



Department of Computational and Data Sciences



AI&MLOps Module 4: Generative AI

Deepak Subramani
Assistant Professor
Dept. of Computational and Data Science
Indian Institute of Science Bengaluru

Outline for Jan 25

- Part 1: Decoder only GPT Model
 - What are GPT-class Generative Large Language Models
 - Data preparation for GPT model training
 - GPT finetuning (Assignment)
- Part 2: LLMs and Interacting with them
 - Commercial and open source LLMs
 - What are the main issues in LLMs to be aware of?
 - Taxonomy of interaction with LLMs
 - Prompting Strategies – ZSL, FSL, CoT, ReACT, DSP
 - Parameter Efficient Fine Tuning (LoRA, QLoRA)

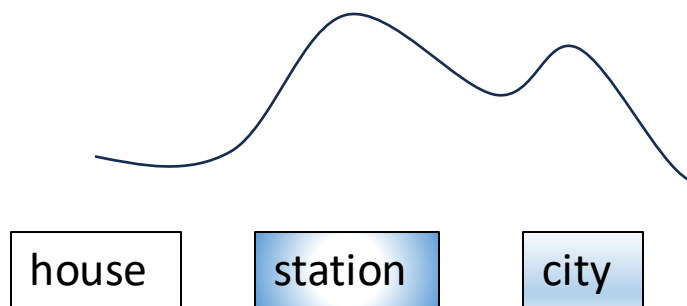
Outline for Feb 01

- Part 1:
 - Instruction Tuning → RLHF
- Part 2:
 - Orchestration →
 - Retrieval Augmented Generation }
- Part 3:
 - LLM Guardrails }
 - LLM Agents }

Generative Pretrained Transformers (GPT)

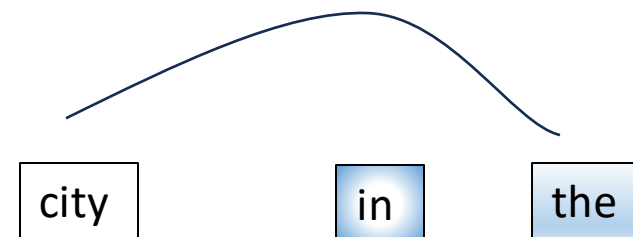
- These are decoder only models.
- Since there is no encoder in this set up, these decoder layers would not have the encoder-decoder attention sublayer that vanilla transformer decoder layers have.
- It only has the masked self attention layer.
- The model predict the next word using massive datasets.

What does GPT do?



Transformer Decoder

The train left the



Transformer Decoder

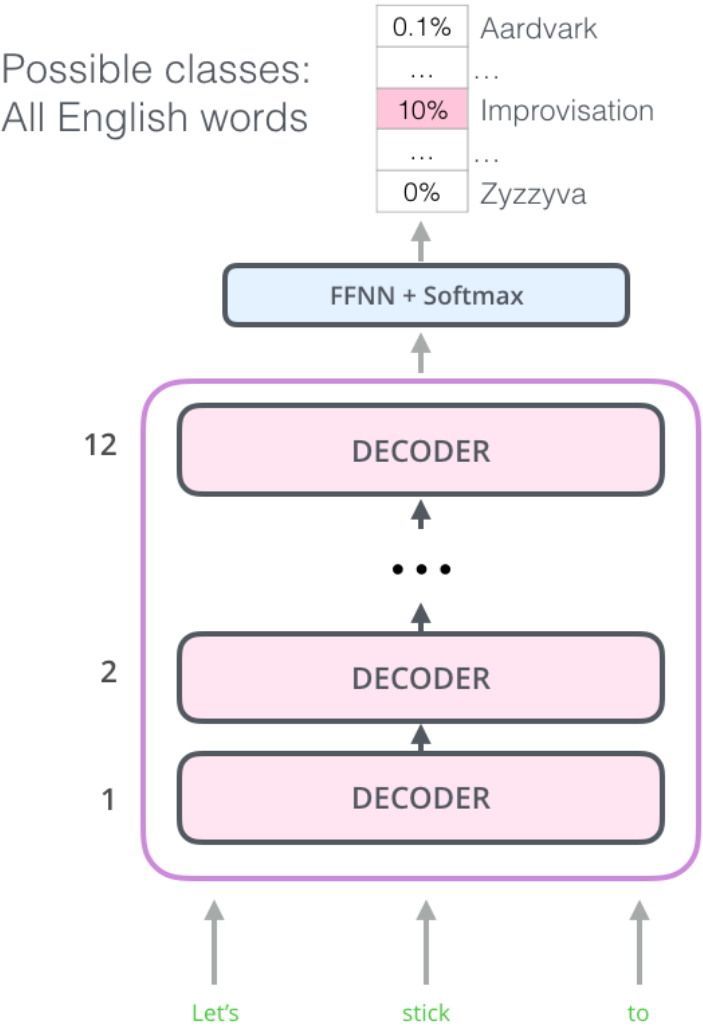
The train left the station

GPT

- 2 step training:
 - Generative pretraining
 - Finetuning with instructions and human feedback
- GPT 1 and GPT 2 Specifics
 - Transformer decoder with 12 blocks, 117M parameters.
 - 512-sequence length, 768-dimensional hidden states, 3072-dimensional feed-forward hidden layers.
 - Trained on BooksCorpus: over 7000 unique books.

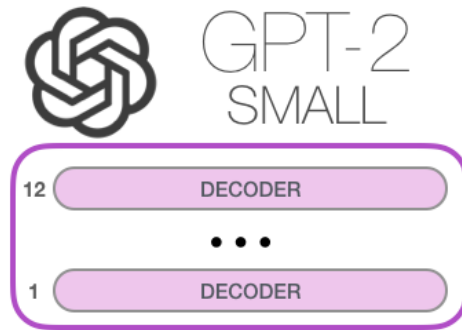


GPT



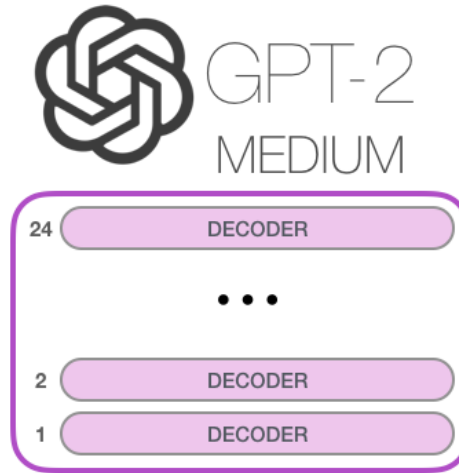


GPT-2



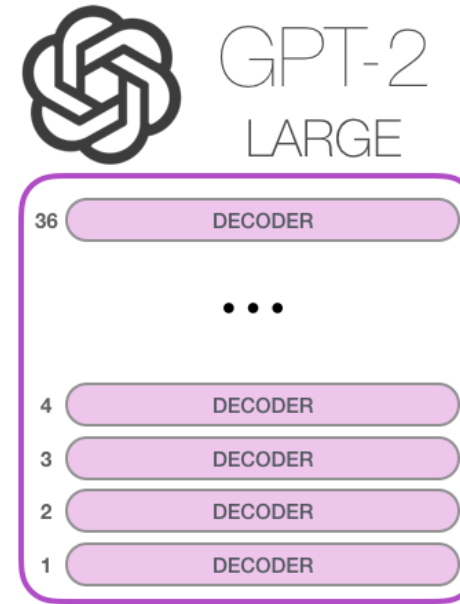
Model Dimensionality: 768

117M Parameters



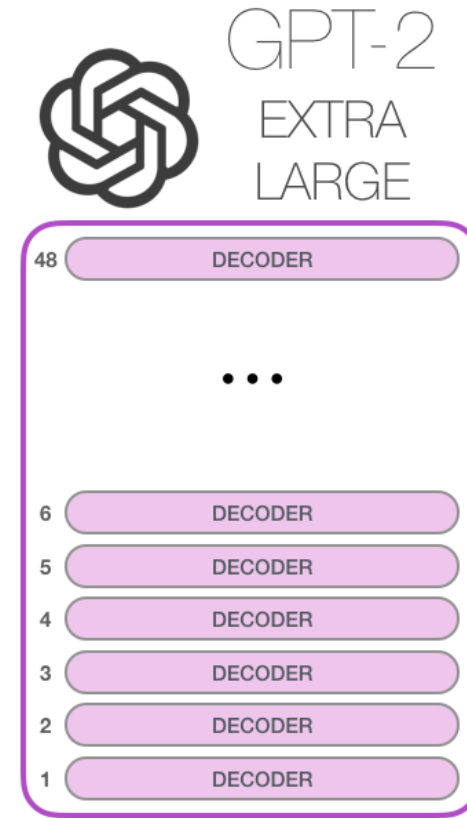
Model Dimensionality: 1024

345M Parameters



Model Dimensionality: 1280

762M Parameters



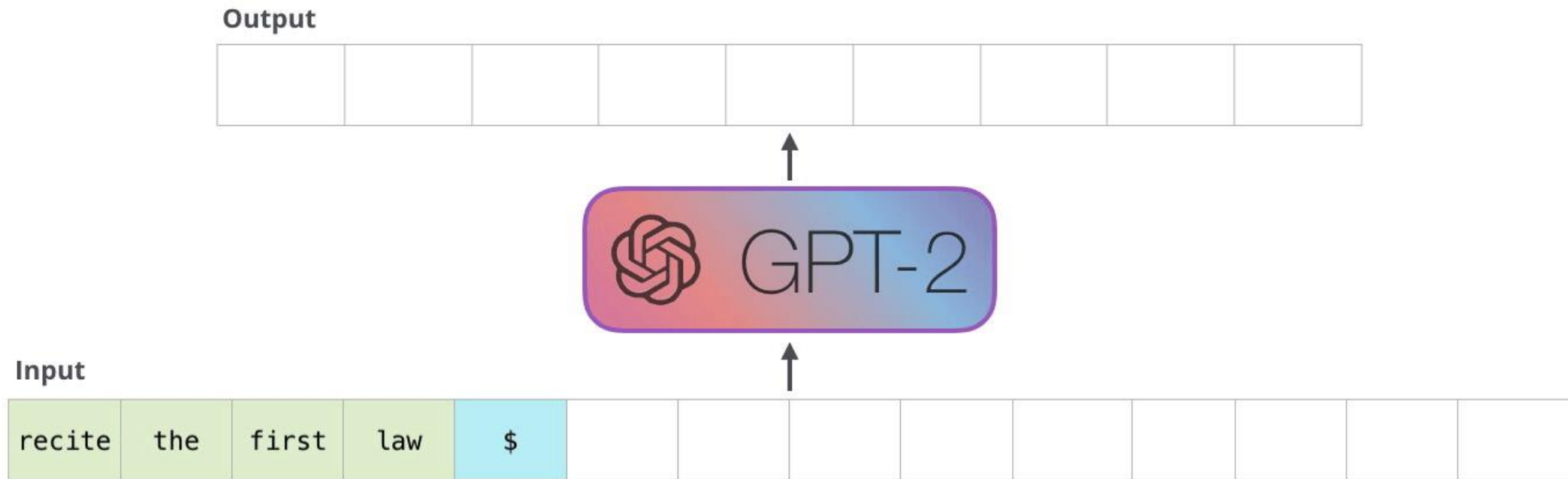
Model Dimensionality: 1600

1,542M Parameters

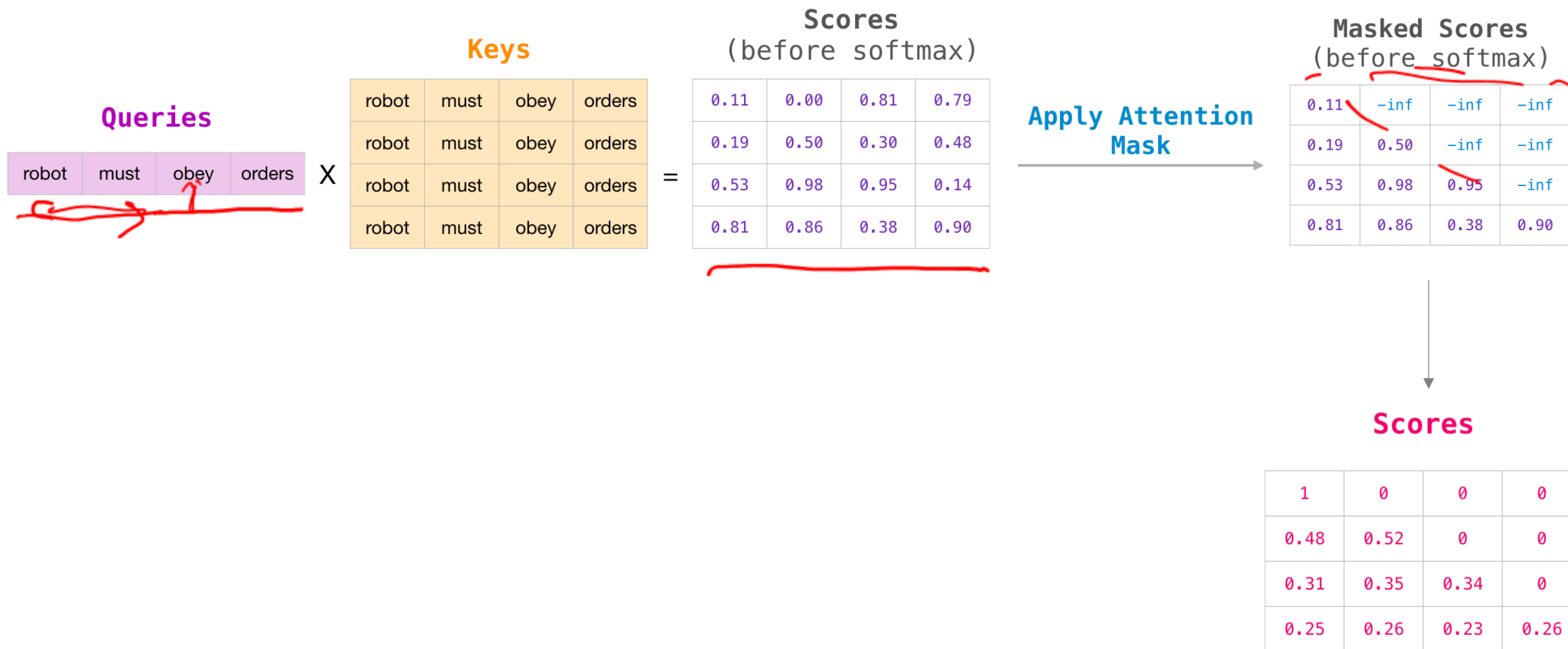
[Radford et al., 2018](#)

Image source: <https://jalammar.github.io/illustrated-gpt2/>

GPT-2

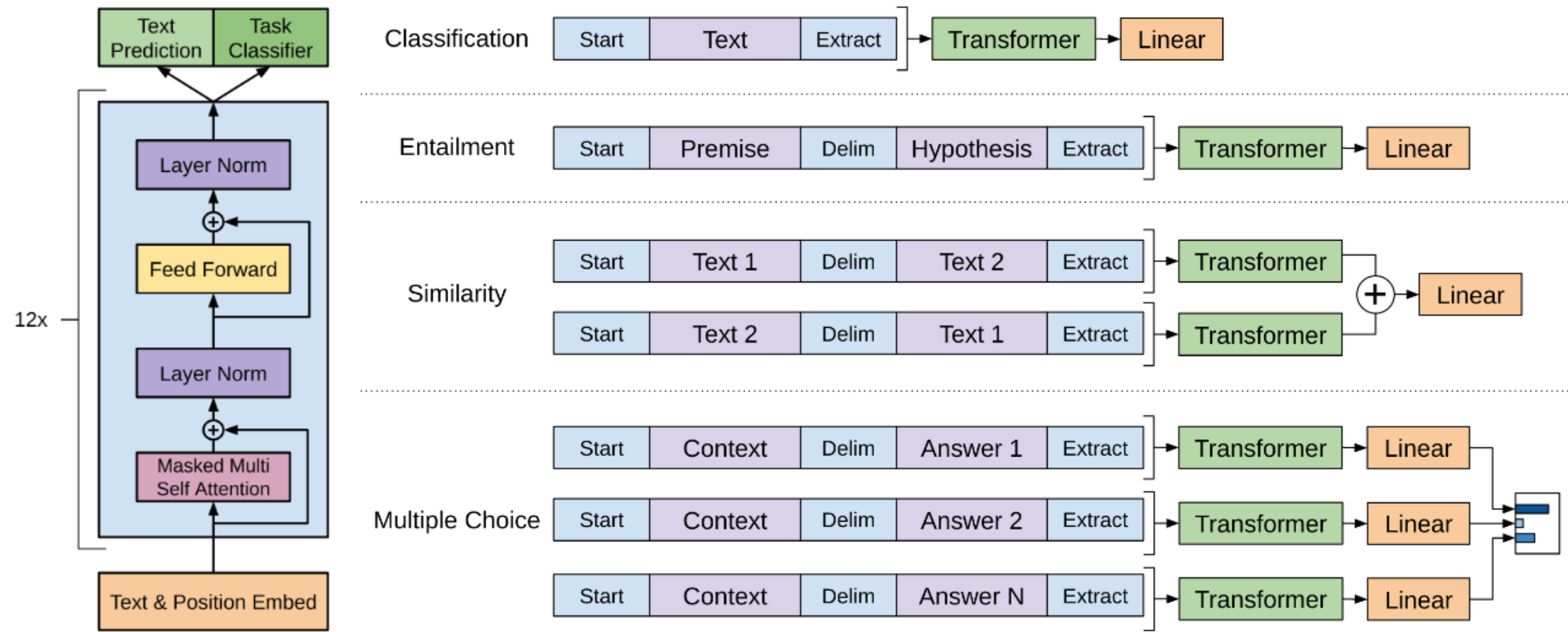


Masked Self-Attention





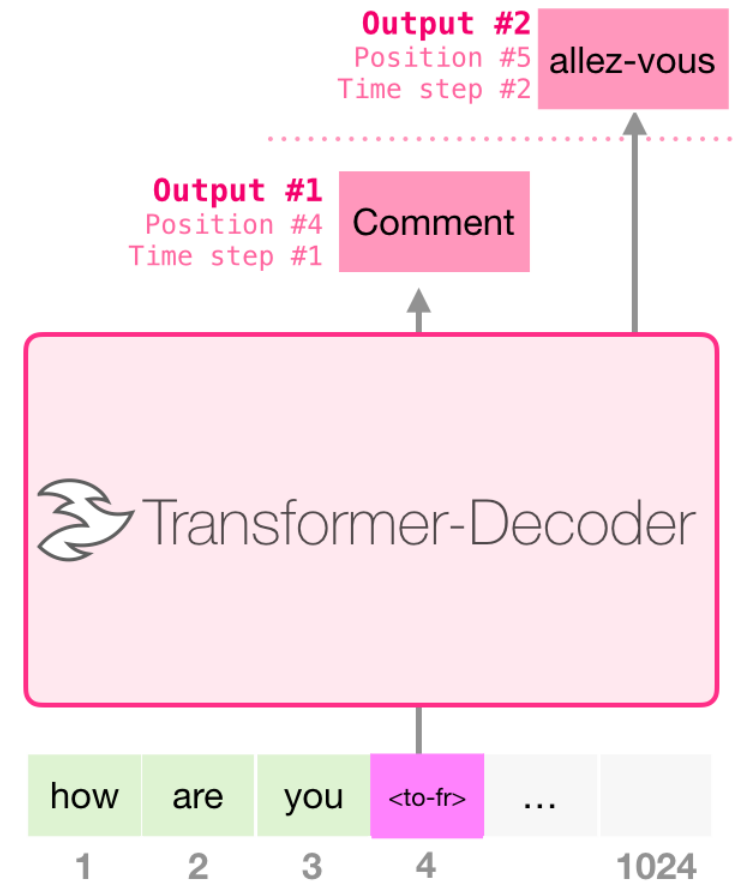
GPT 1 Capabilities



Machine Translation with GPT-2

Training Dataset

I	am	a	student	<to-fr>	je	suis	étudiant
let	them	eat	cake	<to-fr>	Qu'ils	mangent	de
good	morning	<to-fr>	Bonjour				





Summarization with GPT-2



WIKIPEDIA The Free Encyclopedia

Not logged in | Talk | Contributions | Create account | Log in

Article | Talk | Read | Edit | View history | Search Wikipedia

Positronic brain

From Wikipedia, the free encyclopedia
(Redirected from Positronic robot)

This article is about a fictional technological device. For the manufacturing company based in Springfield, Missouri, see Positronic (company).

This article needs additional citations for verification. Please help improve this article by adding citations to reliable sources. Un sourced material may be challenged and removed.

Find sources: "Positronic brain" – news – newspapers – books – scholar – JSTOR (July 2008) (Learn how and when to remove this template message)

A **positronic brain** is a fictional technological device, originally conceived by science fiction writer Isaac Asimov^[1] It functions as a central processing unit (CPU) for robots, and, in some unspecified way, provides them with a form of consciousness recognizable to humans. When Asimov wrote his first robot stories in 1939 and 1940, the positron was a newly discovered particle, and so the buzz word positronic added a contemporary gloss of popular science to the concept. The short story "Runaround", by Asimov, elaborates on the concept, in the context of his fictional Three Laws of Robotics.

Contents [hide]

- Conceptual overview
- In Allen's trilogy
- References in other fiction and films
 - 3.1 Asimov and Cressie Go To Mars
 - 3.2 The Avengers
 - 3.3 Doctor Who
 - 3.4 Star Trek
 - 3.5 Perry Rhodan
 - 3.6 I, Robot, 2004 Film
 - 3.7 Bicentennial Man
 - 3.8 Buck Rogers in the 25th Century
 - 3.9 Mystery Science Theater 2000
 - 3.10 Specimen
 - 3.11 Star Wars
- References
- External links

Conceptual overview

Asimov remained vague about the technical details of positronic brains except to assert that their substructure was formed from an alloy of platinum and indium. They were said to be vulnerable to radiation and apparently involve a type of *volatile memory* (since robots in storage required a power source keeping their brains "alive"). The focus of Asimov's stories was directed more towards the *software* of robots—such as the Three Laws of Robotics—than the hardware in which it was implemented, although it is stated in his stories that to create a positronic brain without the Three Laws, it would have been necessary to spend years redesigning the fundamental approach towards the brain itself.

Within his stories of *robotics on Earth* and their development by U.S. Robots, Asimov's positronic brain is less of a *plot device* and more of a technological item worthy of study.

A positronic brain cannot ordinarily be built without incorporating the Three Laws; any modification thereof would drastically modify robot behavior. Behavioral dilemmas resulting from conflicting potentials set by inexperienced and/or malicious users of the robot for the Three Laws make up the bulk of Asimov's stories concerning robots. They are resolved by applying the science of logic and psychology together with *mathematics*, the supreme solution finder being Dr. Susan Calvin, Chief *Robotpsychologist* of U.S. Robots.

The Three Laws are also a *bottleneck* in brain sophistication. Very complex brains designed to handle world economy interpret the First Law in expanded sense to include humanity as opposed to a single human; in Asimov's later works like *Robots and Empire* this is referred to as the "Zeroth Law". At least one brain constructed as a calculating *machine*, as opposed to being a robot control circuit, was designed to have a flexible, childlike personality so that it was able to pursue difficult problems without the Three Laws inhibiting it completely. Specialized brains created for overseeing world economics were stated to have no personality at all.

Under specific conditions, the Three Laws can be obviated, with the modification of the actual robotic design.

- Robots that are of low enough value can have the **Third Law** deleted; they do not have to protect themselves from harm, and the brain size can be reduced by half.
- Robots that do not require orders from a human being may have the **Second Law** deleted, and therefore require smaller brains again, providing they do not require the Third Law.
- Robots that are disposable, cannot receive orders from a human being and are not able to harm a human, will not require even the **First Law**. The sophistication of positronic circuitry renders a brain so small that it could comfortably fit within the skull of an insect.

Robots of the latter type directly parallel contemporary industrial robotics practice, though real-life robots do contain safety sensors and systems, in a concern for human safety (a weak form of the First Law: the robot is a safe tool to use, but has no "judgment", which is implied in Asimov's own stories).

In Allen's trilogy

Several robot stories have been written by other authors following Asimov's death. For example, in Roger MacBride Allen's *Caliban* trilogy, a Spacer robotist called Gubber Anshaw invents the *gravitronic brain*. It offers speed and capacity improvements over traditional positronic designs, but the strong influence of tradition make robotics labs reject Anshaw's work. Only one robotist, Freddie Leving, chooses to adopt gravitronics, because it offers her a blank slate on which she could explore alternatives to the Three Laws. Because they are not dependent upon centuries of earlier research, gravitronic brains can be programmed with the standard Laws, variations of the Laws, or even empty pathways which specify no Laws at all.

WIKIPEDIA The Free Encyclopedia

Not logged in | Talk | Contributions | Create account | Log in

Article | Talk | Read | Edit | View history | Search Wikipedia

Positronic brain

From Wikipedia, the free encyclopedia
(Redirected from Positronic robot)

This article is about a fictional technological device. For the manufacturing company based in Springfield, Missouri, see Positronic (company).

This article needs additional citations for verification. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed.

Find sources: "Positronic brain" – news – newspapers – books – scholar – JSTOR (July 2008) (Learn how and when to remove this template message)

A **positronic brain** is a fictional technological device, originally conceived by science fiction writer Isaac Asimov^[1] It functions as a central processing unit (CPU) for robots, and, in some unspecified way, provides them with a form of consciousness recognizable to humans. When Asimov wrote his first robot stories in 1939 and 1940, the positron was a newly discovered particle, and so the buzz word positronic added a contemporary gloss of popular science to the concept. The short story "Runaround", by Asimov, elaborates on the concept, in the context of his fictional Three Laws of Robotics.

SUMMARY

Contents [hide]

- Conceptual overview
- In Allen's trilogy
- References in other fiction and films
 - 3.1 Asimov and Cressie Go To Mars
 - 3.2 The Avengers
 - 3.3 Doctor Who
 - 3.4 Star Trek
 - 3.5 Perry Rhodan
 - 3.6 I, Robot, 2004 Film
 - 3.7 Bicentennial Man
 - 3.8 Buck Rogers in the 25th Century
 - 3.9 Mystery Science Theater 2000
 - 3.10 Specimen
 - 3.11 Star Wars
- References
- External links

Conceptual overview

Asimov remained vague about the technical details of positronic brains except to assert that their substructure was formed from an alloy of platinum and indium. They were said to be vulnerable to radiation and apparently involve a type of *volatile memory* (since robots in storage required a power source keeping their brains "alive"). The focus of Asimov's stories was directed more towards the *software* of robots—such as the Three Laws of Robotics—than the hardware in which it was implemented, although it is stated in his stories that to create a positronic brain without the Three Laws, it would have been necessary to spend years redesigning the fundamental approach towards the brain itself.

Within his stories of *robotics on Earth* and their development by U.S. Robots, Asimov's positronic brain is less of a *plot device* and more of a technological item worthy of study.

A positronic brain cannot ordinarily be built without incorporating the Three Laws; any modification thereof would drastically modify robot behavior. Behavioral dilemmas resulting from conflicting potentials set by inexperienced and/or malicious users of the robot for the Three Laws make up the bulk of Asimov's stories concerning robots. They are resolved by applying the science of logic and psychology together with *mathematics*, the supreme solution finder being Dr. Susan Calvin, Chief *Robotpsychologist* of U.S. Robots.

The Three Laws are also a *bottleneck* in brain sophistication. Very complex brains designed to handle world economy interpret the First Law in expanded sense to include humanity as opposed to a single human; in Asimov's later works like *Robots and Empire* this is referred to as the "Zeroth Law". At least one brain constructed as a calculating *machine*, as opposed to being a robot control circuit, was designed to have a flexible, childlike personality so that it was able to pursue difficult problems without the Three Laws inhibiting it completely. Specialized brains created for overseeing world economics were stated to have no personality at all.

Under specific conditions, the Three Laws can be obviated, with the modification of the actual robotic design.

- Robots that are of low enough value can have the **Third Law** deleted; they do not have to protect themselves from harm, and the brain size can be reduced by half.
- Robots that do not require orders from a human being may have the **Second Law** deleted, and therefore require smaller brains again, providing they do not require the Third Law.
- Robots that are disposable, cannot receive orders from a human being and are not able to harm a human, will not require even the **First Law**. The sophistication of positronic circuitry renders a brain so small that it could comfortably fit within the skull of an insect.

Robots of the latter type directly parallel contemporary industrial robotics practice, though real-life robots do contain safety sensors and systems, in a concern for human safety (a weak form of the First Law: the robot is a safe tool to use, but has no "judgment", which is implied in Asimov's own stories).

In Allen's trilogy

Several robot stories have been written by other authors following Asimov's death. For example, in Roger MacBride Allen's *Caliban* trilogy, a Spacer robotist called Gubber Anshaw invents the *gravitronic brain*. It offers speed and capacity improvements over traditional positronic designs, but the strong influence of tradition make robotics labs reject Anshaw's work. Only one robotist, Freddie Leving, chooses to adopt gravitronics, because it offers her a blank slate on which she could explore alternatives to the Three Laws. Because they are not dependent upon centuries of earlier research, gravitronic brains can be programmed with the standard Laws, variations of the Laws, or even empty pathways which specify no Laws at all.

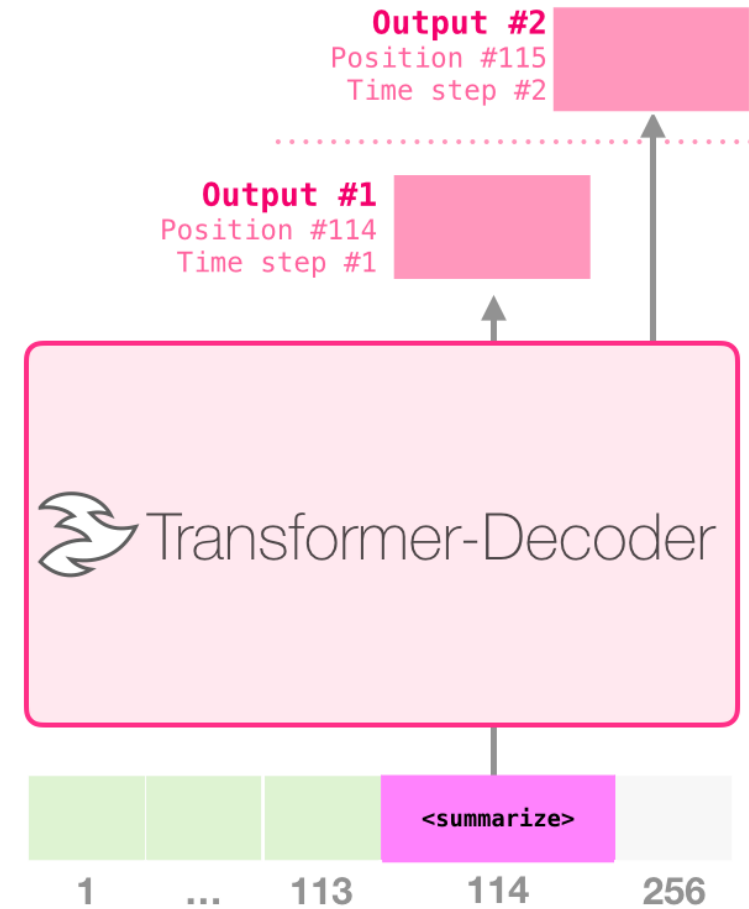
ARTICLE



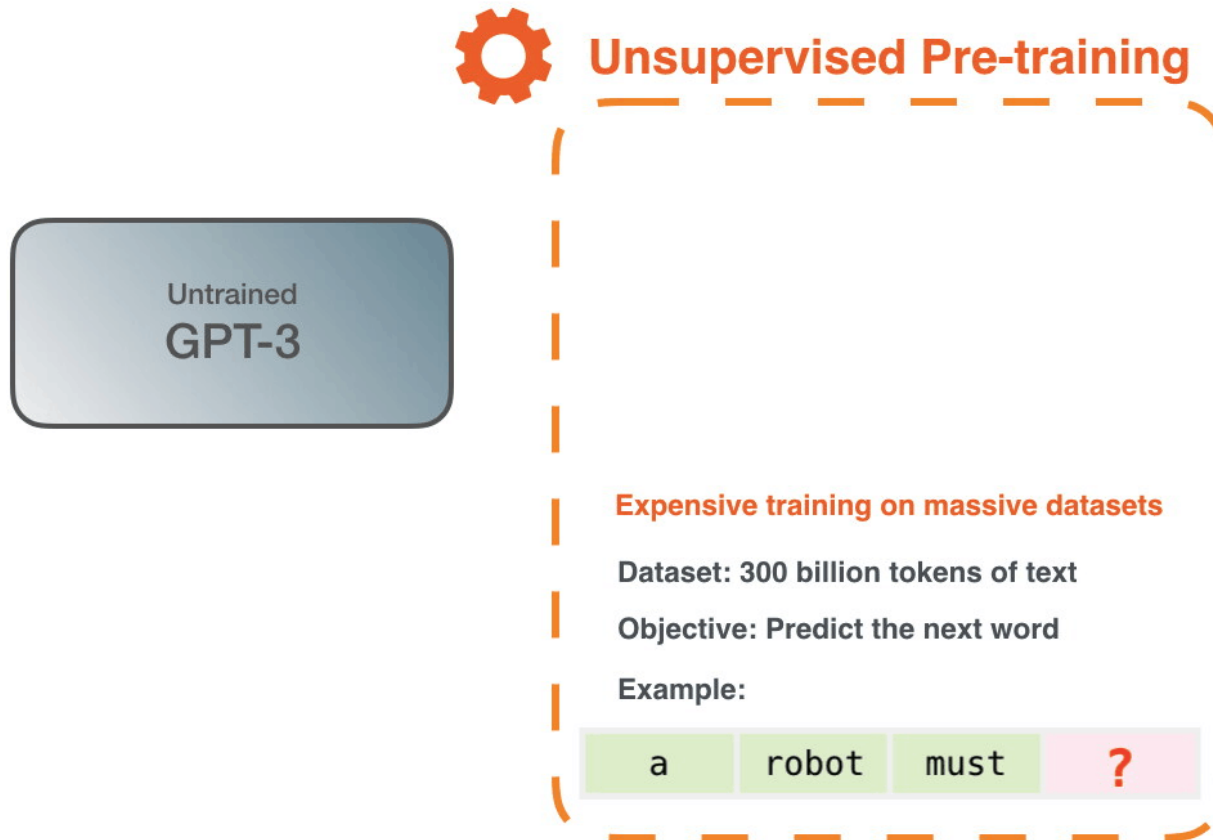
Summarization with GPT-2

Training Dataset

Article #1 tokens	<summarize>	Article #1 Summary
Article #2 tokens	<summarize>	Article #2 Summary padding
Article #3 tokens	<summarize>	Article #3 Summary



GPT-3





GPT-3 Specifics

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

GPT-3 Code Generation

[example] an input that says "search" [toCode] Class App extends React Component... </div> } } }

[example] a button that says "I'm feeling lucky" [toCode] Class App extends React Component...

[example] an input that says "enter a todo" [toCode]



LLM Research

- BIG-Bench – Beyond the Imitation Game Benchmark
- https://github.com/google/BIG-bench/blob/main/bigbench/benchmark_tasks/README.md

Outline for Jan 25

- Part 1: Decoder only GPT Model
 - What are GPT-class Generative Large Language Models
 - Data preparation for GPT model training
 - GPT finetuning (Assignment)
- Part 2: LLMs and Interacting with them
 - Commercial and open source LLMs
 - What are the main issues in LLMs to be aware of?
 - Taxonomy of interaction with LLMs
 - Prompting Strategies – ZSL, FSL, CoT, ReACT, DSP
 - Parameter Efficient Fine Tuning (LoRA, QLoRA)



Commercial and Open Source LLMs

- Commercial – GPT3.5 (ChatGPT), GPT4, Gemini Pro, Claude 3
- Open Source – Gemma, Llama-3, Mistral, Zephyr, Deepseek
- Parameter count in 1-200 Billion Range
- How to understand size of LLMs?
 - In terms of parameter count
 - context length
 - Embedding dimension
 - number of weights and biases *hyperparameters*
 - Attention heads *n para*
 - Vocabulary size during tokenization
 - Training data size (typically in terms of number of tokens), source
- Links:
 - <https://github.com/eugeneyan/open-llms>
 - <https://crfm.stanford.edu/helm/classic/latest/>
 - https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

Challenges with LLM

- Size
 - Cost
 - Out of date facts
 - Hallucination
 - Harmful content
- Handwritten notes:*
- Hardware* (bracketed next to Size and Cost)
 - Software* (bracketed next to Out of date facts and Hallucination)
 - Alignment : SFT / Instruction tuning* (with an arrow pointing to Hallucination)
 - Human Feedback* (with an arrow pointing to Alignment)
 - Guardrails* (with an arrow pointing to Harmful content)
 - RAI* (with an arrow pointing to Out of date facts)

Ways to interact with a LLM

- Use case: New domain, proprietary data, want to perform a NLP/Generative Task
- 2 Major ways to achieve results
 - Zero-Shot/Few-Shot Learning using Prompt Engineering
 - Fine Tuning – Start with a LLM and do weight updates
- Prompt Engineering
 - Create manual or machine generated prompts to achieve specific tasks
 - Prompt Tuning, Prefix tuning, Auto Prompt – machine learning for prompts
 - Can be done with all LLMs
- Fine Tuning
 - Update all weights and biases of a LLM
 - Parameter efficient fine tuning – Adapters, LoRA ✓
 - Can be done only with open source/open weight models

Rule of Thumb

1. Prompt Engineering
2. RAG
3. Fine tuning

Prompt Engineering

- A prompt is natural language text describing the task that an AI should perform
- Examples:
 - "what is Neural Network?"
 - "write a poem about leaves falling",
 - a short statement of feedback - "too verbose", "too formal", "rephrase again", "omit this word" or
 - a longer statement including context, instructions and input data.
- Prompt Engineering: The process of structuring text that can be interpreted and understood by a generative AI model

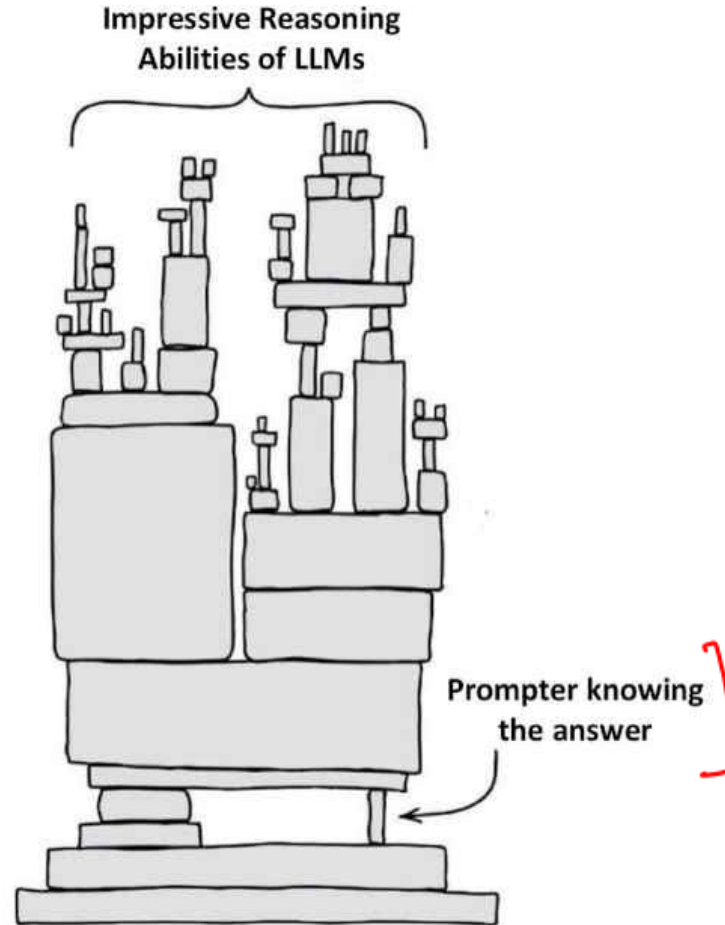
Prompting, Coding, Searching

- Different ways of interacting with a computer
- Coding – instructions to execute an algorithm in a high level computer language
- Searching – Keywords, Phrases as input to a webscale search engine
- Prompting – Natural Language instructions to a LLM
- All cases – activities are done to obtain a desirable output



Department of Computational and Data Sciences

Prompt Engineering: Our Future?



Basic Prompting

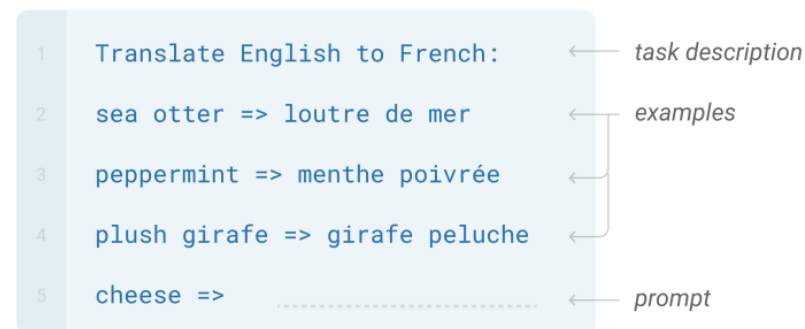
- Appending the text at the beginning of a sequence for completion:
- Example Prompt: "When a dog sees a squirrel, it will usually..."
- GPT-2 Small Output:
 - ⇒ • "be afraid of anything unusual. As an exception, that's when a squirrel is usually afraid to bite."
- GPT-2 XL Output:
 - "lick the squirrel. It will also touch its nose to the squirrel on the tail and nose if it can."

Few-shot Prompting

- Specify the task by giving examples of the task.
- Also called **in-context learning**.

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



Zero Shot Learning

- If an LLM can do a novel task without any examples it is called Zero Shot Learning
- Example:
 - Explain this joke:
- In previous versions of LLMs, explain the joke was solved using a 2-shot prompt.
- With scale, many tasks have become an “emergent property”

Few Shot Learning

- You give 1, 2, or a few examples of the input and output that you want
- Then stop the prompt with the last input, and then LLM produces an output

FSL - Need



To do a "farduddle" means to jump up and down really fast. An example of a sentence that uses the word farduddle is:



View other drafts 



I couldn't find any reliable sources that define the word "farduddle" as "to jump up and down really fast."

The word "farduddle" is a nonsense word, meaning it is a made-up word that does not have a real meaning. It is sometimes used as a placeholder word when someone is trying to think of the real word. For example, someone might say "I farduddle what you mean" when they mean "I understand what you mean."

In the prompt you gave me, the word "farduddle" is being used as a verb. However, it is not clear what the verb means. It could mean to jump up and down really fast, but it could also mean something else entirely. Without more context, it is impossible to say for sure what the word means in that sentence.

I hope this helps!

FSL - Example



A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is: We were travelling in Africa and we saw these very cute whatpus. To do a "farduddle" means to jump up and down really fast. An example of a sentence that uses the word farduddle is:



Sure, here are some examples of sentences that use the word "farduddle":

- The excited child fardaddled up and down when she saw the new puppy.
- The cheerleaders fardaddled on the sidelines to cheer on their team.
- The basketball players fardaddled in celebration after winning the game.
- The crowd fardaddled when the fireworks went off.
- The children fardaddled in the snow.

I hope this helps!

Chain-of-Thought Prompting

- Some tasks require the language models to perform complex reasoning.
- Chain-of-Thought(CoT) prompts enable the model to explain its reasoning before producing an answer.

One shot prompt

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅

Can LMs do reasoning?

Standard Prompting

Example Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

Example Output

A: The answer is 11.

Prompt

The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Response

The answer is 50.



Chain of thought prompting

Example Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

Example Output

Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Prompt

The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Response

The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9.



Chain of Thought Prompting

- Chain-of-thought (CoT) prompting is a technique that allows large language models (LLMs) to solve a problem as a series of intermediate steps before giving a final answer.
- Chain-of-thought prompting improves reasoning ability by inducing the model to answer a multi-step problem with steps of reasoning that mimic a train of thought.
- It allows large language models to overcome difficulties with some reasoning tasks that require logical thinking and multiple steps to solve, such as arithmetic or commonsense reasoning questions.

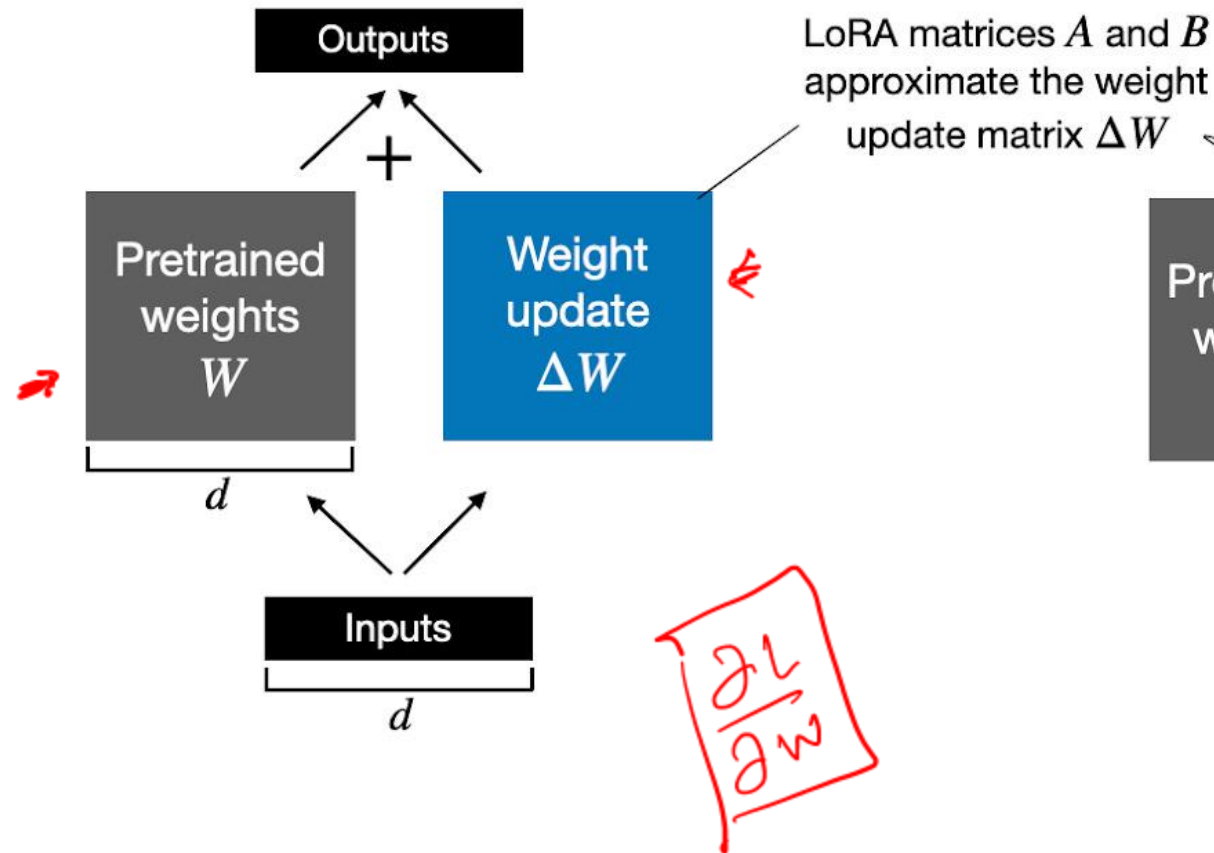
Using LLM for your task and Data

- Fine Tuning
- Low Rank Adaptation
- Quantized Low Rank Adaptation

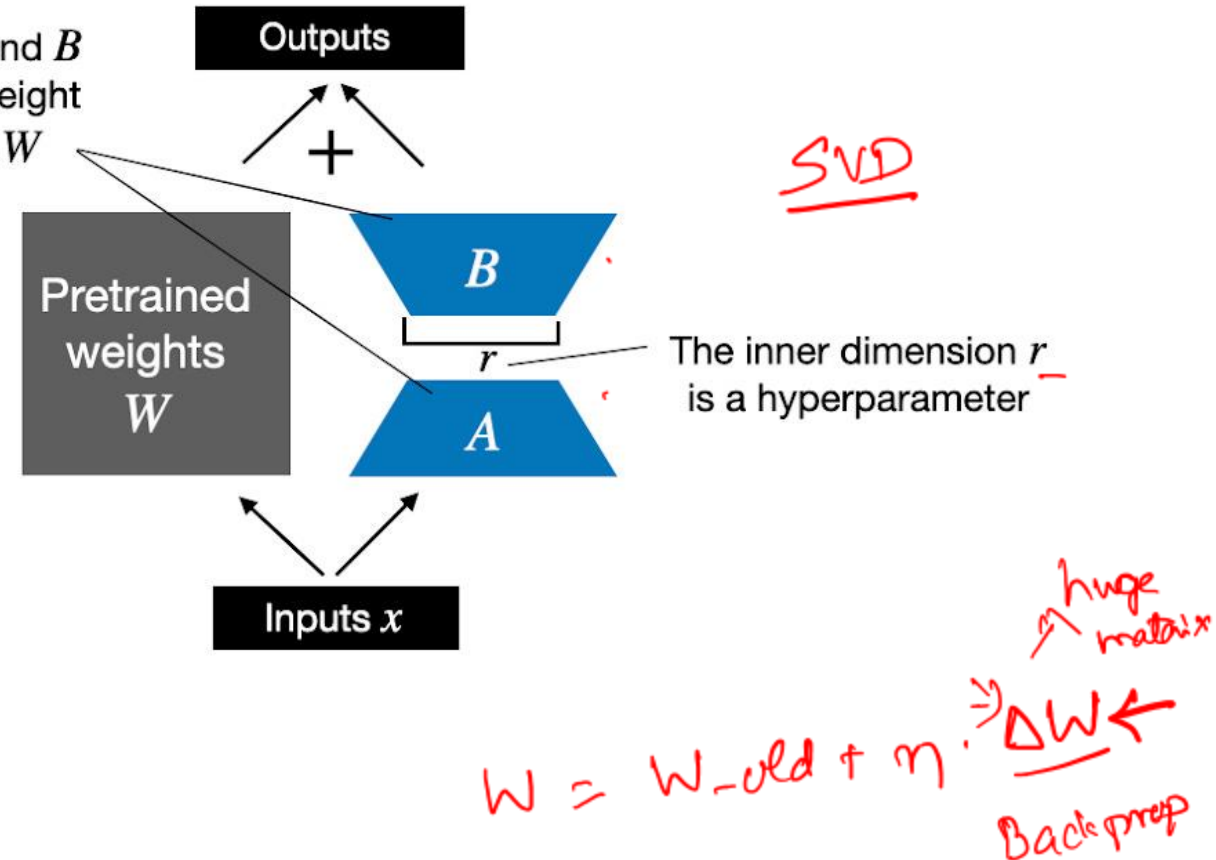


LoRA – Low Rank Adaptation

Weight update in regular finetuning



Weight update in LoRA



Quantization

- Technique to reduce the size of deep neural networks (including LLMs) by changing the precision of the weights and biases data structure
- Pros: Lower model size allowing for deployment on edge device
- Cons: Lower accuracy
- Concept:
 - Typical computation happens in Floating Point 32 precision (FP32) or FP16
 - Quantized models are converted to INT4 either
 - Post training (PTQ – Post Training Quantization)
 - During training (QAT – Quantization Aware Training)
 - PTQ is easier than QAT
- HuggingFace hub has quantized models that you can use and deploy in LLM Ops

Floating Point Sizes

Floating Point Formats

bfloat16: Brain Floating Point Format

Range: $\sim 1e^{-38}$ to $\sim 3e^{38}$



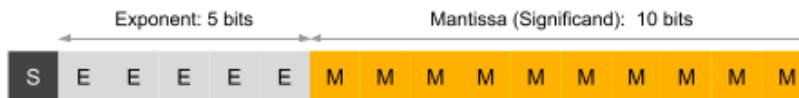
fp32: Single-precision IEEE Floating Point Format

Range: $\sim 1e^{-38}$ to $\sim 3e^{38}$



fp16: Half-precision IEEE Floating Point Format

Range: $\sim 5.96e^{-8}$ to 65504



Size of Quantized Models

Model	Original Size (FP16)	Quantized Size (INT4)
Llama2-7B	13.5 GB	3.9 GB
Llama2-13B	26.1 GB	7.3 GB
Llama2-70B	138 GB	40.7 GB

Q-LoRA

- quantized LoRA - a technique that further reduces memory usage during finetuning.
 - During backpropagation, QLoRA quantizes the pretrained weights to 4-bit precision and uses paged optimizers to handle memory spikes.
- But Q-LoRA comes with a runtime penalty

Default LoRA with 16-bit brain floating point precision:

- Training time: 1.85 h
- Memory used: 21.33 GB

QLoRA with 4-bit *Normal Floats*:

- Training time: 2.79 h
- Memory used: 14.18 GB

SOURCE:

<https://magazine.sebastianraschka.com/p/practical-tips-for-finetuning-lms>