Language Models

Announcements

In the News

Software Generated by Software

20–30% of the code that's inside of our repos today is written by software

Language Models

Language Model

Language model inference is a function f(**context**) that returns the next word. This next word is typically chosen at random among likely next words.

- Historically, the **context** was some incomplete text:
  - "Oski the Bear is the official **...**"
  - "Oski was suspended for two weeks in January 1990, for throwing a cake towards **...**"
- Recent language models have expanded the notion of context to include other data as well
- Long text completions are generated one word at a time by repeating:
  - Pick a word according to f(**context**)
  - Add that word to the context

Querying a Language Model

Steps to complete a string of text using a language model in Python:

- Install and run ollama (free open-source software): **https://ollama.com/**
- Download an open-parameter language model (4Gb): **ollama pull gemma3:4b-it-qat**
- Install a Python module to interact with the ollama server: **pip install ollama**
- Use Python to query the language model. E.g.,

```
import ollama

gemma = 'gemma3:4b-it-qat'  # a small model
prefix = 'Oski the Bear is the official'
output = ollama.generate(model=gemma, prompt=prefix, raw=True)
print(output.response)
```

(Demo)

Neural Networks

Neural Networks (No Math Version)

A neural network defines a function by combining an architecture and parameters (numbers).

The architecture is (typically) chosen; the paramaters are "learned" from data.

The critical property of neural networks is that they identify patterns of similarity, even when those patterns are complex.

(Demo)

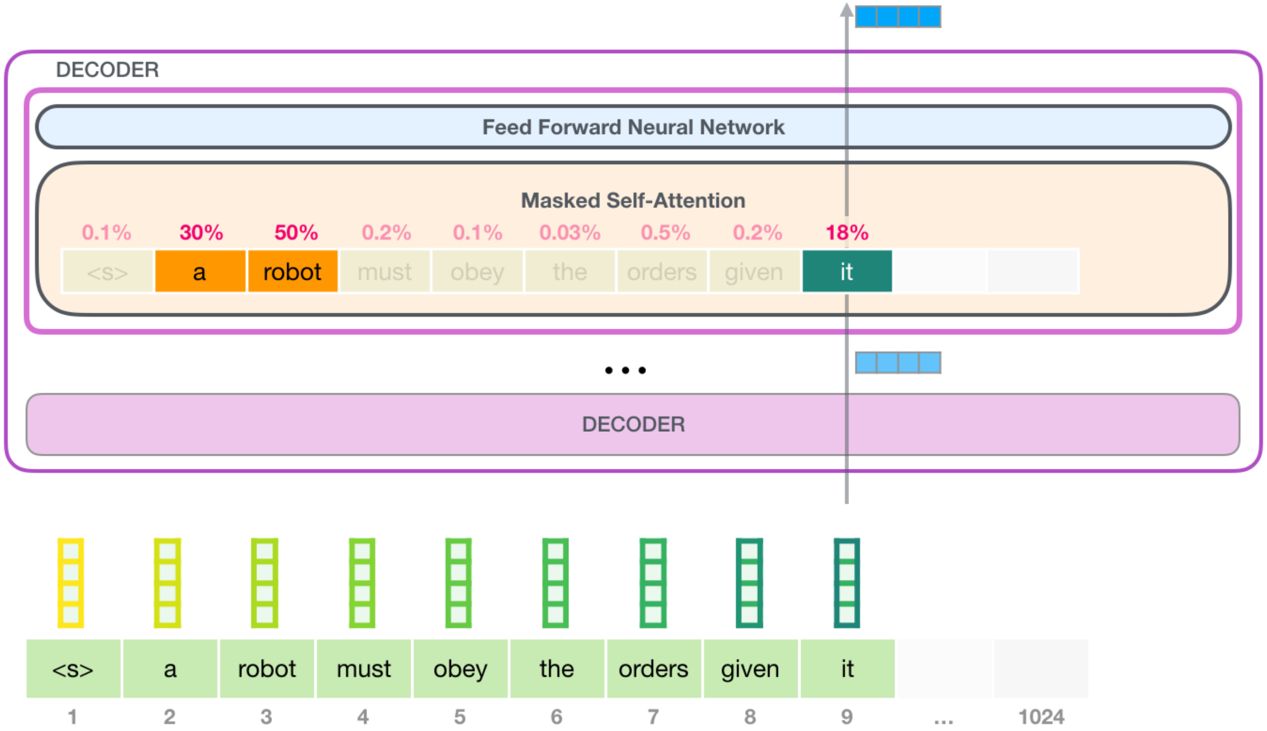
https://playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=circle&regDataset=reg-plane&learningRate=0.03&regularizationRate=0.1&noise=0&networkShape=&seed=0.08645&showTestData=false&discretize=false&percTrainData=90&x=true&y=true&xTimesY=false&xSquared=false&ySquared=false&cosX=false&sinX=false&cosY=false&sinY=false&collectStats=false&problem=classification&initZero=false&hideText=false

Neural Networks for Language Modeling

Each word is associated with a long list of numbers (its "word embedding")

Text is treated as a sequence of words.

A transformer network iteratively generates embeddings of past context using this sequence.



https://jalammar.github.io/illustrated-gpt2/

Training A Language Model

Given a neural network, the first step in training a language model is to make it able to recognize existing web text (which always tends to include wikipedia):

- For the context "Oski the Bear (Oski) is the official", the model is updated to score "mascot" near 1 and other words near 0.
- For the context "Oski was suspended for two weeks in January 1990, for throwing a", the model is updated to score "cake" near 1 and other words near 0.

A critical later step in training a neural language model is to incorporate human ratings of its responses (alignment; reinforcement learning from human feedback).

Sketch of how this can be done:

- Generate two responses for the same context
- Have a human rate which one they prefer
- Train the model parameters so that, given a context, the words scored highly tend to be what humans prefer to read

ChatGPT

ChatGPT is a Software System

Recent innovations that were necessary to build ChatGPT:

- Reinforcement learning algorithms (PPO) that trained the model to provide useful output
- Architectures (Transformer) that trained quickly and found many patterns in language
- Programming environments (Pytorch) that enabled rapid model & algorithm development
- Distributed data processing tools for coordinating many machines and web-scale datasets
- Hardware (NVIDIA) customized for training neural networks

Now that this technology exists, there is a flurry of effort to make it useful:

- Human-computer interaction and integrations with other software systems
- Software and hardware that allow for cost-efficient use of these models at scale

Code Generation

Code Generation

Language models are also trained on all of the open-source code on the web

Three ways that software developers use language models:

- Suggestions (copilot) – input is code context; output is code additions
- Chat – input is code context and a question; output is information
- Agents – input is code context and instructions; output is code changes

Software is often so large that using it all as context is unreliable or infeasible

> "You've got to integrate all of that [AI] with your current repo and your current developer workflow ... That's when you see the productivity."
> – Satya Nadella, CEO of Microsoft, April 29, 2025

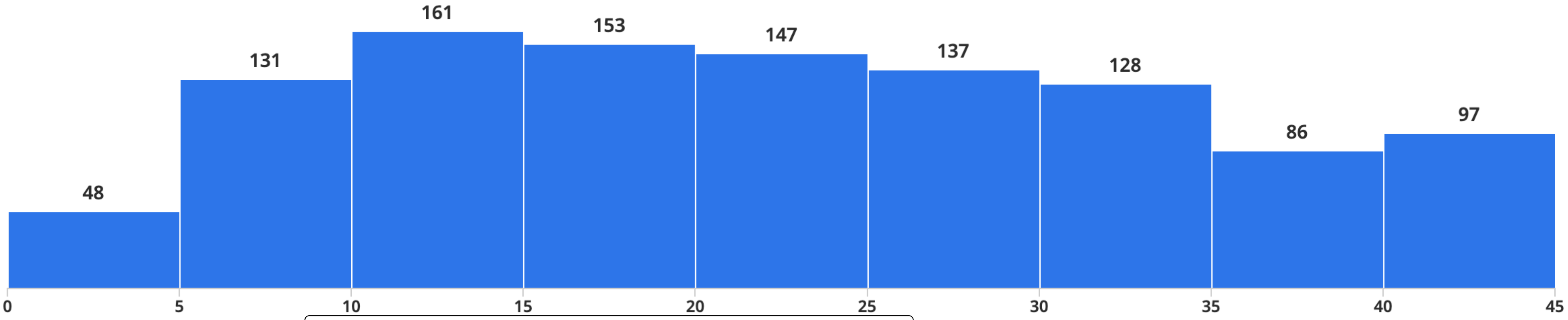
(Demo)

Implications for CS 61A Students

ChatGPT was released to the public in November 2022; Lots of students had found it by Jan 2023

Office hours attendance declined rapidly in Spring 2023 compared to prior semesters

Spring 2023 Midterm 2 scores were unusually low



A typical Midterm 2 median is 60%
21.5/45 is 48%

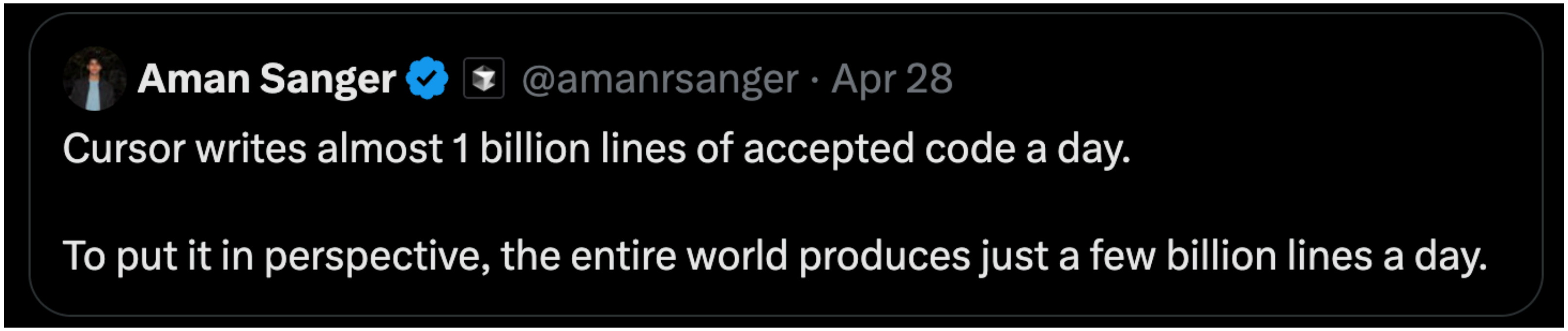
Minimum	Median	Maximum	Mean	Std Dev ?
1.0	21.5	45.0	22.08	11.48

John's Observations (Validated by the Pensieve Team)

1. When very skilled developers use current AI tools, they are able to build applications — especially applications with complex interactive GUIs — really fast, and the resulting software can be stable and highly functional.

2. Beginner programmers are not yet able to build useful novel software using AI. At the moment, the human skill of being able to design, build, and maintain a complex software application is still a necessary condition for actually building one.

3. AI plays a role in development speed, but there are other important factors: software tools such as programming languages, development environments, and application platforms (web frameworks, mobile operating systems, data processing systems) have also accelerated product velocity a lot.



Aman Sanger @amanrsanger · Apr 28

Cursor writes almost 1 billion lines of accepted code a day.

To put it in perspective, the entire world produces just a few billion lines a day.

Take CS 195: Social Implications of Computer Technology