

S&DS 265 / 565

Introductory Machine Learning

Language Models

October 25

Yale

Welcome back!

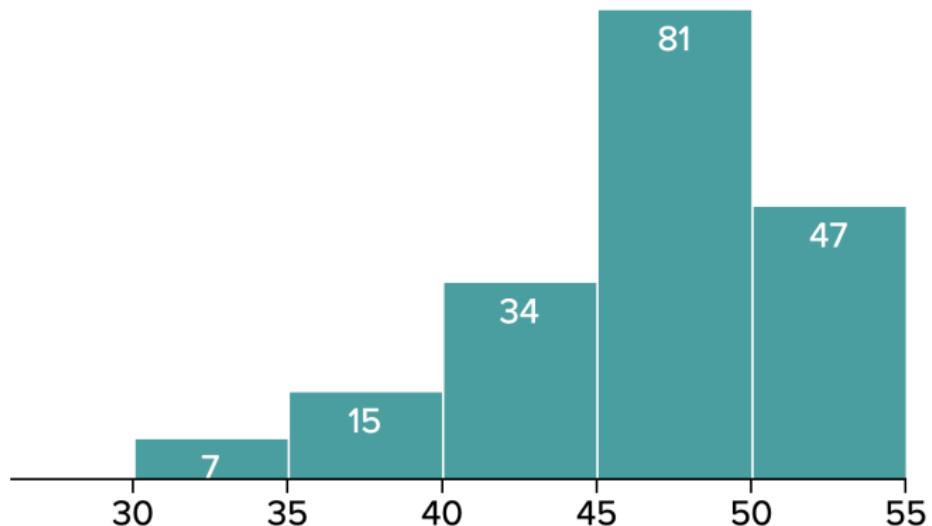
For today:

- Where have we been? Where are we going?
- Language models

But first...

- Midterms scores posted later today
- Provisional grades announced later today
- Happy to meet to discuss grading, standing...
- Assignment 3 due next Tues, Nov. 1
- Assignment 4 out this Thursday

Midterm distribution



median 46.5, mean 45.3, max 54.5, std 7.2

Panel discussion

Class on Tuesday, December 6:

We will have a panel discussion on

Societal issues for AI and Machine Learning

If you are interested in participating, please email me

Subject: iML December 6 panel

Where we've been

1	Sept 1	Course overview		Sept 1: Course overview		
2	Sept 6, 8	Python and background concepts	Python elements Covid trends	Sept 6: Python elements Sept 8: Pandas and linear regression	Data8 Chapters 3, 4, 5	Thu: Quiz 1
3	Sept 13, 15	Linear regression and classification	Covid trends (revisited) Classification examples	Sept 13: Regression concepts Sept 15: Classification	ISL Sections 3.1, 3.2, 3.5 Notes on regression ISL Sections 4.3, 4.4 Notes on classification	Thu: Assn 1
4	Sept 20, 22	Stochastic gradient descent	SGD examples	Sept 20: Classification (continued) Sept 22: Stochastic gradient descent	ISL Section 6.2.2 ISL Section 10.7.2	Thu: Quiz 2
5	Sept 27, 29	Bias and variance, cross-validation	Bias-variance tradeoff Covid trends (revisited) California housing	Sept 27: Bias and variance Sept 29: Cross-validation	ISL Section 2.2 ISL Section 5.1	Thu: Assn 1 in Assn 2 out
6	Oct 4, 6	Tree-based methods	Trees and forests Visualizing trees Bagging operations	Oct 4: Trees Oct 6: Forests	ISL Sections 8.1, 8.2	Thu: Quiz 3
7	Oct 11, 13	PCA and dimension reduction	PCA examples PCA revisited Used for regression	Oct 11: PCA Oct 13: PCA and review	ISL Section 12.2	Thu: Assn 2 in Assn 3 out
8	Oct 18	Midterm exam (in class)			On Canvas: Practice midterms / Sample solns Midterm / Sample soln	

Where we're going

8	Oct 18	Midterm exam (in class)			On Canvas: Practice midterms / Sample solns Midterm / Sample soln	
9	Oct 25, 27	Language models, word embeddings	Word embeddings	Oct 25: Language models Oct 27: Word embeddings		Assn 4 out
10	Nov 1, 3	Bayesian inference, topic models	Mixtures Bayesian inference Topic models	Nov 1: Bayesian inference Nov 3: Bayes and topic models	Notes on Bayesian inference Notes on simulation	Tue: Assn 3 in Thu: Quiz 4
11	Nov 8, 10	Introduction to neural networks	Minimal neural network Regression examples	Nov 8: Topic models Nov 10: Neural networks	ISL Sections 10.1, 10.2	Thu: Assn 4 in Assn 5 out
12	Nov 15, 17	Deep neural networks	Tensorflow playground Autoencoder examples	Nov 15: Neural networks (continued) Nov 17: Autoencoders	ISL Section 10.7 Notes on backpropagation	Thu: Quiz 5
13	Nov 22, 24	No class, Thanksgiving break				
14	Nov 29, Dec 1	Reinforcement learning	Q-learning	Nov 29: Reinforcement learning Dec 1: Deep reinforcement learning		Thu: Assn 5 in Assn 6 out
15	Dec 6, 8	Societal issues for machine learning		Dec 6: Societal issues Dec 8: Course wrap up		Thu: Quiz 6
16	Dec 15					Thu: Assn 6 in
17	Mon, Dec 19, 7pm	Final exam			Registrar: Final exam schedule Practice final, sample solution	

For Today

- Language models
- Concepts...no new methods

When you text someone

- Take out your cell phone...

When you text someone

- Take out your cell phone...
- Text someone

When you text someone

- Take out your cell phone...
- Text someone
- What did you notice?

Language models

- A language model is a way of assigning a probability to any sequence of words (or string of text)

$$p(w_1, \dots, w_n)$$

Language models

- A language model is a way of assigning a probability to any sequence of words (or string of text)

$$p(w_1, \dots, w_n)$$

- By the basic rules of conditional probability we can factor this as

$$p(w_1, \dots, w_n) = p(w_1)p(w_2 | w_1) \dots p(w_n | w_1, \dots, w_{n-1})$$

Language models

- A language model is a way of *generating* any sequence of words

$$P(\text{"the whole forest had been anesthetized"}) =$$

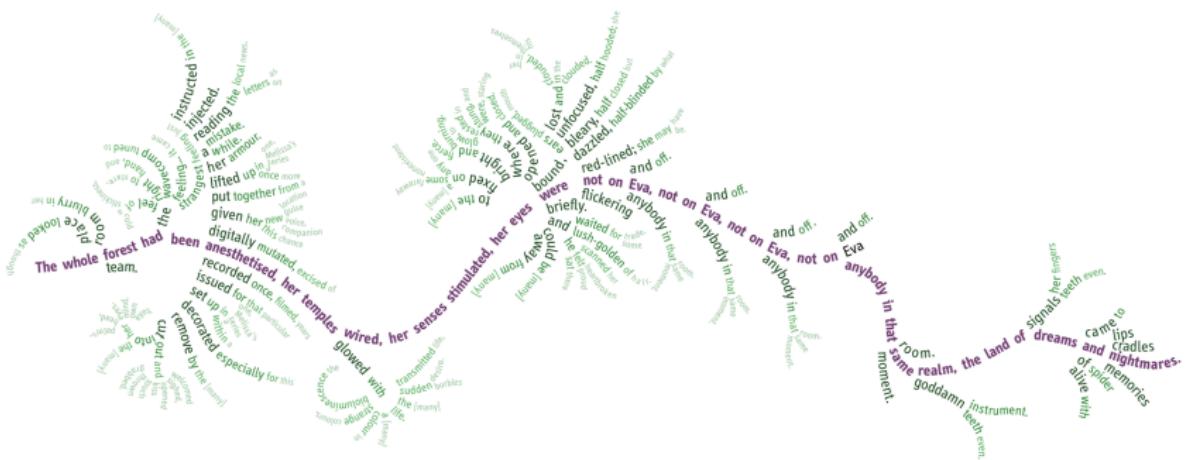
$$\begin{aligned} & P(\text{"the"}) \times P(\text{"whole"} | \text{"the"}) \\ & \quad \times P(\text{"forest"} | \text{"the whole"}) \\ & \quad \times P(\text{"had"} | \text{"the whole forest"}) \\ & \quad \times P(\text{"been"} | \text{"the whole forest had"}) \\ & \quad \times P(\text{"anesthetized"} | \text{"the whole forest had been"}) \end{aligned}$$

Remixing Noon

Text generated from Channel Skin by Jeff Noon

"The whole forest had been anesthetised, her temples wired, her senses stimulated, her eyes were not on Eva, not on Eva, not on Eva, not on anybody in that same realm, the land of dreams and nightmares."

Viability: 0.000000326%



https://revdancatt.com/2017/03/01/markov_noon

Text generation

- Words generated one-by-one
- A word is chosen by sampling from a probability distribution
- Then treated as if it were “real,” as in dreaming
- Result is purely synthetic text

Uses of language models

- Speech recognition

Often built using Bayes' rule: $P(\text{signal} \mid \text{words}) \propto P(\text{words} \mid \text{signal}) \cdot P(\text{words})$

Uses of language models

- Speech recognition
- Machine translation

Often built using Bayes' rule: $P(\text{signal} \mid \text{words}) \propto P(\text{words} \mid \text{signal}) \cdot P(\text{words})$

Uses of language models

- Speech recognition
- Machine translation
- Text compression

Often built using Bayes' rule: $P(\text{signal} \mid \text{words}) \propto P(\text{words} \mid \text{signal}) \cdot P(\text{words})$

Uses of language models

- Speech recognition
- Machine translation
- Text compression
- Texting

Often built using Bayes' rule: $P(\text{signal} \mid \text{words}) \propto P(\text{words} \mid \text{signal}) \cdot P(\text{words})$

Uses of language models

- Speech recognition
- Machine translation
- Text compression
- Texting
- Email completion

Often built using Bayes' rule: $P(\text{signal} \mid \text{words}) \propto P(\text{words} \mid \text{signal}) \cdot P(\text{words})$

Uses of language models

- Speech recognition
- Machine translation
- Text compression
- Texting
- Email completion
- Image captioning

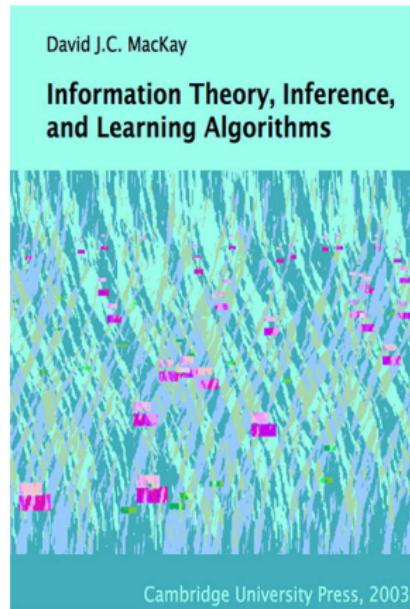
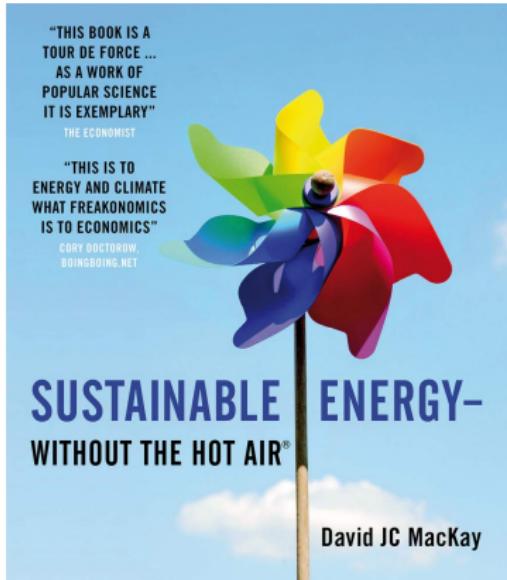
Often built using Bayes' rule: $P(\text{signal} \mid \text{words}) \propto P(\text{words} \mid \text{signal}) \cdot P(\text{words})$

Uses of language models

- Speech recognition
- Machine translation
- Text compression
- Texting
- Email completion
- Image captioning
- Mind reading from fMRI

Often built using Bayes' rule: $P(\text{signal} \mid \text{words}) \propto P(\text{words} \mid \text{signal}) \cdot P(\text{words})$

David MacKay



Dasher: LMs for assistive devices

Language models enable new modes of text input:

https://www.youtube.com/watch?v=quw_Kci4fUg

<https://youtu.be/QxFEUk3J89Q?t=72>

Dasher poetry:

<https://www.youtube.com/watch?v=x-WLiY2p1LQ>

Language models

- A language model is a way of assigning a probability to any sequence of words (or string of text)

$$p(w_1, \dots, w_n)$$

Language models

- A language model is a way of assigning a probability to any sequence of words (or string of text)

$$p(w_1, \dots, w_n)$$

- By the basic rules of conditional probability we can factor this as

$$p(w_1, \dots, w_n) = p(w_1)p(w_2 | w_1) \dots p(w_n | w_1, \dots, w_{n-1})$$

Language models

- A language model is a way of assigning a probability to any sequence of words (or string of text)

$$p(w_1, \dots, w_n)$$

- By the basic rules of conditional probability we can factor this as

$$p(w_1, \dots, w_n) = p(w_1)p(w_2 | w_1) \dots p(w_n | w_1, \dots, w_{n-1})$$

- The number of *histories* grows as V^{n-1} . Number of parameters in model grows as V^n , where V is number of words in vocabulary.

Language models

- A language model is a way of assigning a probability to any sequence of words (or string of text)

$$p(w_1, \dots, w_n)$$

- By the basic rules of conditional probability we can factor this as

$$p(w_1, \dots, w_n) = p(w_1)p(w_2 | w_1) \dots p(w_n | w_1, \dots, w_{n-1})$$

- The number of *histories* grows as V^{n-1} . Number of parameters in model grows as V^n , where V is number of words in vocabulary.
- What are some ways of reducing the number of parameters?

One approach: Grouping histories

- Let $g(w_1, \dots, w_n)$ be the group assigned to the history

One approach: Grouping histories

- Let $g(w_1, \dots, w_n)$ be the group assigned to the history
- Our model becomes

$$p(w_{n+1} | w_1, \dots, w_n) = p(w_{n+1} | g(w_1, \dots, w_n))$$

One approach: Grouping histories

- Let $g(w_1, \dots, w_n)$ be the group assigned to the history
- Our model becomes

$$p(w_{n+1} | w_1, \dots, w_n) = p(w_{n+1} | g(w_1, \dots, w_n))$$

- Number of parameters: $O(V \cdot \text{number of groups})$

One approach: Grouping histories

- Let $g(w_1, \dots, w_n)$ be the group assigned to the history
- Our model becomes

$$p(w_{n+1} | w_1, \dots, w_n) = p(w_{n+1} | g(w_1, \dots, w_n))$$

- Number of parameters: $O(V \cdot \text{number of groups})$
- What are some example groupings?

Grouping histories

- Unigrams: $g(w_1, \dots, w_n) = \emptyset$.

Grouping histories

- Unigrams: $g(w_1, \dots, w_n) = \emptyset$.
- Bigrams: $g(w_1, \dots, w_n) = w_n$.

Grouping histories

- Unigrams: $g(w_1, \dots, w_n) = \emptyset$.
- Bigrams: $g(w_1, \dots, w_n) = w_n$.
- Trigrams: $g(w_1, \dots, w_n) = (w_{n-1}, w_n)$.

Grouping histories

- Unigrams: $g(w_1, \dots, w_n) = \emptyset$.
- Bigrams: $g(w_1, \dots, w_n) = w_n$.
- Trigrams: $g(w_1, \dots, w_n) = (w_{n-1}, w_n)$.
- Number of parameters grows as $O(V)$, $O(V^2)$, and $O(V^3)$, respectively.

Estimating parameters

- The maximum likelihood estimate of a trigram model:

$$\hat{p}(w_3 | w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)}$$

Estimating parameters

- The maximum likelihood estimate of a trigram model:

$$\hat{p}(w_3 | w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)}$$

- What are some problems with this model?

Sparse data problem

The next group of slides present one way of quantifying the problem of sparse data in language modeling

Half Earth

Aug 06, 2018

by Foundation Staff

0 Comment

Google Earth, Half-Earth Project, Map of Life

By Jeremy Malczyk, Michelle Duong, Ajay Ranipeta, Chris Heltne, Walter Jetz of Map of Life, Yale University, and the E.O. Wilson Biodiversity Foundation Half-Earth Project

This article originally in *Medium*, July 30, 2018

The Significance of Biodiversity



Narrow-billed Tody, *Todus angustirostris*. Photo by Julie Hart.



Learn how you can be part of bringing
Half-Earth to life.

<https://eowilsonfoundation.org/mapping-species-for-half-earth/>

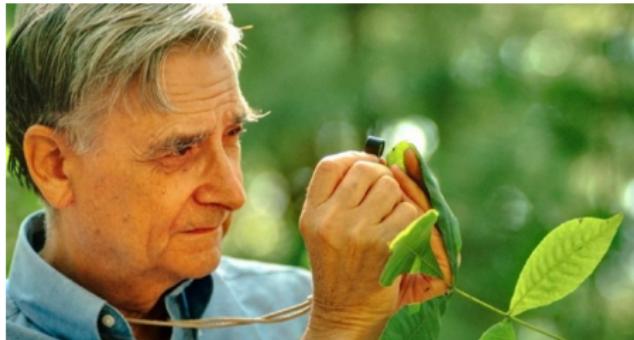
<https://www.half-earthproject.org/>

Specious species



- A naturalist (say, Edward O. Wilson) explores a region and observes animals (organisms).

Specious species



- A naturalist (say, Edward O. Wilson) explores a region and observes animals (organisms).
- He finds 100,000 animals and 5,000 species.

Specious species



- A naturalist (say, Edward O. Wilson) explores a region and observes animals (organisms).
- He finds 100,000 animals and 5,000 species.
- What is the chance I'll find a new species?

Specious species



- A naturalist (say, Edward O. Wilson) explores a region and observes animals (organisms).
- He finds 100,000 animals and 5,000 species.
- What is the chance I'll find a new species?
- What if Wilson observes 100 unique species?

Missing species: Good-Turing

- Wilson observes 100,000 animals and 5,000 species, 100 of them are unique (only one observation of that species).

Missing species: Good-Turing

- Wilson observes 100,000 animals and 5,000 species, 100 of them are unique (only one observation of that species).
- Good-Turing: Estimate of the probability that the next animal is a new species? $\hat{p}_{GT} = 100/100,000 = 10^{-3}$.

Missing species: Good-Turing

- Wilson observes 100,000 animals and 5,000 species, 100 of them are unique (only one observation of that species).
- Good-Turing: Estimate of the probability that the next animal is a new species? $\hat{p}_{GT} = 100/100,000 = 10^{-3}$.
- This is an estimate of the missing probability mass.

Yale researchers create map of undiscovered life

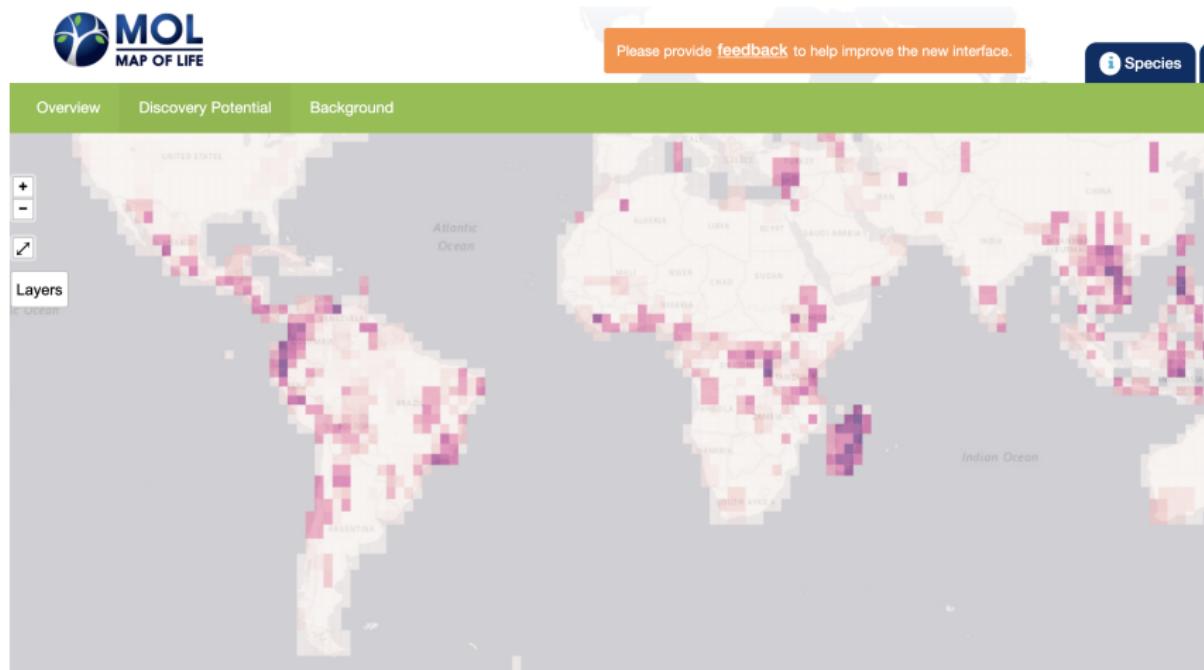
By Bill Hathaway | MARCH 22, 2021



Less than a decade after unveiling the "[Map of Life](#)," a global database that marks the distribution of known species across the planet, Yale researchers have launched an ambitious and perhaps even more important project — creating a map of where life has yet to be discovered.

For [Walter Jetz](#), a professor of ecology and evolutionary biology at Yale who spearheaded the Map of Life project, the new effort is a moral imperative that can help support biodiversity discovery and preservation around the world.

Map of Life



Map of Life



[contact us](#) [login](#) [register](#)

Putting biodiversity on the map

The page features a green header bar at the top. Below it is a grid of eight cards, each representing a different feature of the Map of Life platform. The cards are arranged in two rows of four. Each card includes an image, a title, a brief description, and a larger explanatory text below.

- Map species**
View species range map, inventory, and occurrence data
- Project species**
Explore species habitat loss projected for a range of plausible futures
- Species by location**
Select a location, filter by distance or group, and view a list of species along with source data
- Explore Places**
Dashboard for biodiversity data coverage and conservation information
- Indicators**
Explore trends in biodiversity knowledge, distribution, and conservation
- Patterns**
Explore richness patterns and biodiversity facets
- Datasets**
Explore datasets used across MOL
- Mobile App**
Discover, identify, and record biodiversity worldwide

Sparse data: Species \approx words

- Suppose we have a corpus of 500 million words. I count trigrams and find that 50 million of them are unique.

Sparse data: Species \approx words

- Suppose we have a corpus of 500 million words. I count trigrams and find that 50 million of them are unique.
- When I see a new trigram, 10% of the time it won't have been seen before. (This is a typical number.)

Sparse data: Species \approx words

- Suppose we have a corpus of 500 million words. I count trigrams and find that 50 million of them are unique.
- When I see a new trigram, 10% of the time it won't have been seen before. (This is a typical number.)
- This means that the MLE is zero, and the probability that my model predicts the next word will be zero.

Sparse data: Species \approx words

- Suppose we have a corpus of 500 million words. I count trigrams and find that 50 million of them are unique.
- When I see a new trigram, 10% of the time it won't have been seen before. (This is a typical number.)
- This means that the MLE is zero, and the probability that my model predicts the next word will be zero.
- The MLE is supported on the observed data. We need to spread out the probability over unseen events.

Estimating parameters

- The maximum likelihood estimate of a trigram model:

$$\hat{p}(w_3 | w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)}$$

Estimating parameters

- The maximum likelihood estimate of a trigram model:

$$\hat{p}(w_3 | w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)}$$

- Some kind of “shrinkage” or smoothing needs to be done.

Estimating parameters

- The maximum likelihood estimate of a trigram model:

$$\hat{p}(w_3 | w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)}$$

- Some kind of “shrinkage” or smoothing needs to be done.
- How else can the model be strengthened?

Interpolation

Linear interpolation:

$$p(w_3 | w_1, w_2) = \lambda_3 \hat{p}(w_3 | w_1, w_2) + \lambda_2 \hat{p}(w_3 | w_2) + \lambda_1 \hat{p}(w_3)$$

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

Interpolation

Linear interpolation:

$$p(w_3 | w_1, w_2) = \lambda_3 \hat{p}(w_3 | w_1, w_2) + \lambda_2 \hat{p}(w_3 | w_2) + \lambda_1 \hat{p}(w_3)$$

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

This is a type of “mixture model”

How good is a language model?

The next group of slides present a useful way of quantifying how good a language model is

Recall: Geometric mean

The *arithmetic mean* of 1/4, 4, 8 is

$$\frac{1}{3} \left(\frac{1}{4} + 4 + 8 \right) = 4.08\bar{3}$$

Recall: Geometric mean

The *arithmetic mean* of 1/4, 4, 8 is

$$\frac{1}{3} \left(\frac{1}{4} + 4 + 8 \right) = 4.08\bar{3}$$

The *geometric mean* of 1/4, 4, 8 is

$$\sqrt[3]{\frac{1}{4} \cdot 4 \cdot 8} = 2$$

Recall: Geometric mean

The *arithmetic mean* of 1/4, 4, 8 is

$$\frac{1}{3} \left(\frac{1}{4} + 4 + 8 \right) = 4.08\bar{3}$$

The *geometric mean* of 1/4, 4, 8 is

$$\sqrt[3]{\frac{1}{4} \cdot 4 \cdot 8} = 2$$

The geometric mean is no greater than the arithmetic mean

Recall: Geometric mean

The *geometric mean* of x_1, \dots, x_n is

$$\sqrt[n]{x_1 x_2 \cdots x_n} = \left(\prod_{i=1}^n x_i \right)^{\frac{1}{n}}$$

Recall: Geometric mean

The *geometric mean* of x_1, \dots, x_n is

$$\sqrt[n]{x_1 x_2 \cdots x_n} = \left(\prod_{i=1}^n x_i \right)^{\frac{1}{n}}$$

How good is a language model? Perplexity

Perplexity is defined as

$$\text{Perplexity}(\theta) = \left(\prod_{i=1}^N p_\theta(w_i | w_{1:i-1}) \right)^{-\frac{1}{N}}$$

where w_1, w_2, \dots, w_N is a large chunk of text that wasn't used to train the language model.

How good is a language model? Perplexity

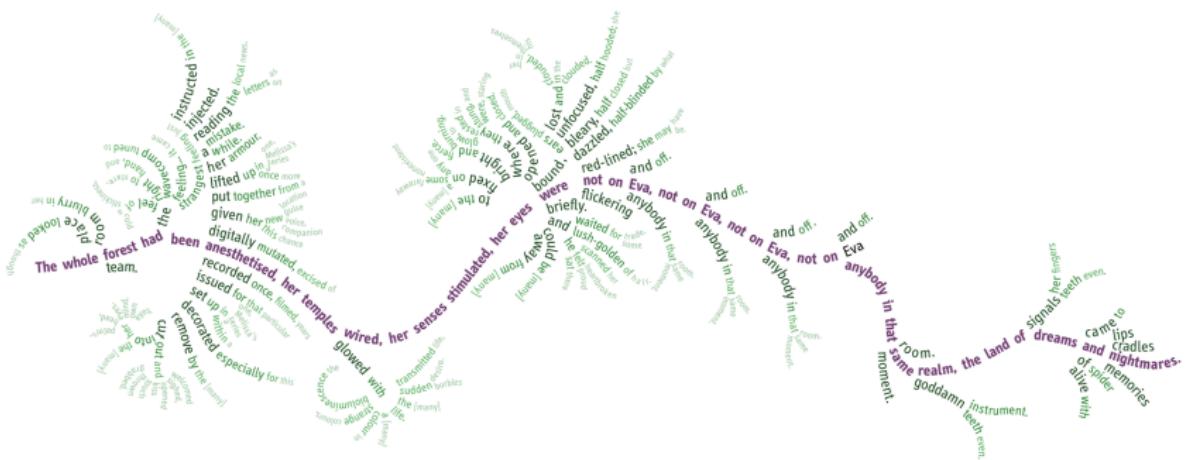
- Perplexity is the inverse of the geometric mean of the word probabilities
- If the perplexity is 100, the model predicts, on average, as if there were 100 equally likely words to follow
- This is the (geometric) average “branching factor” for the model on real text

Remixing Noon

Text generated from Channel Skin by Jeff Noon

"The whole forest had been anesthetised, her temples wired, her senses stimulated, her eyes were not on Eva, not on Eva, not on Eva, not on anybody in that same realm, the land of dreams and nightmares."

Viability: 0.000000326%



https://revdancatt.com/2017/03/01/markov_noon

Modern language models

Suppose a computer program assigns a “score” to possible next words:

$$s(v; \underbrace{w_1, \dots, w_n}_{\text{word history}})$$

Modern language models

Suppose a computer program assigns a “score” to possible next words:

$$s(v; \underbrace{w_1, \dots, w_n}_{\text{word history}})$$

Can convert this to a language model by the “softmax” operation:

$$p(w | w_1, \dots, w_n) = \frac{\exp(s(w; w_1, \dots, w_n))}{\sum_{v \in V} \exp(s(v; w_1, \dots, w_n))}$$

Modern language models

Suppose a computer program assigns a “score” to possible next words:

$$s(v; \underbrace{w_1, \dots, w_n}_{\text{word history}})$$

Can convert this to a language model by the “softmax” operation:

$$p(w | w_1, \dots, w_n) = \frac{\exp(s(w; w_1, \dots, w_n))}{\sum_{v \in V} \exp(s(v; w_1, \dots, w_n))}$$

In GPT-3, the function $s(v; w_{1:n})$ is learned on large amounts of text (unsupervised) using a type of deep neural network called a *transformer*.

Large language models: GPT-3

In this notebook we illustrate the interface to OpenAI's API. To run this for yourself, you need to register and obtain an [API key](#). You can then try out a range of [different models](#):

GPT-3

Our GPT-3 models can understand and generate natural language. We offer four main models with different levels of power suitable for different tasks. Davinci is the most capable model, and Ada is the fastest.

LATEST ENGINE	DESCRIPTION	MAX REQUEST	TRAINING DATA
text-davinci-002	Most capable GPT-3 model. Can do any task the other models can do, often with less context. In addition to responding to prompts, also supports inserting completions within text.	4,000 tokens	Up to Jun 2021
text-curie-001	Very capable, but faster and lower cost than Davinci.	2,048 tokens	Up to Oct 2019
text-babbage-001	Capable of straightforward tasks, very fast, and lower cost.	2,048 tokens	Up to Oct 2019

Background and press on GPT-3 (optional)



Better Language Models and Their Implications

We've trained a large-scale unsupervised language model which generates coherent paragraphs of text, achieves state-of-the-art performance on many language modeling benchmarks, and performs rudimentary reading comprehension, machine translation, question answering, and summarization—all without task-specific training.



February 14, 2019
24 minute read

GPT-3 101: a brief introduction

It has been almost impossible to avoid the GPT-3 hype in the last weeks. This article offers a quick introduction to its architecture, use cases already available, as well as some thoughts about its ethical and green IT implications.



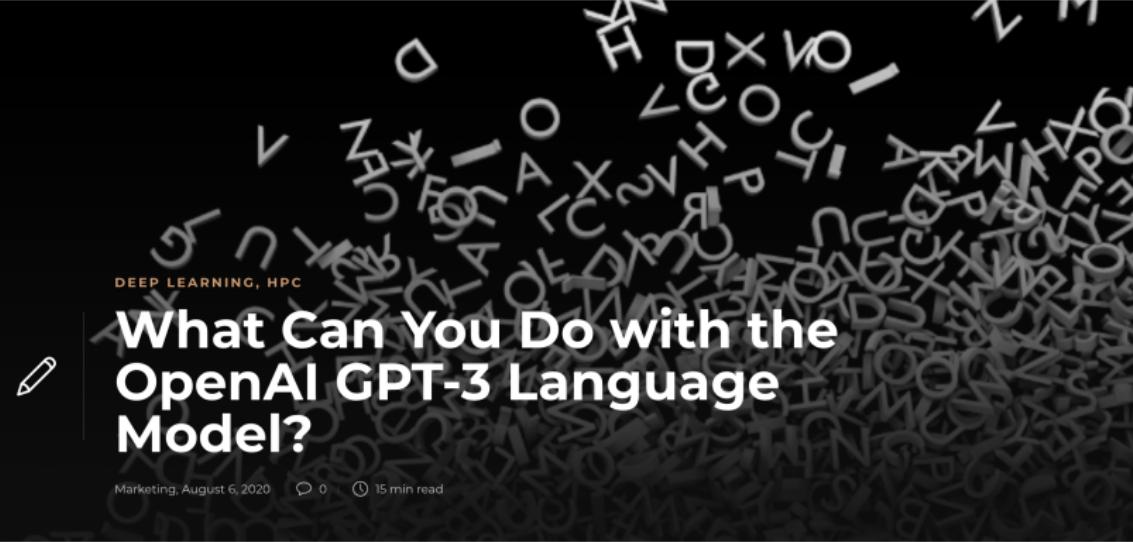
David Pereira Jul 25, 2020 · 6 min read ★



Introduction

Let's start with the basics. GPT-3 stands for Generative Pretrained Transformer version 3, and it is a sequence transduction model. Simply put, sequence transduction is a technique that transforms an input sequence to an output sequence.

GPT-3 is a language model, which means that, using sequence transduction, it can predict the likelihood of an output sequence given an input sequence. This can be used, for instance to predict which word makes the most sense given a text sequence.



DEEP LEARNING, HPC

What Can You Do with the OpenAI GPT-3 Language Model?

Marketing, August 6, 2020

0

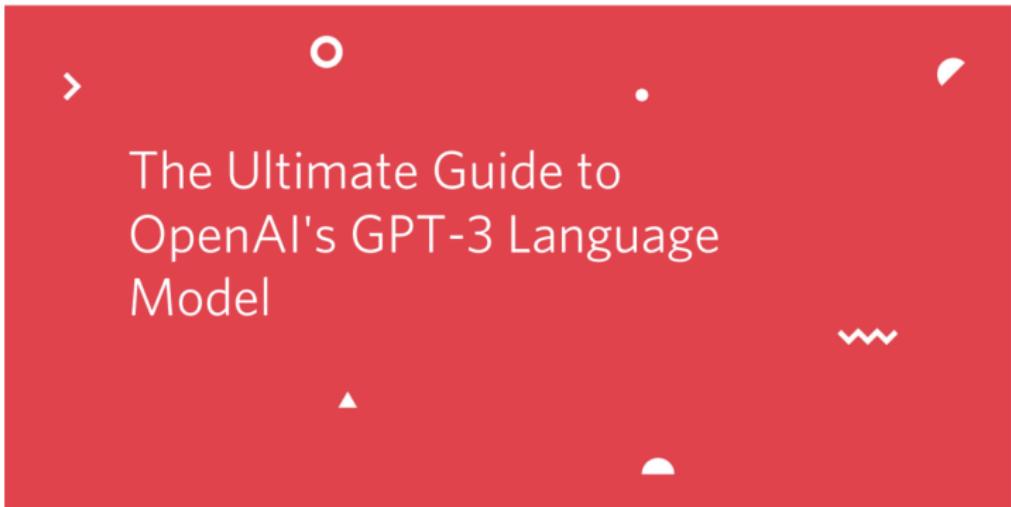
15 min read



**Exploring GPT-3: A New Breakthrough in Language
Generation**



The Ultimate Guide to OpenAI's GPT-3 Language Model



Generative Pre-trained Transformer 3 (GPT-3) is a new language model created by OpenAI that is able to generate written text of such quality that is often difficult to differentiate from text written by a human.

Opinion

How Do You Know a Human Wrote This?

Machines are gaining the ability to write, and they are getting terrifyingly good at it.



By Farhad Manjoo
Opinion Columnist

July 29, 2020





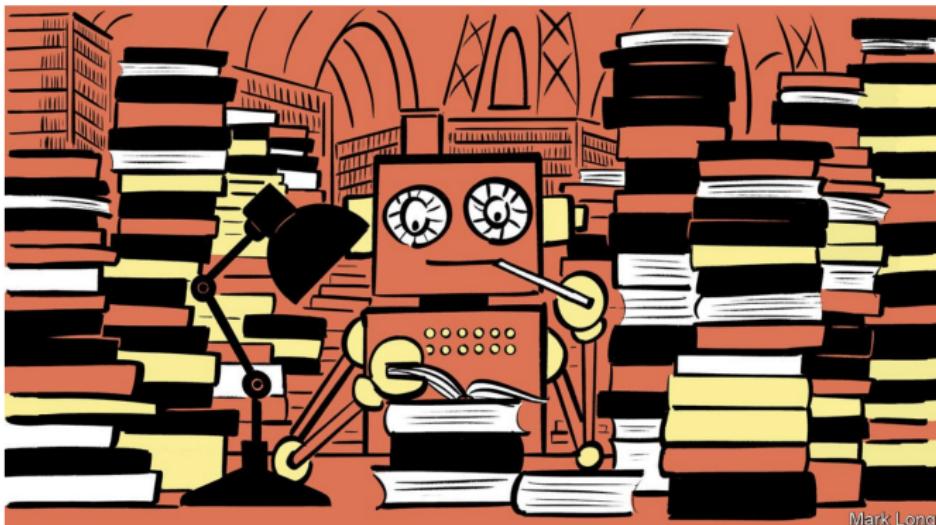
Science &
technology

[Aug 6th 2020 edition >](#)

Artificial intelligence

A new AI language model generates poetry and prose

GPT-3 can be eerily human-like—for better and for worse



Mark Long

Meet GPT-3. It Has Learned to Code (and Blog and Argue).

The latest natural-language system generates tweets, pens poetry, summarizes emails, answers trivia questions, translates languages and even writes its own computer programs.



Next time

- Words that have similar neighbors should be similar
- ... a recursive notion of similarity!
- This will be an intuition behind “word embeddings”

Summary of today: Language models

- A language model is used to predict or generate the next word
- Used many different applications
- Probabilities need to be “smoothed” to avoid zeros
- Perplexity is a measure of a language model’s predictive power
- GPT-3 is a hint at current frontier of AI