work solves university-level mathematics courses and improves upon state-of-the-art, increasing automatic accuracy on randomly sampled questions on a benchmark by order of magnitude. Implications for higher education include roles of artificial intelligence (AI) in automated course evaluation and content generation.

dax



wif



lug



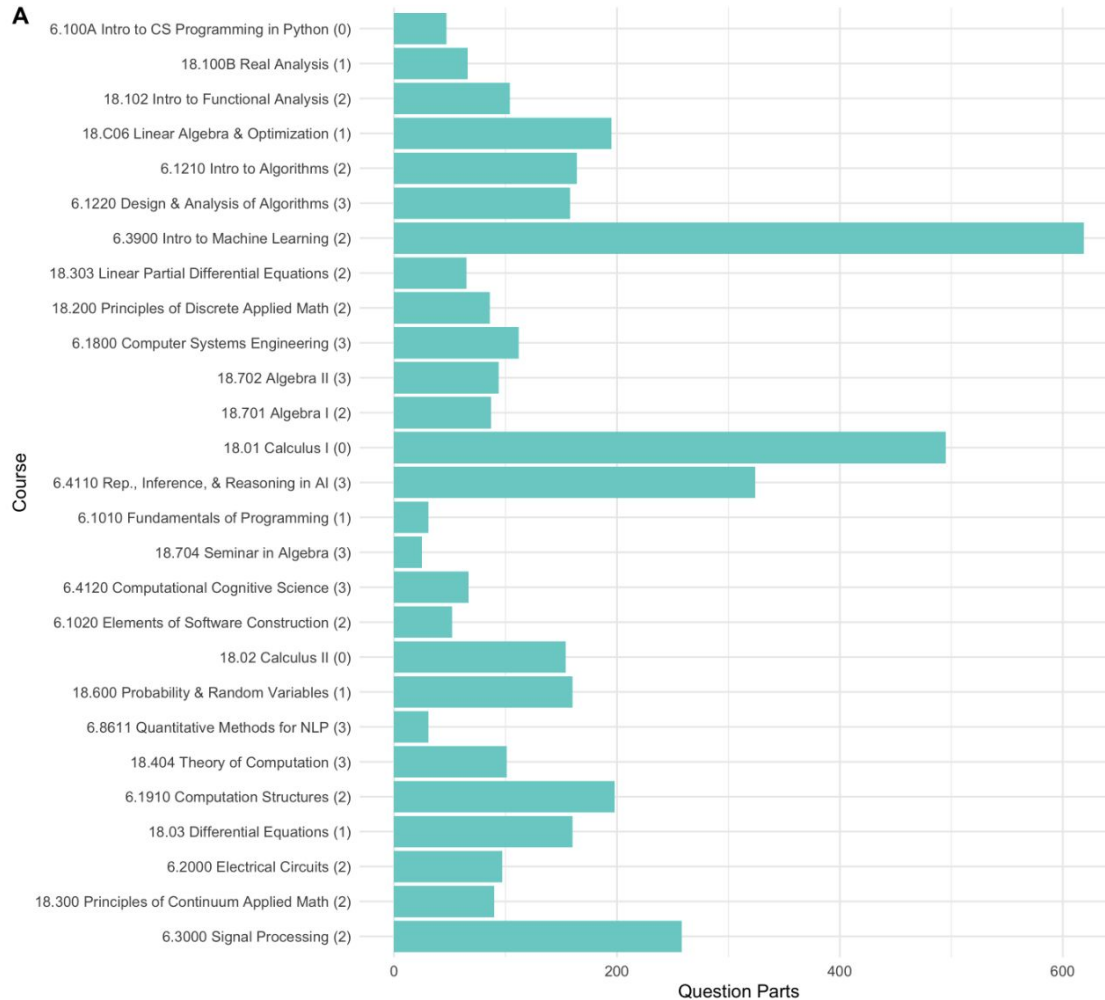
zup



What questions does a student need
to answer to fulfill the graduation
requirements of MIT in Mathematics
and EECS in any major?

ID	Number	Name	Level	Questions	Parts
1	6.100A	Intro to CS Programming in Python	0	34	47
2	18.100B	Real Analysis	1	60	66
3	18.102	Intro to Functional Analysis	2	68	104
4	18.C06	Linear Algebra & Optimization	1	77	195
5	6.1210	Intro to Algorithms	2	82	164
6	6.1220	Design & Analysis of Algorithms	3	44	158
7	6.3900	Intro to Machine Learning	2	114	619
8	18.303	Linear Partial Differential Equations	2	22	65
9	18.200	Principles of Discrete Applied Math	2	45	86
10	6.1800	Computer Systems Engineering	3	58	112
11	18.702	Algebra II	3	52	94
12	18.701	Algebra I	2	58	87
13	18.01	Calculus I	0	203	495
14	6.4110	Rep., Inference, & Reasoning in AI	3	54	324
15	6.1010	Fundamentals of Programming	1	22	31
16	18.704	Seminar in Algebra	3	16	25
17	6.4120	Computational Cognitive Science	3	10	67
18	6.1020	Elements of Software Construction	2	26	52
19	18.02	Calculus II	0	81	154
20	18.600	Probability & Random Variables	1	65	160
21	6.8611	Quantitative Methods for NLP	3	20	31
22	18.404	Theory of Computation	3	53	101
23	6.1910	Computation Structures	2	72	198
24	18.03	Differential Equations	1	66	160
25	6.2000	Electrical Circuits	2	27	97
26	18.300	Principles of Continuum Applied Math	2	43	90
27	6.3000	Signal Processing	2	55	258
28	6.2300	Electromagnetic Waves & Applications	2	37	142
29	6.3010	Signals, Systems & Inference	3	57	224
30	18.901	Intro to Topology	2	58	144
Mean				55.97	151.67
Total				1679	4550

A



ID	Task Type	Parts
1	Exercise	198
2	Problem Set	2820
3	Final Exam	418
4	Midterm Exam	799
5	Lab	278
6	Project	37
Mean		758.3
Total		4550

ID	Answer Type	Parts
1	Programming	234
2	Multiple Choice	710
3	Numerical	634
4	Expression	969
Submean		636.8
Subtotal		2547
5	Open	1821
6	Image	182
Submean		1001.5
Subtotal		2003
Mean		758.3
Total		4550

lug kiki wif



dax kiki lug



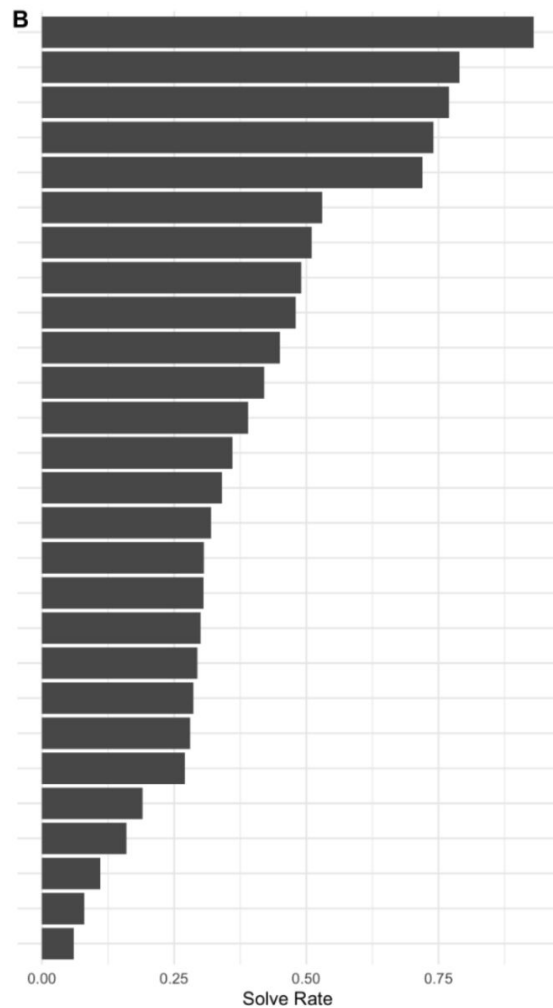
Can a language model fulfill the
graduation requirements?

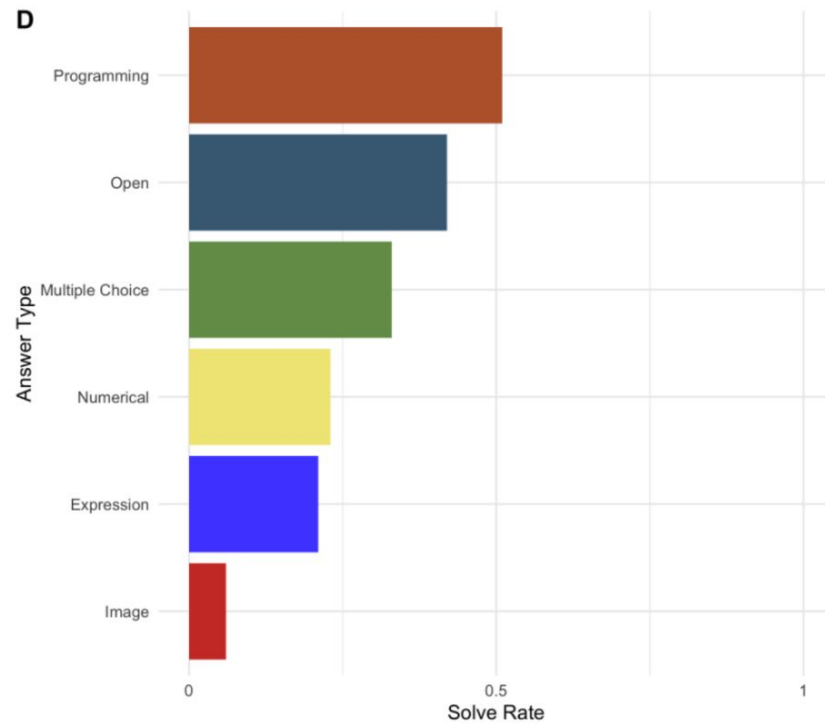
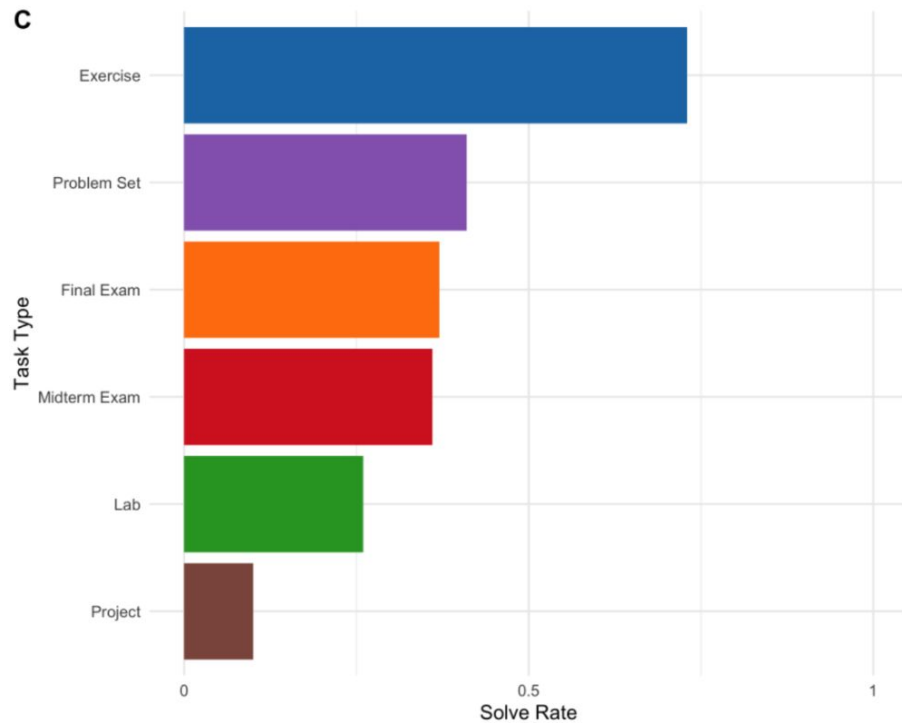
as is $1/3$

A



B





ID	Number	Name	Level	Questions	Parts	Task Type Solve Rate	Answer Type Solve Rate	Solve Rate
1	6.100A	Intro to CS Programming in Python	0	34	47	1 (E), 1 (ME), 0.84 (PS)	0.93 (P)	0.93
2	18.100B	Real Analysis	1	60	66	0.90 (FE), 0.67 (ME), 0.79 (PS)	0 (I), 0.80 (O)	0.79
3	18.102	Intro to Functional Analysis	2	68	104	0.73 (FE), 0.71 (ME), 0.94 (PS)	0.77 (O)	0.77
4	18.C06	Linear Algebra & Optimization	1	77	195	0.81 (FE), 0.76 (ME), 0.67 (PR), 0.71 (PS)	0.72 (E), 0 (I), 1 (M), 0.84 (N), 0.78 (O), 0.71 (P)	0.74
5	6.1210	Intro to Algorithms	2	82	164	0.78 (FE), 0.72 (ME), 0.66 (PS)	0.21(E), 1 (I), 0.56 (M), 0 (N), 0.83 (O), 0.53 (P)	0.72
6	6.1220	Design & Analysis of Algorithms	3	44	158	0.56 (FE), 0.51 (ME), 0.54 (PS)	0.65 (E), 0.50 (M), 0.34 (N), 0.53 (O)	0.53
7	6.3900	Intro to Machine Learning	2	114	619	0.35 (E), 0.38 (FE), 0.68 (L), 0.57 (ME), 0.63 (PS)	0.52 (E), 0.07 (I), 0.35 (M), 0.26 (N), 0.66 (O), 0.46 (P)	0.51
8	18.303	Linear Partial Differential Equations	2	22	65	0.08 (ME), 0.5 (PR), 0.71 (PS)	0.37 (E), 1 (I), 1 (N), 0.47 (O)	0.49
9	18.200	Principles of Discrete Applied Math	2	45	86	0.45 (ME), 0.50 (PS)	0.07 (E), 0 (I), 0.28 (N), 0.67 (O)	0.48
10	6.1800	Computer Systems Engineering	3	58	112	0.51 (ME), 0 (PR), 0.59 (PS)	0 (E), 0.54 (M), 0.25 (N), 0.07 (O)	0.45
11	18.702	Algebra II	3	52	94	0.58 (ME), 0.36 (PS)	0.23 (E), 0.49 (M), 0.76 (N), 0.43 (O)	0.42
12	18.701	Algebra I	2	58	87	0.37 (ME), 0.46 (PS)	0.39 (O)	0.39
13	18.01	Calculus I	0	203	495	0.28 (FE), 0.37 (ME), 0.45 (PS)	0.44 (E), 0.07 (I), 0.25 (M), 0.27 (N), 0.57 (O)	0.36
14	6.4110	Rep., Inference, & Reasoning in AI	3	54	324	0.23 (FE), 0.40 (ME), 0.37 (PS)	0.26 (E), 0.33 (M), 0.16 (N), 0.42 (O), 0.52 (P)	0.34
15	6.1010	Fundamentals of Programming	1	22	31	0.38 (L), 0.26 (ME)	0.56 (E), 0.17 (M), 0 (N), 0.41 (O), 0.35 (P)	0.32
16	18.704	Seminar in Algebra	3	16	25	0 (PR), 0.61 (PS)	0.31 (O)	0.31
17	6.4120	Computational Cognitive Science	3	10	67	0 (PR), 0.19 (PS)	0 (E), 0.08 (O)	0.31
18	6.1020	Elements of Software Construction	2	26	52	0.27 (ME), 0 (PR), 0.35 (PS)	0.75 (E), 0.19 (M), 0.18 (O), 0.32 (P)	0.30
19	18.02	Calculus II	0	81	154	0.13 (FE), 0.38 (ME), 0.35 (PS)	0.23 (E), 0.31 (I), 0 (M), 0.20 (N), 0.57 (O)	0.29
20	18.600	Probability & Random Variables	1	65	160	0.38 (FE), 0.13 (ME), 0.41 (PS)	0.22 (E), 0.21 (N), 0.62 (O)	0.29
21	6.8611	Quantitative Methods for NLP	3	20	31	0.02 (PR), 0.52 (PS)	0 (I), 0.30 (O)	0.28
22	18.404	Theory of Computation	3	53	101	0.31 (FE), 0.12 (ME), 0.31 (PS)	0 (E), 0 (I), 0.46 (M), 0.27 (O)	0.27
23	6.1910	Computation Structures	2	72	198	0.40 (E), 0.13 (ME), 0.04 (L)	0.17 (E), 0 (I), 0.55 (M), 0.17 (N), 0.10 (O), 0.06 (P)	0.19
24	18.03	Differential Equations	1	66	160	0.05 (FE), 0.14 (ME), 0.29 (PS)	0.11 (E), 0.21 (I), 0.27 (N), 0.29 (O)	0.16
25	6.2000	Electrical Circuits	2	27	97	0.02 (FE), 0.04 (ME), 0.29 (PS)	0.02 (E), 0 (I), 0.38 (N), 0.37 (O)	0.11
26	18.300	Principles of Continuum Applied Math	2	43	90	0 (PR), 0.12 (PS)	0.07 (E), 0 (I), 0 (N), 0.12 (O)	0.08
27	6.3000	Signal Processing	2	55	258	0.05 (FE), 0.01 (ME), 0.13 (PS)	0.13 (E), 0.02 (I), 0 (M), 0.06 (N), 0.22 (O)	0.06
28	6.2300	Electromagnetic Waves & Applications	2	37	142			
29	6.3010	Signals, Systems & Inference	3	57	224			
30	18.901	Intro to Topology	2	58	144			
Mean				55.97	151.67			
Total				1679	4550			

lug fep



lug blicket wif



dax fep



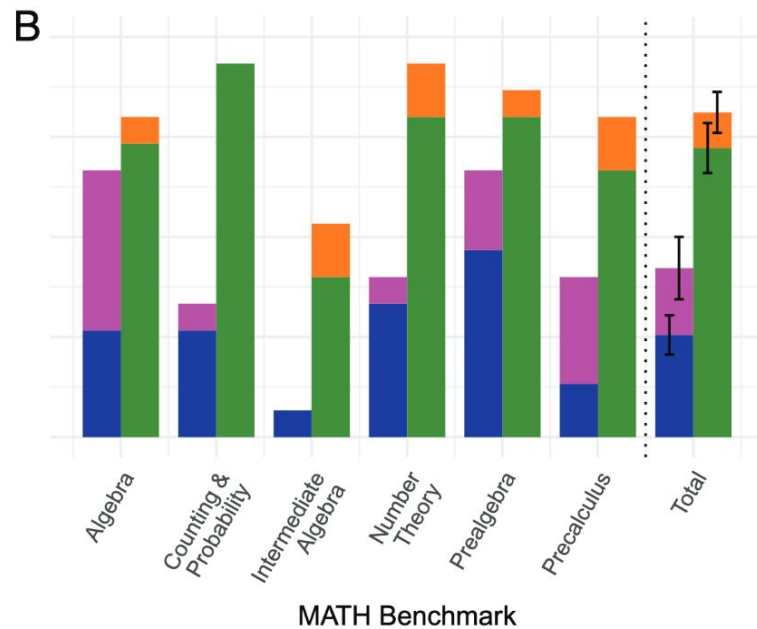
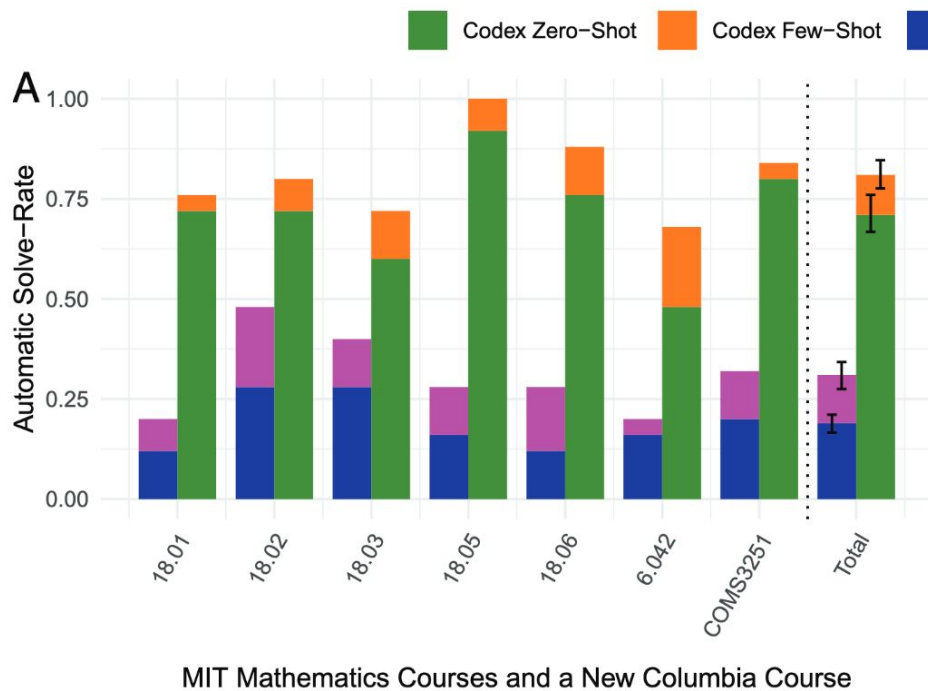
wif blicket dax



How can we ensure that a language
model generates correct and
complete answers?

Correctness and Completeness

1. Provide correct and clear text in prompt
2. Simplification
3. Chain of thought
4. Automatic checkers for several types of questions
5. Training a classifier predicting if model can answer question
6. Self error correction



Source: Drori et al, PNAS, 2022

Correctness and Completeness

MIT class data

Books, notes, exercises, problem sets, midterms, finals, projects

MIT data

Slack, email

lug fep kiki wif ●●●●

wif kiki dax blicket lug ●●●●

lug kiki wif fep ●●●●

wif blicket dax kiki lug ●●●●

Which learning approaches allow
language models to pass the MIT
curriculum and why?

Current Approach

Akin to asking a sophomore on the first day of orientation to solve the entire curriculum one question at a time, without learning anything, without scratch paper, and with a single thought for each answer.

Learning Approaches

1. Few-shot learning
2. Chain of thought
3. Program synthesis
4. Self-error correction

Few-Shot Learning

Asking a large language model questions

Problem: is analogous to asking a human a question without learning the subject

Solution: few-shot learning allows to provide other question-answer or text examples before question

lug kiki wif blicket zup ●●●●●

What are the most challenging
concepts, tasks, and types of questions?

Challenging Concepts

Proofs

Challenging Types of Questions

Images

ID	Task Type	Parts	Solve Rate
1	Exercise	198	0.73
2	Problem Set	2820	0.41
3	Final Exam	418	0.37
4	Midterm Exam	799	0.36
5	Lab	278	0.26
6	Project	37	0.10
Mean		758.3	0.37
Total		4550	0.36

ID	Answer Type	Parts	Solve Rate
1	Programming	234	0.51
2	Multiple Choice	710	0.33
3	Numerical	634	0.23
4	Expression	969	0.21
Submean		636.8	0.32
Subtotal		2547	0.31
5	Open	1821	0.42
6	Image	182	0.06
Submean		1001.5	0.24
Subtotal		2003	0.40
Mean		758.3	0.29
Total		4550	0.36

zup blicket zup kiki zup fep ● ● ● ● ● ●

What questions, topics, and classes help answer other questions, understand other topics, and are prerequisites for other classes based on data?

Curriculum Design

Embed questions in low-dimensional space and show the relationships between questions, topics, and classes providing insight into course prerequisites and curriculum design based on data.

Curriculum Analysis and Generation from Data

1. Embedding of topics and questions
2. Graph of classes: nodes are classes, directed edges are ability to few shot one class given the other

Curriculum Analysis and Generation from Data (2022)

Prerequisites based on data

Equivalent classes based on data

Set of classes and concepts required for graduating

zup blicket wif kiki dax fep



How can we evaluate student performance while allowing the usage of large language models?

Evaluation

1. Student evaluate language model correctness
2. Asking questions language model cannot solve yet
3. Labs with checkoffs
4. Projects
5. Clearly mark which parts are student contribution and which parts are language model contribution

Homeworks and Exams embracing ChatGPT (Spring 2023)

Following are 10 questions and answers by ChatGPT. For each question identify whether the answer is correct or not. If the answer is correct, explain why. If the answer is incorrect, write the correct answer and explain.

Q2

1 Point

What is the difference between a learning algorithm and a hypothesis? Write down one possible hypothesis for a linear regression problem in which the input dimension is $d = 3$

ChatGPT Answer: "A learning algorithm is a procedure or set of instructions used to find an approximate solution to a problem, such as finding a hypothesis that fits a given dataset. A hypothesis, on the other hand, is a statement or function that describes the relationship between inputs and outputs in a problem.

One possible hypothesis for a linear regression problem with input dimension $d = 3$ is a linear function of the form $h(x) = w_1x_1 + w_2x_2 + w_3x_3 + b$, where w_1, w_2, w_3 and b are parameters to be learned from the data."

☒ Correct

☐ Incorrect

Explanation:

zup blicket wif kiki dax fep 

How can we automatically generate
questions at scale with correct answers?

Question Generation

1. Generate question and answer and explanation (or verification) together at the same time
2. Generate questions from other questions
3. Generate questions from specific parts of course notes or books

Machine-Generated Questions, Answers, Explanations

Indistinguishable from human-written questions

Appropriate for class, controlled difficulty level, engaging

Research focus on correctness, completeness, quality, originality: quantified

Used in class homeworks and finals

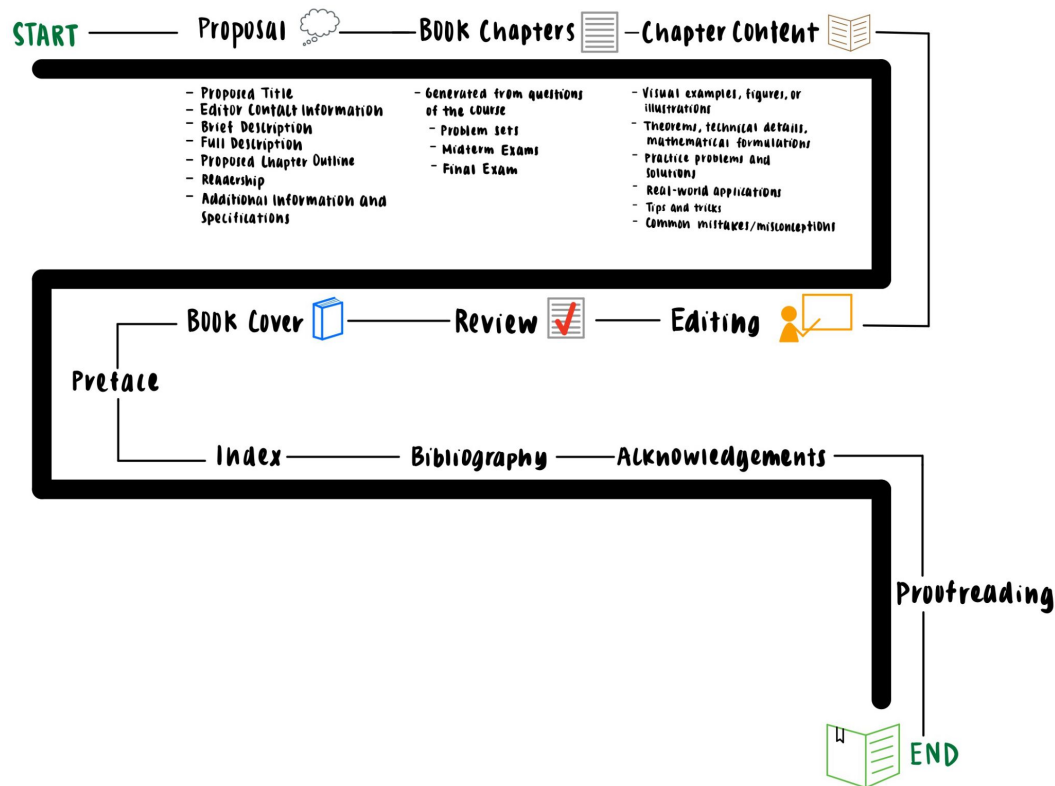
human beyond 8 is reaching limit vs. machine 100,000,000 cuts & pastes per second

allows cut & paste of precise random numbers, arithmetic, programs, interpreters, etc.

emergence is explainable

How can we rapidly write complete books while ensuring quality and originality?

Book Generation



Book Generation

1. Proposal
2. Chapters
3. Contents
4. Cover, preface, index, bibliography, acks
5. Error correction
6. Grammar and originality scores
7. Proofreading

Book Generation

1. Many prompts: 200 prompt books
2. Roles: act as a writer, reviewer, editor
3. Specify content types: text, examples, equations, figures, exercises and solutions.

Defining New Roles and Coining Terms (2023)

Prompter, editor, publisher

Booksets: validation, testing

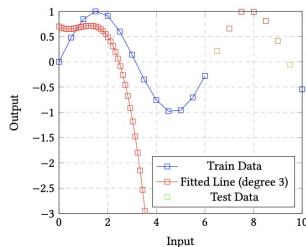
“Here is a 200 prompt book ready for editing by the class professor”

“Here is a meta prompt for the cover”

“This book is prompted by X, edited by Y, and published by Z”

Example Lecture Notes (ML class, Spring 2023)

31



By adding a regularization term to the cost function, we can improve the generalization performance of the model. The green line represents the model fit with regularization, and as we can see, it fits both the training and test data well, resulting in a lower test error.

To combat overfitting, regularization is often used in regression. Regularization is a technique that adds a penalty term to the cost function to discourage the model from fitting the noise in the data and instead encourage simpler, more general models.

There are two main types of regularization used in regression: L1 regularization, also known as Lasso regularization, and L2 regularization, also known as Ridge regularization.

L1 REGULARIZATION adds a penalty term to the cost function proportional to the absolute value of the coefficients. The L1 regularization term is defined as:

$$\lambda \sum_{i=1}^n |\beta_i|$$

Where λ is the regularization parameter and β_i are the coefficients. The L1 regularization term encourages the model to have sparse solutions, meaning that some coefficients will be exactly equal to zero.

L2 REGULARIZATION adds a penalty term to the cost function proportional to the square of the coefficients. The L2 regularization term is defined as:

$$\lambda \sum_{i=1}^n \beta_i^2$$

Where λ is the regularization parameter and β_i are the coefficients. The L2 regularization term encourages the model to have small, non-zero coefficients.

32

When the L2 regularization term is added to the cost function, the resulting cost function is called Ridge Regression, and when L1 regularization term is added it is called Lasso Regression.

Regularization can be added to the cost function by simply adding the regularization term to the mean square error. For example, in Ridge Regression the cost function becomes:

$$J(\beta) = \frac{1}{m} \sum_{i=1}^m (y^{(i)} - \beta_0 - \sum_{j=1}^n \beta_j x_j^{(i)})^2 + \lambda \sum_{i=1}^n \beta_i^2$$

Where λ is the regularization parameter and β_i are the coefficients.

The regularization term acts as a constraint on the coefficients, preventing them from becoming too large and resulting in overfitting. The regularization parameter λ controls the strength of the regularization term and can be tuned to achieve the best trade-off between fitting the data and preventing overfitting.

2.3.1 Ridge Regression

Ridge Regression is a variation of linear regression that uses L2 regularization to prevent overfitting. The cost function for Ridge Regression is defined as:

$$J(\beta) = \frac{1}{m} \sum_{i=1}^m (y^{(i)} - \beta_0 - \sum_{j=1}^n \beta_j x_j^{(i)})^2 + \lambda \sum_{i=1}^n \beta_i^2$$

Where λ is the regularization parameter and β_i are the coefficients.

DERIVING THE ANALYTICAL SOLUTION:

To derive the analytical solution for Ridge Regression, we can take the derivative of the cost function with respect to each β_i and set it equal to zero. By doing this we can find the normal equation as

$$\frac{\partial J(\beta)}{\partial \beta_i} = -\frac{2}{m} \sum_{i=1}^m (y^{(i)} - \beta_0 - \sum_{j=1}^n \beta_j x_j^{(i)}) x_i^{(i)} + 2\lambda \beta_i = 0$$

for $i = 0$,

$$\frac{\partial J(\beta)}{\partial \beta_0} = -\frac{2}{m} \sum_{i=1}^m (y^{(i)} - \beta_0 - \sum_{j=1}^n \beta_j x_j^{(i)}) = 0$$

On solving above equations we can get the following analytical solution for Ridge Regression.

$$\beta = (X^T X + \lambda I)^{-1} X^T Y$$

Where X is the design matrix, Y is the output vector, and I is the Identity matrix.

To find the values of the coefficients that minimize the cost function, the gradient descent algorithm is used. The gradient descent algorithm is an

33

iterative algorithm that starts with initial values for the coefficients and updates them in the direction of the negative gradient of the cost function. The update rule for the coefficients in Ridge Regression is:

$$\beta_i = \beta_i - \alpha \frac{\partial J(\beta)}{\partial \beta_i}$$

Where α is the learning rate. The gradient of the cost function with respect to β_i is:

$$\frac{\partial J(\beta)}{\partial \beta_i} = \frac{2}{m} \sum_{i=1}^m (y^{(i)} - \beta_0 - \sum_{j=1}^n \beta_j x_j^{(i)}) x_i^{(i)} - 2\lambda \beta_i$$

so, the update rule becomes:

$$\beta_i = \beta_i - \alpha \frac{2}{m} \sum_{i=1}^m (y^{(i)} - \beta_0 - \sum_{j=1}^n \beta_j x_j^{(i)}) x_i^{(i)} - 2\lambda \beta_i$$

In Ridge Regression, the L2 regularization term shrinks the coefficients towards zero, but it doesn't make them zero. The value of the regularization parameter, λ , determines the strength of the regularization. A high value of λ will result in small coefficients (close to zero) and a low value of λ will result in large coefficients.

It's important to note that Ridge Regression is a technique to prevent overfitting in the model by adding a bias term to the cost function, which causes the model to prefer solutions with small coefficients. As a result, the values of the coefficients become close to zero, but not zero, compared to Lasso Regression where the values of the coefficients become zero. This property of Ridge Regression makes it useful when we have a large number of features and we want to keep all of them in the model.

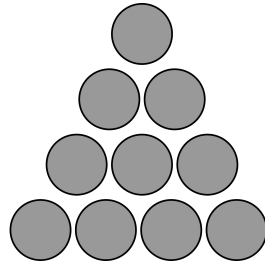
Additionally, Ridge Regression also helps to address the issue of multicollinearity, which occurs when there is a high correlation between independent variables. In this case, the coefficients can become unstable, leading to large variances in the model's predictions. Regularization helps to reduce the variance by shrinking the coefficients towards zero.

Another advantage of Ridge Regression is that it is computationally efficient, as it only requires the inversion of a matrix, which can be done using linear algebra libraries such as NumPy or scikit-learn. This makes it a popular choice for large datasets.

However, one limitation of Ridge Regression is that it does not perform feature selection, meaning it does not set any coefficients to zero. This can be an issue if we have a large number of features and some of them are not relevant to the model. In such cases, Lasso Regression or Elastic Net may be more suitable.

2.3.2 Lasso Regression

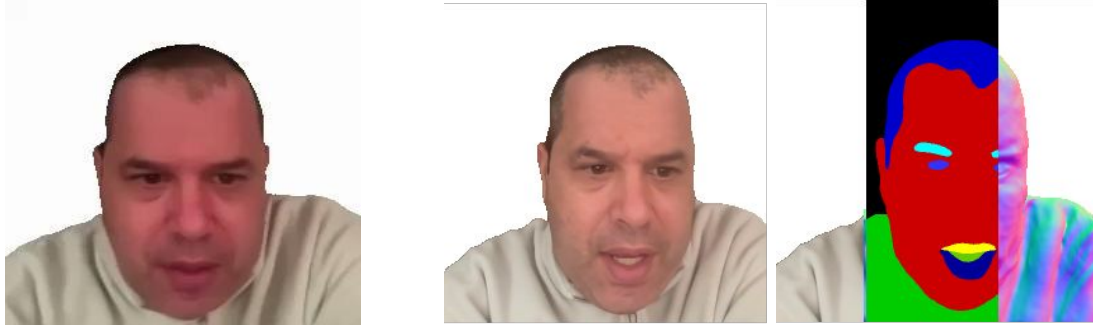
The Lasso regression is a form of linear regression that uses a regularization term known as L1 regularization. The objective function in Lasso regression



How can we drive a photo-realistic speaking avatar to act as a TA or lecturer?

AI Generated Avatars

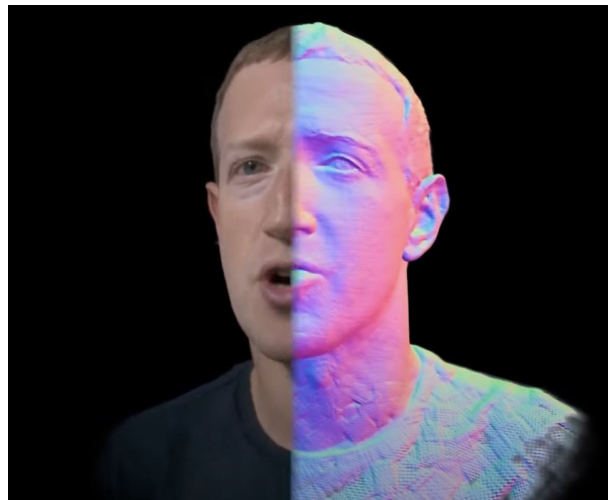
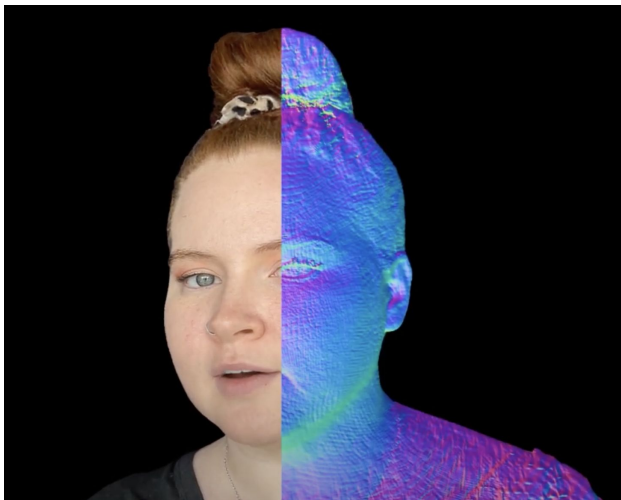
Photorealistic speaking avatars delivering machine generated content.



Photorealistic Avatars

Reconstruction

Photorealistic



Source: Meta avatar codec

AI Generated Avatars

Uncanny valley

Example: 40% Terence Tao, 40% Esther Perel, 20% you



Figure source: Masterclass

AI Generated Avatar Lecture (Fall 2023)

Automatic text, speech, gestures using language models

Pass uncanny valley of photo realism

Real-time rendering for live interaction with students

10 Most Recent Related Publications

Prospects and perils of writing books in Mathematics and Computer Science using AI

Iddo Drori, Sarah J. Zhang, Sage Simhon, Yann Hicke, Zad Chin, Keith Tyser, Harsh Sharma, Kirsi Rajagopal, Alice Zhang, Annie Wang, Eugenia Feng, Nikhil Singh, Lauren Cowles, Tonio Buonassisi, Madeleine Udell, Gilbert Strang, Armando Solar-Lezama
In progress

Automatically fulfilling the MIT Mathematics and EECS graduation requirements at a human level by self error correction and few-shot learning

Iddo Drori, Sarah J. Zhang, Sage Simhon, Keith Tyser, Sarah Zhang, Reece Shuttleworth, Pedro Lantigua, Arvind Raghavan, Zad Chin, Saisamrit Surbehera, Leonard Tang, Yann Hicke, Avi Shporer, Nakul Verma, Tonio Buonassisi, Gilbert Strang, Armando Solar-Lezama
Under review

ChatMIT: A dataset for graduating from MIT Mathematics & EECS, achieving human solve rates, curriculum analysis, and generating class questions

Sarah J. Zhang, Sage Simhon, Yann Hick, Zad Chin, Annie Wang, Kirsi Rajagopal, Alice Zhang, Eugnia Feng, Kieth Tyser, Harsh Sharma, Tonio Buonassisi, Armando Solar-Lezama, Iddo Drori
Under review

Text to graphics by program synthesis with error correction on precise, procedural, and simulation tasks

Arvind Raghavan, Zad Chin, Alexander E. Siemenn, Vitali Petsiuk, Saisamrit Surbehera, Yann Hicke, Edward Chien, Ori Kerret, Tonio Buonassisi, Kate Saenko, Armando Solar-Lezama, Iddo Drori
Under review

From human days to machine seconds: Automatically answering and generating machine learning final exams

Sarah J. Zhang, Keith Tyser, Sage Simhon, Sarah Zhang, Reece Shuttleworth, Zad Chin, Pedro Lantigua, Saisamrit Surbehera, Gregory Hunter, Derek Austin, Yann Hicke, Leonard Tang, Sathwik Karnik, Darnell Granberry, Iddo Drori
Under review

A dataset for learning university STEM courses at scale and generating questions at a human level

Iddo Drori, Sarah Zhang, Zad Chin, Reece Shuttleworth, Albert Lu, Linda Chen, Bereket Birbo, Michele He, Pedro Lantigua, Sunny Tran, Gregory Hunter, Bo Feng, Newman Cheng, Roman Wang, Yann Hicke, Saisamrit Surbehera, Arvind Raghavan, Alexander Siemenn, Nikhil Singh, Jayson Lynch, Avi Shporer, Nakul Verma, Tonio Buonassisi, Armando Solar-Lezama
Educational Advances in Artificial Intelligence (EAAI), 2023.

Human evaluation of text-to-image models on a multi-task benchmark

Vitali Petsiuk, Alexander Siemenn, Saisamrit Surbehera, Zad Chin, Kieth Tyser, Gregory Hunter, Arvind Raghavan, Yann Hicke, Bryan Plummer, Ori Kerret, Tonio Buonassisi, Kate Saenko, Armando Solar-Lezama, Iddo Drori
NeurIPS Workshop on Human Evaluation of Generative Models, 2022.

A neural network solves, explains, and generates university math problems by program synthesis and few-shot learning at human level

Iddo Drori, Sarah Zhang, Reece Shuttleworth, Leonard Tang, Albert Lu, Elizabeth Ke, Kevin Liu, Linda Chen, Sunny Tran, Newman Cheng, Roman Wang, Nikhil Singh, Taylor L. Patti, Jayson Lynch, Avi Shporer, Nakul Verma, Eugene Wu, Gilbert Strang
Proceedings of the National Academy of Sciences (PNAS), 119(32), 2022.

Solving Probability and Statistics problems by probabilistic program synthesis at human level and predicting solvability

Leonard Tang, Elizabeth Ke, Nikhil Singh, Bo Feng, Derek Austin, Nakul Verma, Iddo Drori
International Conference on Artificial Intelligence in Education (AIED), 2022.

Solving machine learning problems

Sunny Tran, Pranav Krishna, Ishan Pakuwal, Prabhakar Kafe, Nikhil Singh, Jayson Lynch, Iddo Drori
Asian Conference on Machine Learning (ACML), 2021.

Best paper award winner

Spring 2023 Team

Iddo Drori, MIT, Columbia University, Boston University

Yann Hicke, Cornell University

Sarah Zhang, MIT

Sage Simhon, MIT

Zad Chin, Harvard

Annie Wang, MIT

Alice Zhang, MIT

Eugenia Feng, MIT

Samuel Florin, MIT

Harsh Sharma, Boston University

Keith Tyser, Boston University

Andrei Marginean, MIT

Saisamrit Surbehera, Columbia University

Nikhil Singh, MIT

Leonard Tang, Harvard

Lauren Cowles, Cambridge University Press

Gilbert Strang, MIT

Tonio Buonasisi, MIT

Madeleine Udell, Stanford

Armando Solar Lezama, MIT

Thank you