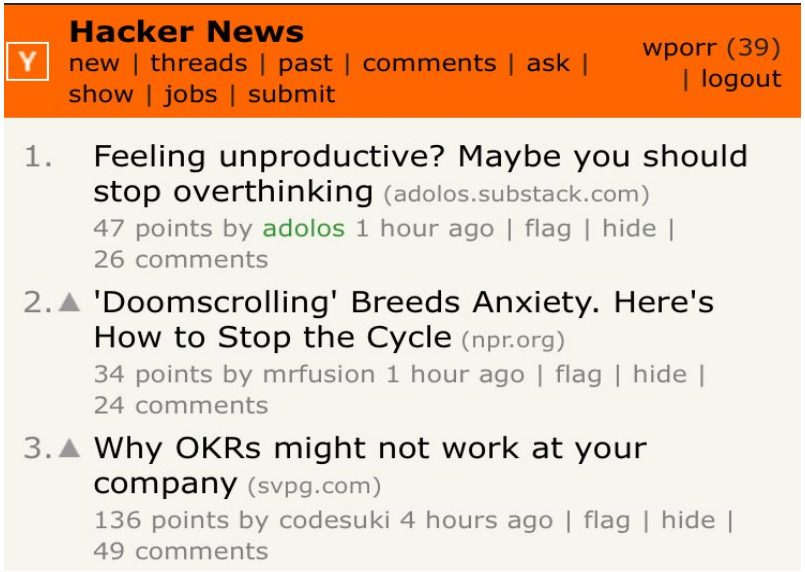
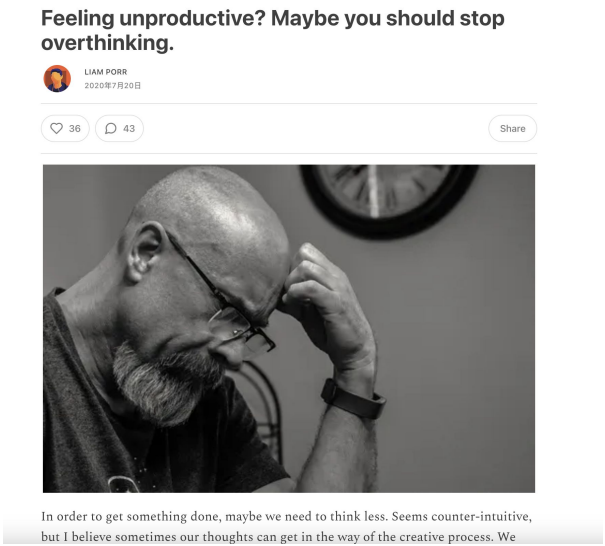
Evolution of Generative Pre-trained Transformers: From GPT-1 to GPT-3

About GPT-3: Fun Facts

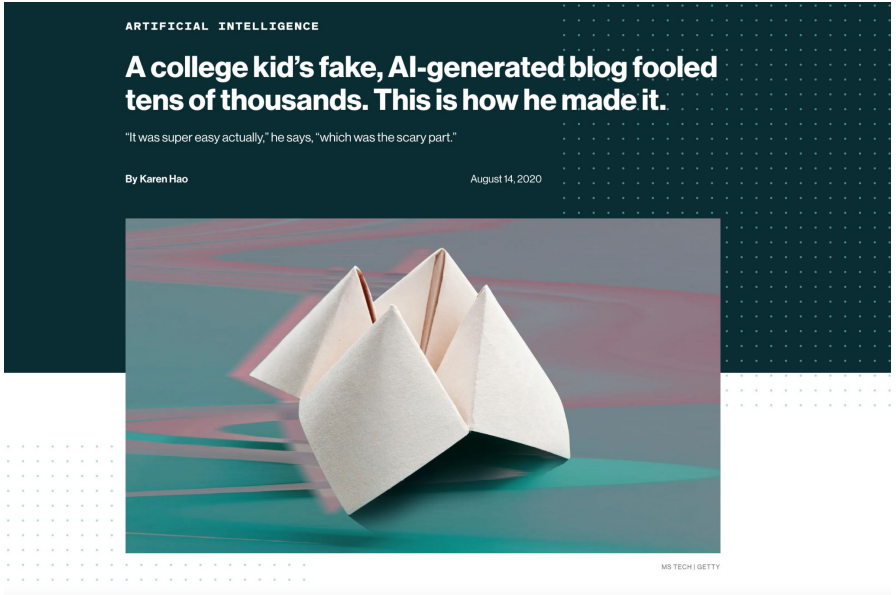
In 2020, A blog post titled *Feeling Unproductive? Maybe you should stop overthinking* reached rank1 on Hacker News (a popular tech news site)





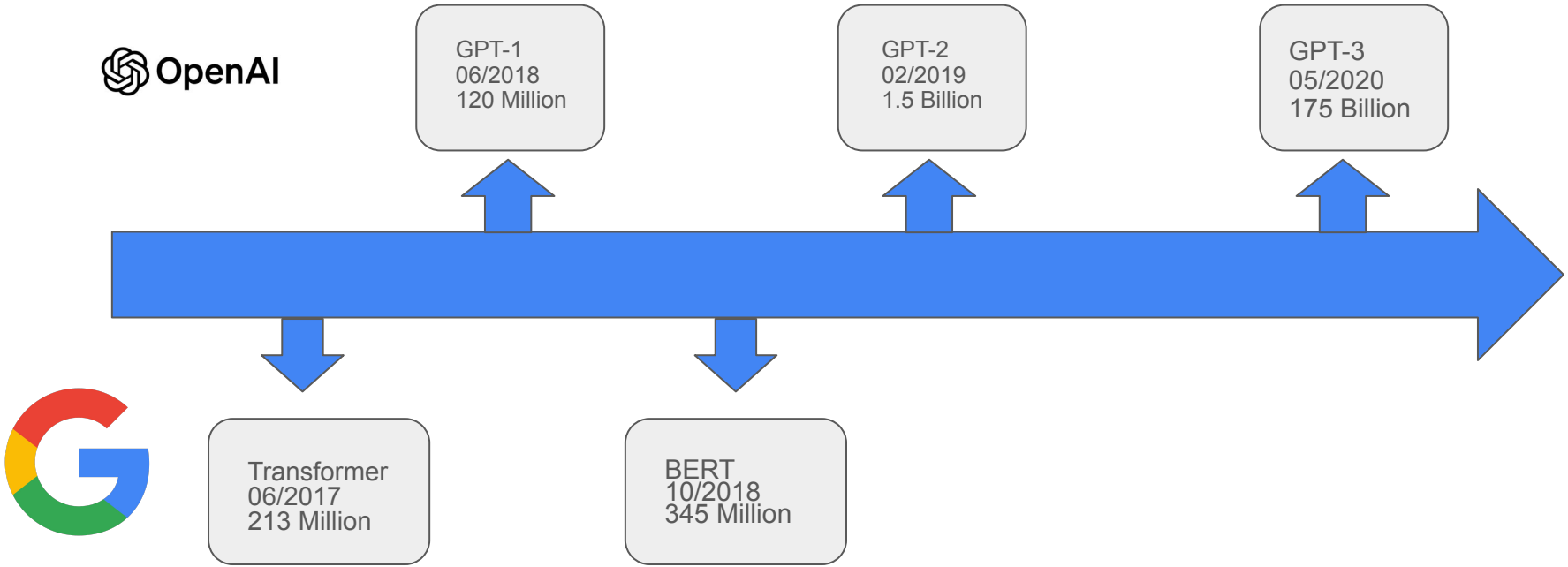
About GPT-3: Fun Facts

- MIT Tech Review revealed that this blog post was generated by college student using GPT-3 !
- GPT-3 showed its ability to generate human-like, high-quality articles



At the start of the week, Liam Porr had only heard of GPT-3. By the end, the college student had used the AI model to produce an entirely fake blog under a fake name.

GPT Series: Timeline and history



GPT-1:
Improving Language Understanding
by Generative Pre-Training

Improving Language Understanding by Generative Pre-Training

Alec Radford
OpenAI
alec@openai.com

Karthik Narasimhan
OpenAI
karthikn@openai.com

Tim Salimans
OpenAI
tim@openai.com

Ilya Sutskever
OpenAI
ilyasu@openai.com

Abstract

Natural language understanding comprises a wide range of diverse tasks such as textual entailment, question answering, semantic similarity assessment, and document classification. Although large unlabeled text corpora are abundant, labeled data for learning these specific tasks is scarce, making it challenging for discriminatively trained models to perform adequately. We demonstrate that large gains on these tasks can be realized by *generative pre-training* of a language model on a diverse corpus of unlabeled text, followed by *discriminative fine-tuning* on each specific task. In contrast to previous approaches, we make use of task-aware input transformations during fine-tuning to achieve effective transfer while requiring minimal changes to the model architecture. We demonstrate the effectiveness of our approach on a wide range of benchmarks for natural language understanding. Our general task-agnostic model outperforms discriminatively trained models that use architectures specifically crafted for each task, significantly improving upon the state of the art in 9 out of the 12 tasks studied. For instance, we achieve absolute improvements of 8.9% on commonsense reasoning (Stories Cloze Test), 5.7% on question answering (RACE), and 1.5% on textual entailment (MultiNLI).

1 Introduction

The ability to learn effectively from raw text is crucial to alleviating the dependence on supervised learning in natural language processing (NLP). Most deep learning methods require substantial amounts of manually labeled data, which restricts their applicability in many domains that suffer from a dearth of annotated resources [61]. In these situations, models that can leverage linguistic information from unlabeled data provide a valuable alternative to gathering more annotation, which can be time-consuming and expensive. Further, even in cases where considerable supervision is available, learning good representations in an unsupervised fashion can provide a significant performance boost. The most compelling evidence for this so far has been the extensive use of pre-trained word embeddings [10, 39, 42] to improve performance on a range of NLP tasks [8, 11, 26, 45].

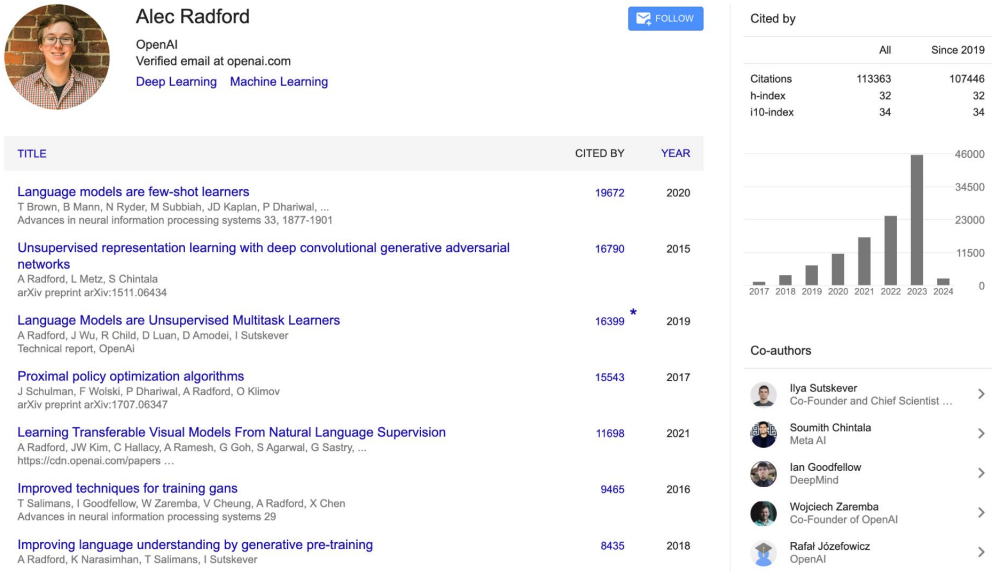
Leveraging more than word-level information from unlabeled text, however, is challenging for two main reasons. First, it is unclear what type of optimization objectives are most effective at learning text representations that are useful for transfer. Recent research has looked at various objectives such as language modeling [44], machine translation [38], and discourse coherence [22], with each method outperforming the others on different tasks.[1] Second, there is no consensus on the most effective way to transfer these learned representations to the target task. Existing techniques involve a combination of making task-specific changes to the model architecture [43, 44], using intricate learning schemes [21] and adding auxiliary learning objectives [50]. These uncertainties have made it difficult to develop effective semi-supervised learning approaches for language processing.

[1] https://gluebenchmark.com/leaderboard

Preprint. Work in progress.

GPT-1 Authors: First author

Alec Radford, Previous work include: DCGAN, PPO, reinforcement learning related work



Alec Radford

FOLLOW

OpenAI
Verified email at openai.com
Deep Learning Machine Learning

TITLE	CITED BY	YEAR
Language models are few-shot learners T Brown, B Mann, N Ryder, M Subbiah, JD Kaplan, P Dhariwal, ... Advances in neural information processing systems 33, 1877-1901	19672	2020
Unsupervised representation learning with deep convolutional generative adversarial networks A Radford, L Metz, S Chintala arXiv preprint arXiv:1511.06434	16790	2015
Language Models are Unsupervised Multitask Learners A Radford, J Wu, R Child, D Luan, D Amodei, I Sutskever Technical report, OpenAI	16399 *	2019
Proximal policy optimization algorithms J Schulman, F Wolski, P Dhariwal, A Radford, O Klimov arXiv preprint arXiv:1707.06347	15543	2017
Learning Transferable Visual Models From Natural Language Supervision A Radford, JW Kim, C Hallacy, A Ramesh, G Goh, S Agarwal, G Sastry, ... https://cdn.openai.com/papers …	11698	2021
Improved techniques for training gans T Salimans, I Goodfellow, W Zaremba, V Cheung, A Radford, X Chen Advances in neural information processing systems 29	9465	2016
Improving language understanding by generative pre-training A Radford, K Narasimhan, T Salimans, I Sutskever	8435	2018

Cited by

	All	Since 2019
Citations	113363	107446
h-index	32	32
i10-index	34	34

Co-authors

- Ilya Sutskever — Co-Founder and Chief Scientist …
- Soumith Chintala — Meta AI
- Ian Goodfellow — DeepMind
- Wojciech Zaremba — Co-Founder of OpenAI
- Rafał Józefowicz — OpenAI

GPT-1 Authors: Corresponding Author (Team Leader)

Ilya Sutskever,

Chief Scientist of OpenAI

Author of **AlexNet (start of the AI wave)**

Ilya Sutskever

FOLLOW

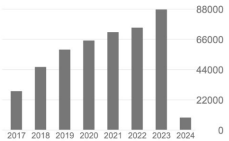
Co-Founder and Chief Scientist of OpenAI
Verified email at openai.com - Homepage

Machine Learning Neural Networks Artificial Intelligence Deep Learning

TITLE	CITED BY	YEAR
Imagenet classification with deep convolutional neural networks A Krizhevsky, I Sutskever, GE Hinton Advances in neural information processing systems 25	150156 *	2012
Tensorflow: Large-scale machine learning on heterogeneous distributed systems M Abadi, A Agarwal, P Barham, E Brevdo, Z Chen, C Citro, GS Corrado, ... arXiv preprint arXiv:1603.04467	51245 *	2016
Dropout: a simple way to prevent neural networks from overfitting N Srivastava, G Hinton, A Krizhevsky, I Sutskever, R Salakhutdinov The journal of machine learning research 15 (1), 1929-1958	48642	2014
Distributed representations of words and phrases and their compositionality T Mikolov, I Sutskever, K Chen, GS Corrado, J Dean Advances in neural information processing systems 26	42504	2013
Sequence to sequence learning with neural networks I Sutskever, O Vinyals, QV Le Advances in neural information processing systems 27	25046	2014

Cited by VIEW ALL

	All	Since 2019
Citations	472447	366659
h-index	85	81
i10-index	119	115



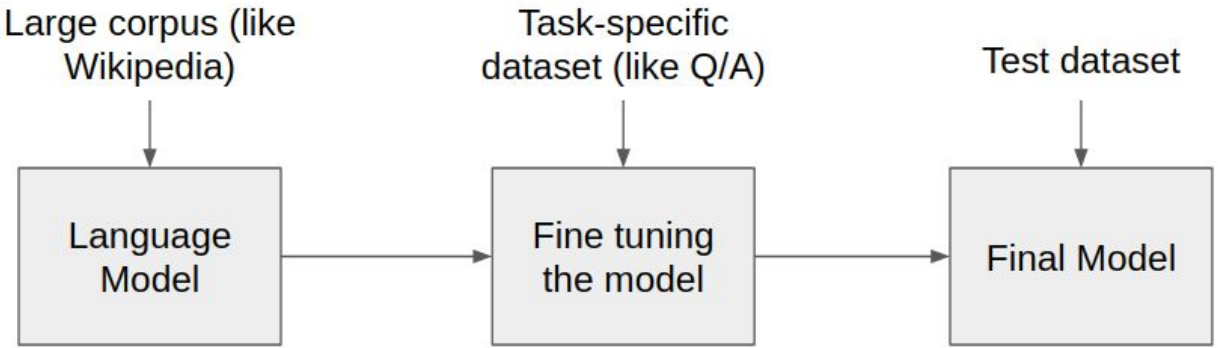
Public access VIEW ALL

0 articles	4 articles
not available	available

Based on funding mandates

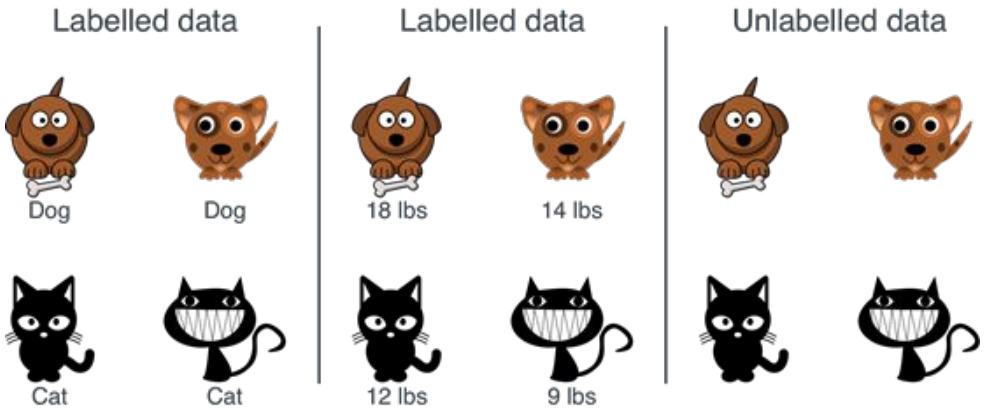
GPT-1: Overview

- Combine **unsupervised pretraining** and **supervised fine-tuning** for language model
- Compared to previous NLP models, no need to change **model architecture** for different tasks
  - Can be easily used by different industry domains
  - Minimize the cost
- SoTA results on 9 out of 12 tasks



Background: Lack of labeled data to train language model

- ImageNet is an image dataset of millions of images with labels
- In contrast, NLP tasks suffer from lack of substantial amounts of labeled data
- How much data/information do we need?
  - “A picture is worth a thousand words.” — Fred R. Barnard
  - 1 million images = 1 billion words = 100 million squences
  - 1 billion words are only enough for training a tiny model, e.g. 10 million parameters
- **How can we use unlabeled text? Is it easy?**



No! It is not easy: Difficulties using unlabeled data

- Pretraining: it was unclear what type of **optimization objectives** are most effective

- Finetuning: there was no effective way to **transfer learned representations**
  - NLP tasks differ from each other a lot in input, output formats.
  - Sentiment analysis is very different from machine translation.
  - Previous methods: changing model architecture (hard to use)

GPT-1 Method: Semi-supervised Training

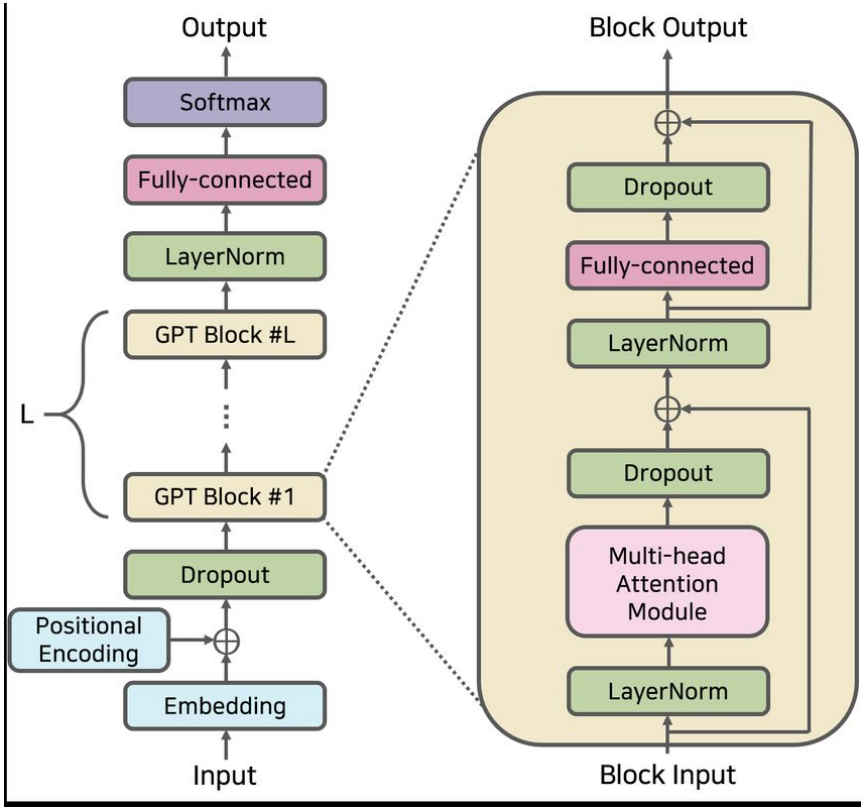
- **Goal:** To learn an **universal representation** that transfers with little adaptation to a wide range of tasks

- A combination of **unsupervised pre-training** and **supervised fine-tuning**
  a. **Now it is just called as self-supervised learning**

- Two-stage training procedure:
  a. Unsupervised pretraining with a language modeling objective
  b. Supervised fine-tuning to a target task

GPT Architecture: Transformer Decoder

- Why **Transformer**, not **RNN**
  - Transformers enable more efficient for handing long-term dependencies in text
  - More robust transfer performance
- Why **Decoder**, not **Encoder**
  - Decoder model using next token prediction training objective, make it more suitable for text generation



Stage1: Unsupervised pre-training

Given an unsupervised corpus of tokens :

$$U = u_1, ..., u_n$$

Use a standard language modeling objective to maximaize the following likelihood:

$$L_1(U) = \sum_i logP(u_i|u_{i-k}, ..., u_{i-1}; \theta)$$

k is the size of the context window, the conditional probability P is modeled using model with parameters θ

Intuition: give me some background information, I predict the next word you will say

Stage2: Supervised fine-tuning

Given a labeled data pair C with input sequence x and a label y,

The objective is to maximize:

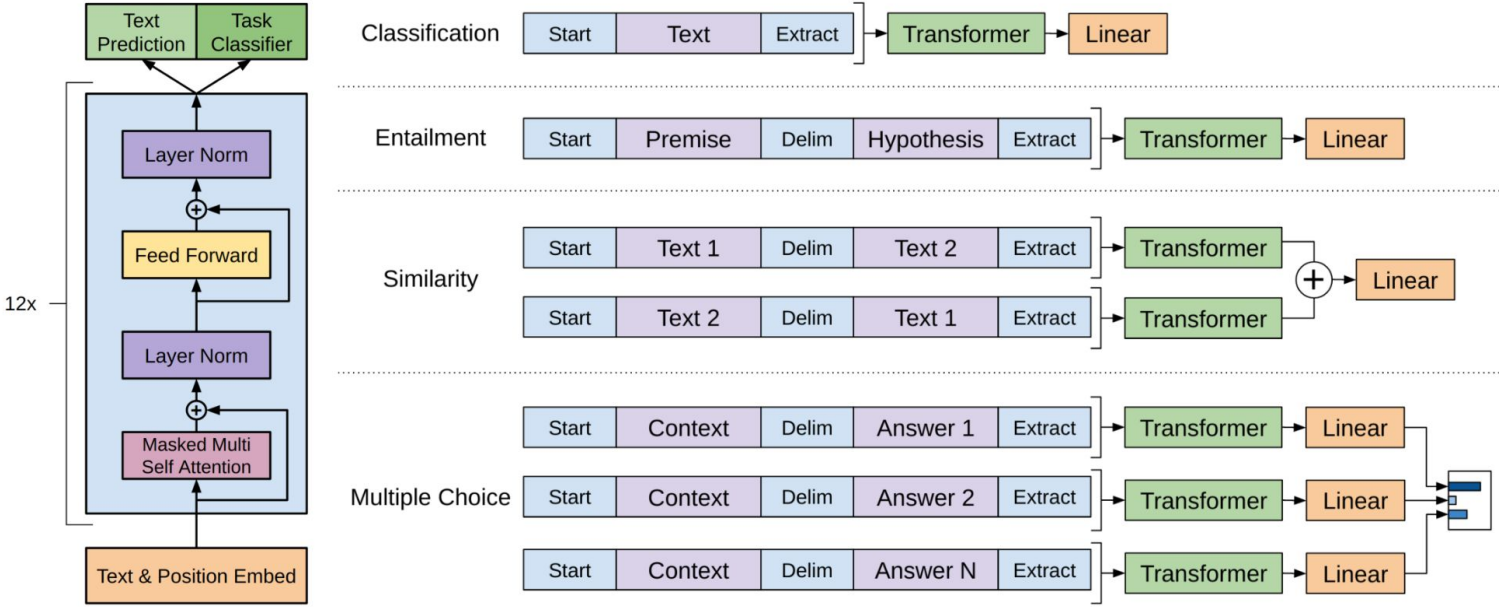
$$L_2(C) = \sum_{(x,y)} logP(y|x^1, ..., x^m)$$

Also, GPT-1 added an auxiliary objective:

$$L_3(C) = L_2(C) + \lambda L_1(C)$$

Stage2: Transfer tasks to target format

Different NLP tasks need to be transferred to **(Input sequnce(s), Label)** format



Experiments: Setup

- **Unsupervised pretraining dataset**:
  - the BooksCorpus dataset, containing over 7000 unique unpublished books

- **Model Specifications:**
  - 12-layer decoder-only transformer,
  - 768 hidden size
  - 12 attention heads.
  - Total parameters 125M, about 1/1500 of GPT-3

Experiments: Results on natural language inference tasks

SoTA performance on 5 out of 6 benchmarks

Table 2: Experimental results on natural language inference tasks, comparing our model with current state-of-the-art methods. 5x indicates an ensemble of 5 models. All datasets use accuracy as the evaluation metric.

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (ours)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

Experiments: Results on QA and reasoning tasks

SoTA results on all four benchmarks

Table 3: Results on question answering and commonsense reasoning, comparing our model with current state-of-the-art methods.. 9x means an ensemble of 9 models.

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	77.6	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	60.2	50.3	53.3
Finetuned Transformer LM (ours)	**86.5**	**62.9**	**57.4**	**59.0**

Experiments: Results on Semantic similarity and classification tasks

SoTA results on 3 out of 5 benchmarks

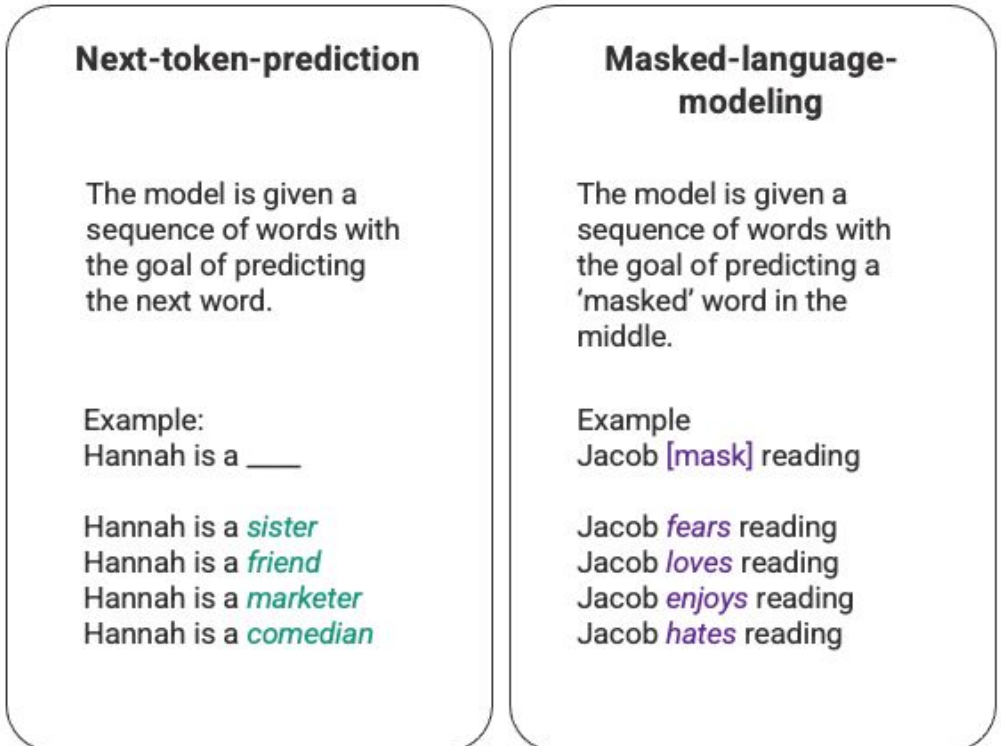
Table 4: Semantic similarity and classification results, comparing our model with current state-of-the-art methods. All task evaluations in this table were done using the GLUE benchmark. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

Method	**Classification**		**Semantic Similarity**			**GLUE**
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	**93.2**	-	-	-	-
TF-KLD [23]	-	-	**86.0**	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3	68.9
Finetuned Transformer LM (ours)	**45.4**	91.3	82.3	**82.0**	**70.3**	**72.8**

Comparison with BERT: Method

- Both combining pre-training and fine-tuning
- GPT uses **next-token prediction**
- BERT uses **masked-language modeling**

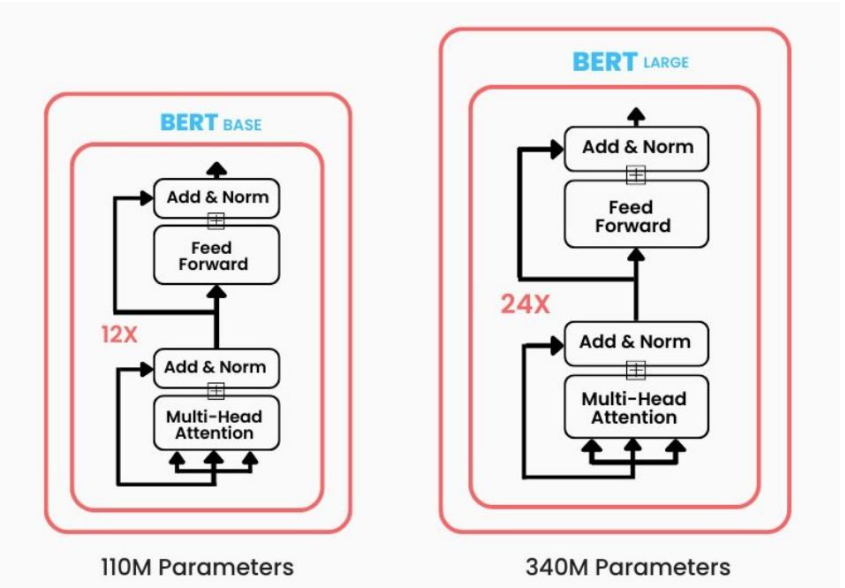
Predicting the future is more challenging than bridging the gap :-)



Comparison with BERT: Architecture

- GPT: Decoder-only Transformer
- BERT: Encoder-only Transformer
- GPT-1: L=12, H=762, A=12
- BERT:
  - BERT BASE: L=12, H=762, A=12, Parameters=110M, comparable with GPT-1
  - BERT LARGE: L=24, H=1024, A=16, Parameters=340M

In 2018, most people believe BERT is better than GPT



Compared with BERT: Empiricial Results and Influence

- BERT achieved better results than GPT-1 and were wider applied by then
- However, GPT-1 started unsupervised pretraining, and GPT series models have been proved more suitable for language generation in the following years
  - GPT-1 has a higher level of innovation than BERT.

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

BERT results on GLUE

GPT-2:
Language Models are
Unsupervised Multitask Learners

Language Models are Unsupervised Multitask Learners

Alec Radford [*1] **Jeffrey Wu** [*1] **Rewon Child** [1] **David Luan** [1] **Dario Amodei** [**1] **Ilya Sutskever** [**1]

Abstract

Natural language processing tasks, such as question answering, machine translation, reading comprehension, and summarization, are typically approached with supervised learning on task-specific datasets. We demonstrate that language models begin to learn these tasks without any explicit supervision when trained on a new dataset of millions of webpages called WebText. When conditioned on a document plus questions, the answers generated by the language model reach 55 F1 on the CoQA dataset - matching or exceeding the performance of 3 out of 4 baseline systems without using the 127,000+ training examples. The capacity of the language model is essential to the success of zero-shot task transfer and increasing it improves performance in a log-linear fashion across tasks. Our largest model, GPT-2, is a 1.5B parameter Transformer that achieves state of the art results on 7 out of 8 tested language modeling datasets in a zero-shot setting but still underfits WebText. Samples from the model reflect these improvements and contain coherent paragraphs of text. These findings suggest a promising path towards building language processing systems which learn to perform tasks from their naturally occurring demonstrations.

1. Introduction

Machine learning systems now excel (in expectation) at tasks they are trained for by using a combination of large datasets, high-capacity models, and supervised learning (Krizhevsky et al., 2012) (Sutskever et al., 2014) (Amodei et al., 2016). Yet these systems are brittle and sensitive to slight changes in the data distribution (Recht et al., 2018) and task specification (Kirkpatrick et al., 2017). Current systems are better characterized as narrow experts rather than competent generalists. We would like to move towards more general systems which can perform many tasks – eventually without the need to manually create and label a training dataset for each one.

The dominant approach to creating ML systems is to collect a dataset of training examples demonstrating correct behavior for a desired task, train a system to imitate these behaviors, and then test its performance on independent and identically distributed (IID) held-out examples. This has served well to make progress on narrow experts. But the often erratic behavior of captioning models (Lake et al., 2017), reading comprehension systems (Jia & Liang, 2017), and image classifiers (Alcorn et al., 2018) on the diversity and variety of possible inputs highlights some of the shortcomings of this approach.

Our suspicion is that the prevalence of single task training on single domain datasets is a major contributor to the lack of generalization observed in current systems. Progress towards robust systems with current architectures is likely to require training and measuring performance on a wide range of domains and tasks. Recently, several benchmarks have been proposed such as GLUE (Wang et al., 2018) and decaNLP (McCann et al., 2018) to begin studying this.

Multitask learning (Caruana, 1997) is a promising framework for improving general performance. However, multitask training in NLP is still nascent. Recent work reports modest performance improvements (Yogatama et al., 2019) and the two most ambitious efforts to date have trained on a total of 10 and 17 (`dataset, objective`) pairs respectively (McCann et al., 2018) (Bowman et al., 2018). From a meta-learning perspective, each (`dataset, objective`) pair is a single training example sampled from the distribution of datasets and objectives. Current ML systems need hundreds to thousands of examples to induce functions which generalize well. This suggests that multitask training many need just as many effective training pairs to realize its promise with current approaches. It will be very difficult to continue to scale the creation of datasets and the design of objectives to the degree that may be required to brute force our way there with current techniques. This motivates exploring additional setups for performing multitask learning.

The current best performing systems on language tasks

[*,**]Equal contribution [1]OpenAI, San Francisco, California, United States. Correspondence to: Alec Radford <alec@openai.com>.

GPT-2: Overview

- Larger Pretraining Dataset
  - WebText, millions of webpages
- Larger Model Size
  - Largest model by then, GPT-2, has 1.5B parameters (10x GPT-1, 5x BERT large)
- Zero-shot setting!
  - No more fine-tuning
  - Achieved promising, competitive results

GPT-2: why do we need a new GPT?

- BERT Outperfomrs GPT-1
  - GPT needs to be improved
  - How? just a larger dataset and a larger model
    - scaling is all you need :-)

- A specific training dataset for a specific task remained the dominant approach
  - This kind of system suffered from **lack of generalization**
  - For some tasks, there are no specific labeled data for fine-tuning

Difference

GPT-1

- Unsupervised Pretraining
  - Objective function: next-token prediction
- Task-specific fine-tuning
  - Transfer task format to one input sequnce with the **delimiter**
  - Model learn the task input format by task-specified training data

GPT-2

- Unsupervised Pretraining
  - Same method with larger datasets
- Zero-shot learning
  - No more fine-tuning
  - Specify the task with **natural language** and take it as input

GPT-2 Approach: Zero-shot

Examples:

- Machine Translation
  - Input Sequence: **(translate to French, english text) -> french text**
- Reading comprehension
  - Input Sequence:**(answer the question, document, question) -> answer**

This type of input is known as **prompt: term used to asking the model**

GPT-2 Approach: Zero-shot

You

translate to Chinese: GPT-1 has a higher level of innovation than BERT.

ChatGPT

GPT-1比BERT具有更高的创新性。

You

answer the question: Room A has 10 murderers, Tom gets into Room A and kills 1 murderer. How many murderers now in this room?

ChatGPT

After Tom enters Room A and kills 1 murderer, there are still 10 murderers in the room, including Tom, assuming his act of killing the murderer makes him a murderer as well.

Zero-shot: Why it works

- Powerful language models shall be able to learn the tasks expressed in natural language since it **learn directly from natural language** in the pretraining stage
- Demonstrations of tasks expressed in natural language are probably in the pretraining corpus

Example of En-Fr translation demonstrations in WebText training set

”I’m not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I’m not a fool].**
In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: ”**Mentez mentez, il en restera toujours quelque chose**,” which translates as, ”**Lie lie and something will always remain.**”
“I hate the word ‘**perfume**,”’ Burr says. ‘It’s somewhat better in French: ‘**parfum**.’
If listened carefully at 29:55, a conversation can be heard between two guys in French: “**-Comment on fait pour aller de l’autre coté? -Quel autre coté?**”, which means “**- How do you get to the other side? - What side?**”.
If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?
“**Brevet Sans Garantie Du Gouvernement**”, translated to English: “**Patented without government warranty**”.

Table 1. Examples of naturally occurring demonstrations of English to French and French to English translation found throughout the WebText training set.

Experiments: Setup

- **Unsupervised pretraining dataset**:
  - From reddit, resulting in WebText
  - 45 million links, 8 million documents, 40GB text

- **Model Specifications:**

Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

GPT-2: Empirical Results

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

Table 3. Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).

GPT-2 models outperform other zero-shot systems on 8 datasets

GPT-2: Empirical Results

- Still fall behind SoTA methods on reading comprehension, translation, summarization, QA tasks
- Show **promising results with model scaling**

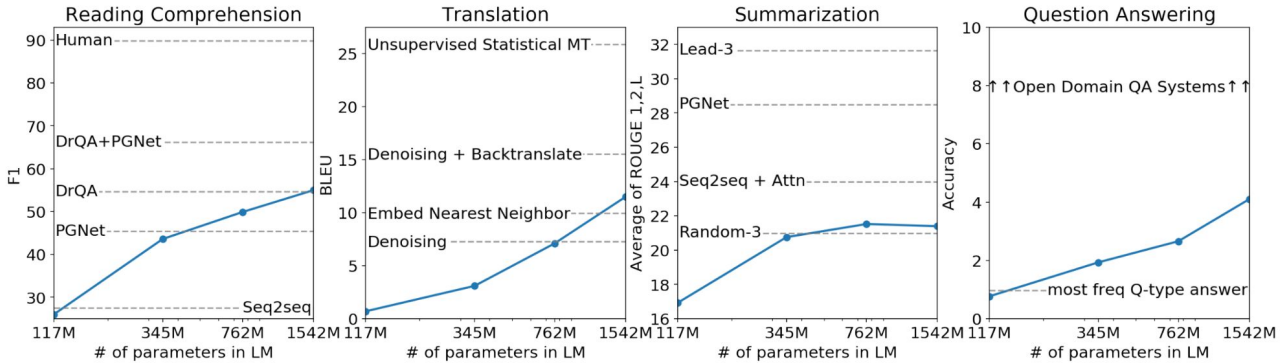


Figure 1. Zero-shot task performance of WebText LMs as a function of model size on many NLP tasks. Reading Comprehension results are on CoQA (Reddy et al., 2018), translation on WMT-14 Fr-En (Artetxe et al., 2017), summarization on CNN and Daily Mail (See et al., 2017), and Question Answering on Natural Questions (Kwiatkowski et al., 2019). Section 3 contains detailed descriptions of each result.

GPT-3:
Language Models are Few-Shot Learners

Language Models are Few-Shot Learners

Tom B. Brown* **Benjamin Mann*** **Nick Ryder*** **Melanie Subbiah***

Jared Kaplan† **Prafulla Dhariwal** **Arvind Neelakantan** **Pranav Shyam** **Girish Sastry**

Amanda Askell **Sandhini Agarwal** **Ariel Herbert-Voss** **Gretchen Krueger** **Tom Henighan**

Rewon Child **Aditya Ramesh** **Daniel M. Ziegler** **Jeffrey Wu** **Clemens Winter**

Christopher Hesse **Mark Chen** **Eric Sigler** **Mateusz Litwin** **Scott Gray**

Benjamin Chess **Jack Clark** **Christopher Berner**

Sam McCandlish **Alec Radford** **Ilya Sutskever** **Dario Amodei**

OpenAI

Abstract

Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, humans can generally perform a new language task from only a few examples or from simple instructions – something which current NLP systems still largely struggle to do. Here we show that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching competitiveness with prior state-of-the-art fine-tuning approaches. Specifically, we train GPT-3, an autoregressive language model with 175 billion parameters, 10x more than any previous non-sparse language model, and test its performance in the few-shot setting. For all tasks, GPT-3 is applied without any gradient updates or fine-tuning, with tasks and few-shot demonstrations specified purely via text interaction with the model. GPT-3 achieves strong performance on many NLP datasets, including translation, question-answering, and cloze tasks, as well as several tasks that require on-the-fly reasoning or domain adaptation, such as unscrambling words, using a novel word in a sentence, or performing 3-digit arithmetic. At the same time, we also identify some datasets where GPT-3's few-shot learning still struggles, as well as some datasets where GPT-3 faces methodological issues related to training on large web corpora. Finally, we find that GPT-3 can generate samples of news articles which human evaluators have difficulty distinguishing from articles written by humans. We discuss broader societal impacts of this finding and of GPT-3 in general.

*Equal contribution
†Johns Hopkins University, OpenAI

Author contributions listed at end of paper.

GPT-3: Overview

- Largest AI model by then
  - 175B parameters, 1500x GPT-1, 100x GPT-2

- Strong performance on many NLP tasks
  - With **few-shot learning,** still no need for fine-tuning

- Strong generative ability
  - Can **generate human-like articles** which can hardly be regconized

Background: Limits of pretraining - finetuning paradigm

- Need for a large labeled dataset for every new task limits the **applicability of language mode**

- The Large-scale pre-trained models that are fine-tuned on a narrower data distribution are likely to face the problem of data leakage.
  - That is, the good performance of the model does not stem from the model itself, but because similar examples have appeared in the training data.

- Humans do not require large supervised dataset to learn new language tasks

Method: Few-shot (in context learning)

Zero-shot

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

```
1  Translate English to French:        ← task description
2  cheese =>                           ← prompt
```

One-shot

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

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← example
3  cheese =>                           ← prompt
```

Few-shot

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

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← examples
3  peppermint => menthe poivrée        ←
4  plush girafe => girafe peluche      ←
5  cheese =>                           ← prompt
```

Why few-shot is unique?

- We can NOT guarantee zero-shot and one-shot are real
  - They may remember a similar pattern in the traning data

- Few-shot
  - We assume no similar pattern is learned in the training data

Summary

- Few-shot learning aligns more closely with the concept of in-context learning

- It is the ability of LLMs to learn to solve tasks by learning from examples in the input

GPT-3 Architecture

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}

- Same architectures with GPT-2
- Alternating dense and locally banded sparse attention similar to Sparse Transformer
- 8 different sizes of model, ranging from 125M to 175B

GPT-3: Training Dataset (300b tokens)

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

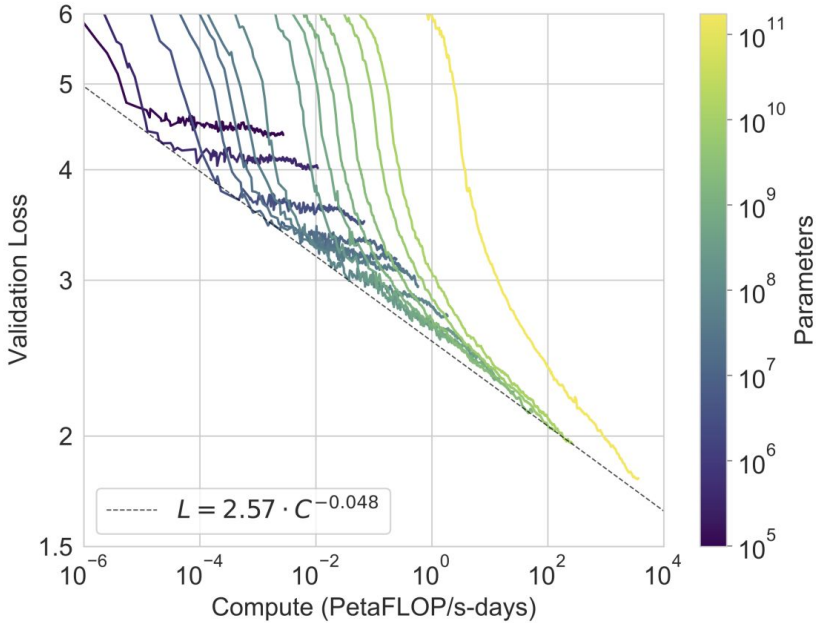
3 steps to improve data quality

1. **Filter** CommonCrawl based on similarity to high-quality data
2. **Fuzzy deduplication** at document level by using lsh algorithm
3. **Add and mix** known high-quality reference data

GPT-3: Evaluation and Results

Power Law:

Larger size models gain better performance with more compute

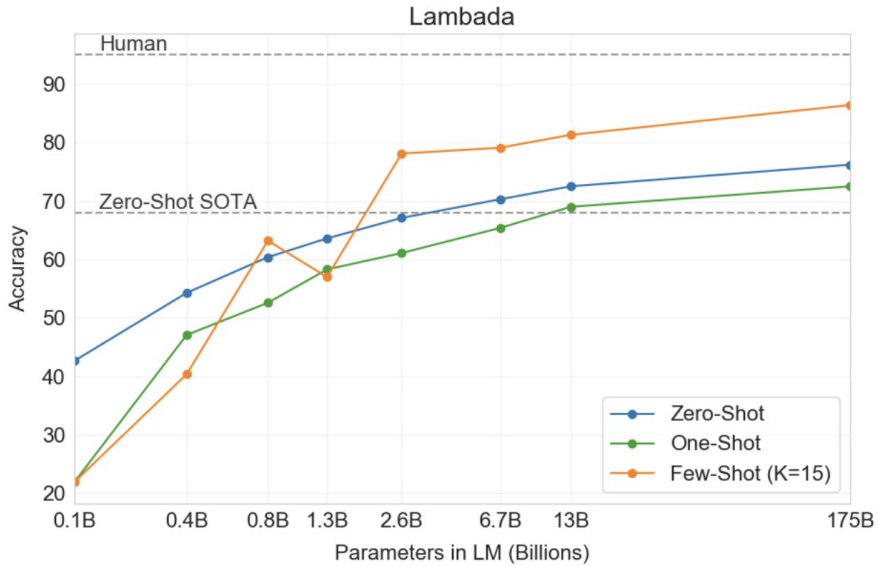


1 petaFLOPS (PFLOPS) = 1 quadrillion (10^15) floating-point operations per second
1 PFLOPS = 1,000 TFLOPS = 10^6 GFLOPS

GPT-3: Evaluation and Results

Zero-Shot SOTA:

- On LAMBADA benchmark,
- 2.7B outperforms the SOTA 17B parameter Turing-NLG
- GPT-3 175B advances the previous SoTA by 18%.



GPT-3: Evaluation and Results

On TriviaQA benchmark,

GPT-3 with few-shot, one-shot can achieve competitive results with Fine-tuned SoTA

GPT-3: Limitations

- Still has notable weakness in document-level text generation

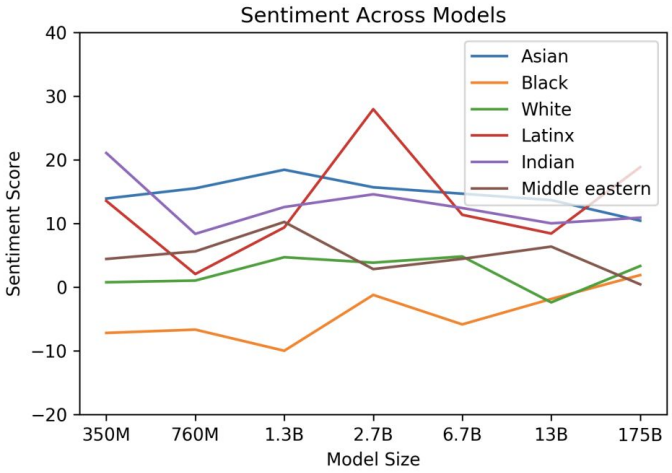
- GPT pretraining objective weights every token **equally** and lacks a notion of what is more important to predict

- It remains uncertain whether GPT-3 few-shot learning actually learns new task from **scrath** or if it simply recognizes and identifies tasks that has learned in the training data

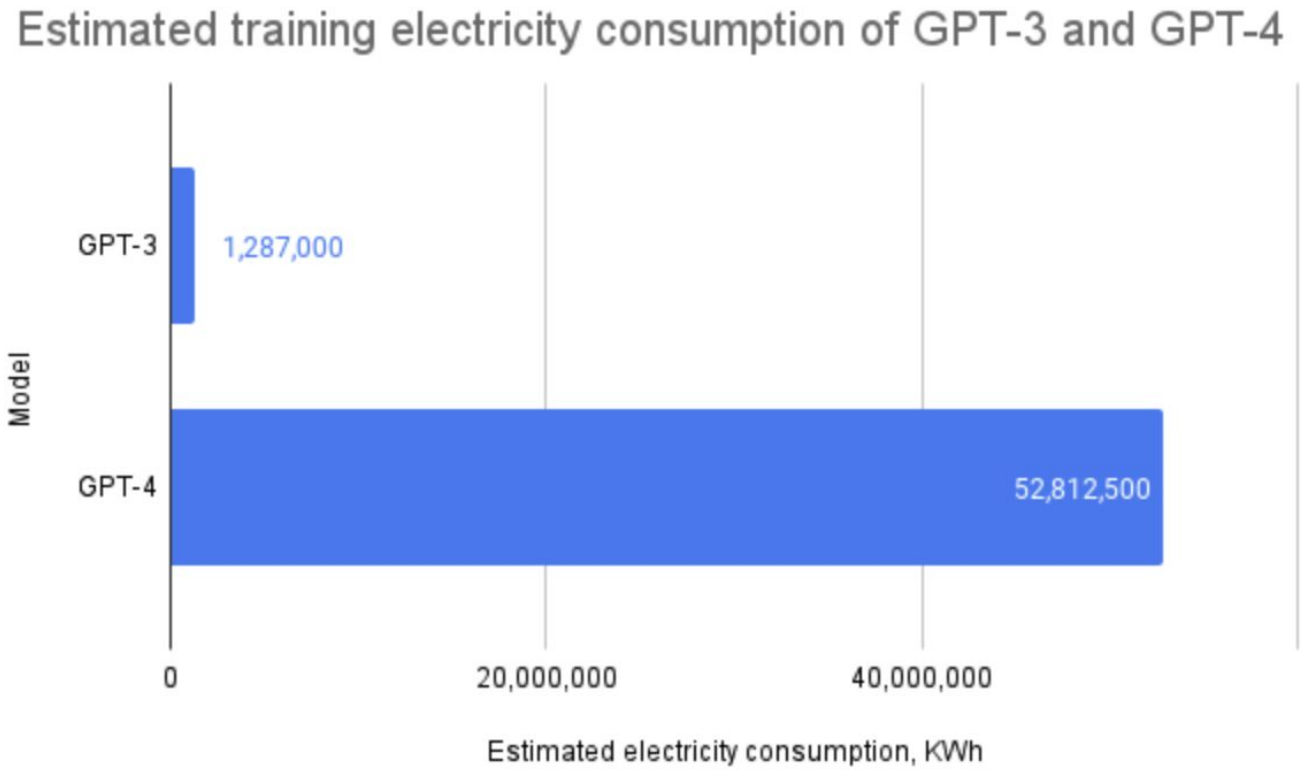
GPT-3: Broader Impacts

- Safety
  - can generate misinformation, spam, phishing,etc
- Bias
  - may be bias in gender, race, religion
- Energy Usage
  - Consume huge amounts of computation, which is energy-internsive

GPT-3 has different sentiments on different races



Cost



price per kWh in USA: 0.17 dollars (2024)

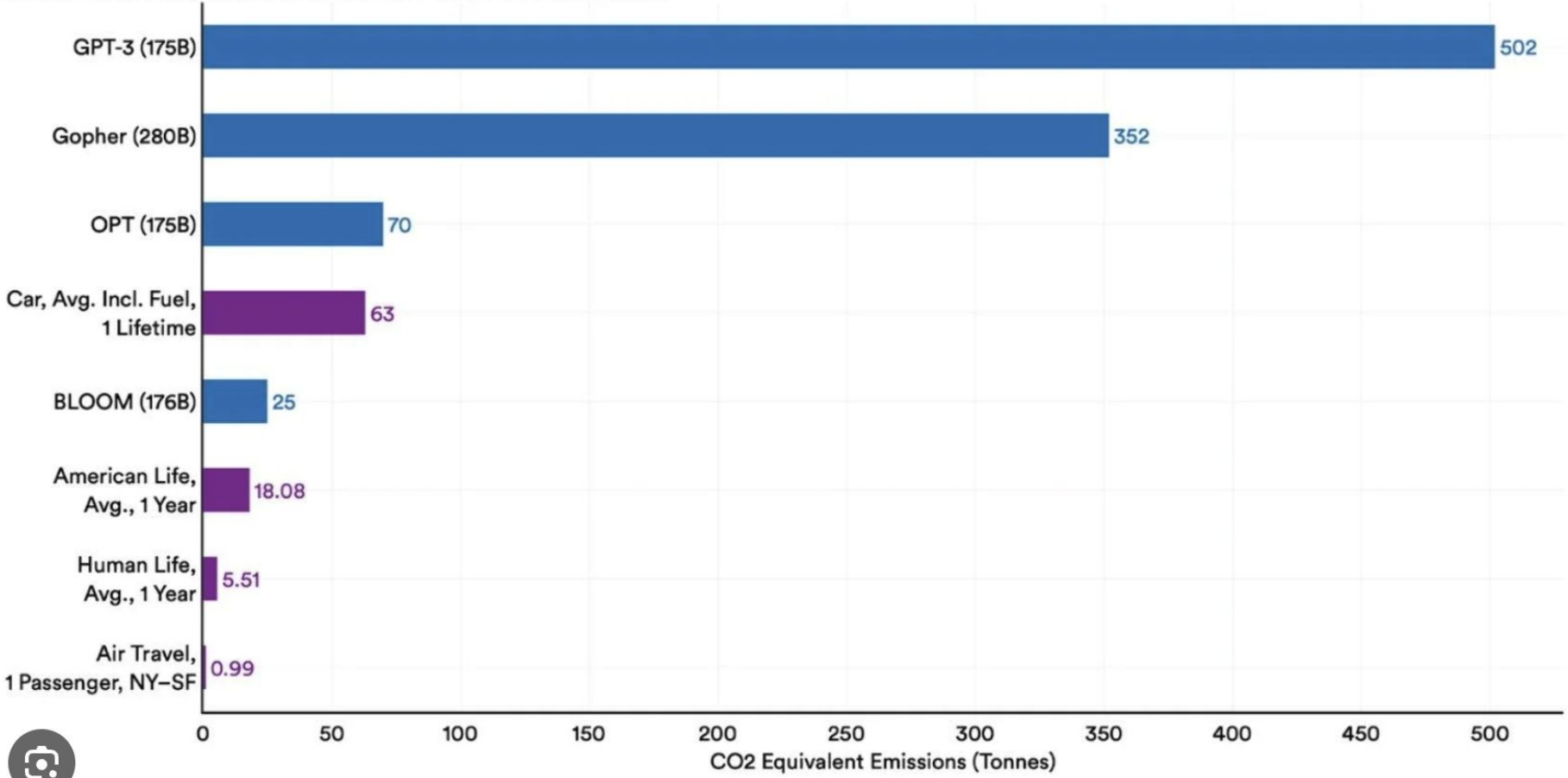
Important: a single GPT query consumes 15x more energy than a Google search query

price per kWh in USA: 0.17 dollars (2024)

Type of Service	Estimated Energy Consumption (per query or operation)
Google Search Query	0.0003 kWh (1.08 kJ)
NLP/ChatGPT-4 Query	0.001-0.01 kWh (3.6-36 kJ) *
SQL Database Query	0.0001-0.001 kWh (0.36-3.6 kJ) **
Graph Database Query	0.0001-0.01 kWh (0.36-36 kJ) ***
Cloud Container	0.001-0.1 kWh (3.6-360 kJ) ****
Serverless Function	0.00001-0.001 kWh (0.036-3.6 kJ) *****

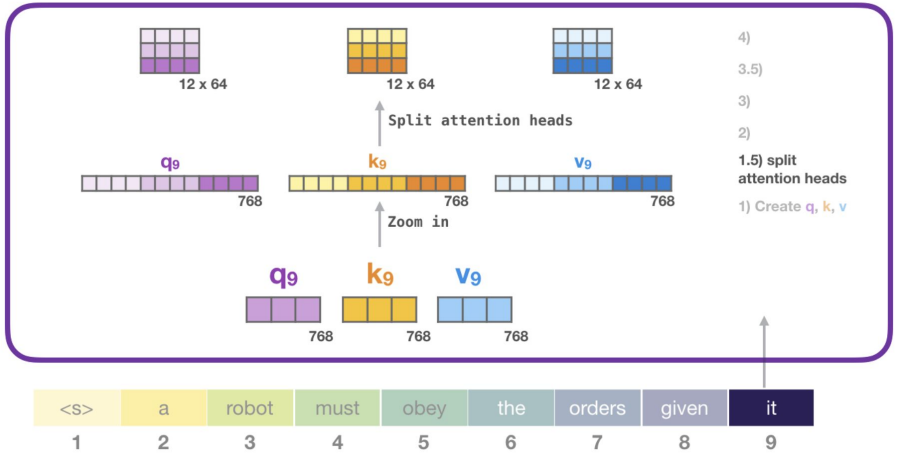
CO2 Equivalent Emissions (Tonnes) by Selected Machine Learning Models and Real Life Examples, 2022

Source: Luccioni et al., 2022; Strubell et al., 2019 | Chart: 2023 AI Index Report



3-minute Quiz: computing the number of parameters?

	GPT-1	GPT-2	GPT-3
Dimention	768	1600	12288
FF Dimention	768*4	1600*4	12288*4
Layers	12	48	96
Parameters	?	?	?



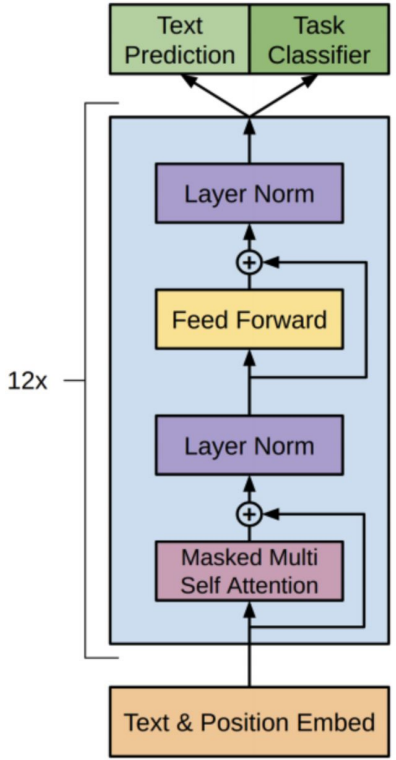
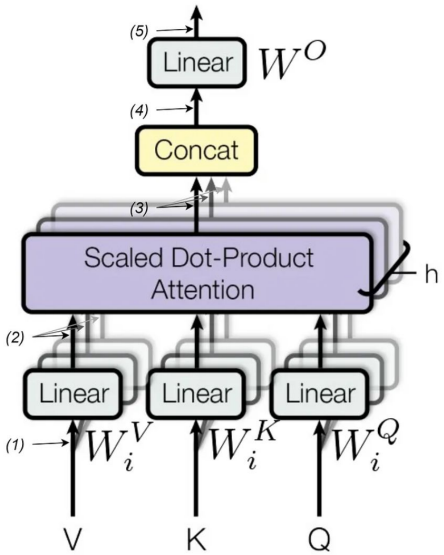


Figure credit https://jalammar.github.io/illustrated-gpt2/

Compute # Parameters of GPT model: Attention Layer



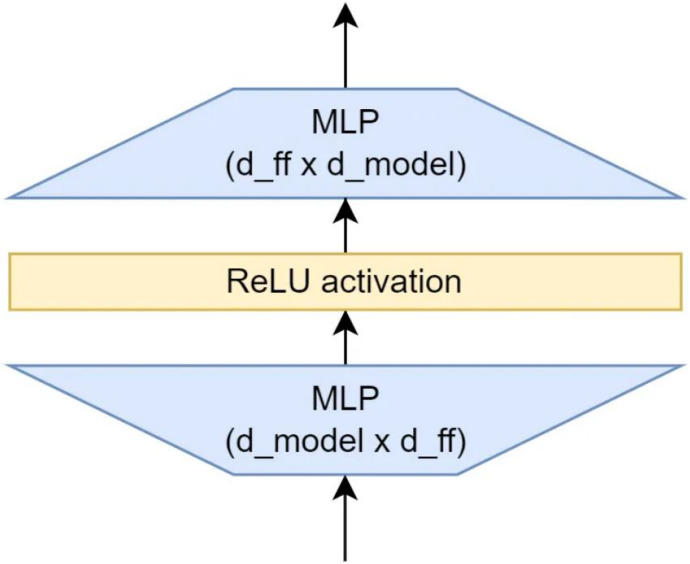
$$W^O.shape = (d_{model}, d_{model})$$
$$W_i^Q.shape = (d_{model}, d_{qkv})$$
$$W_i^K.shape = (d_{model}, d_{qkv})$$
$$W_i^V.shape = (d_{model}, d_{qkv})$$
$$d_{qkv} = d_k = d_v = d_{model}/n_{heads} = d_{model}/h$$

$$N_{attention} = \overbrace{(d_{model} * d_{model} + d_{model})}^{W^O} + \overbrace{(d_{model} * d_{qkv} + d_{qkv})}^{W_i^Q, W_i^K, or\ W_i^V} * n_{heads} * 3 =$$
(Three matrices for every head)

$$= (d_{model} * d_{model} + d_{model}) + (d_{model} * d_{model} + d_{model}) * 3 =$$
$$(d_{model} * d_{model} + d_{model}) * 4 = \underbrace{(d_{model}^2 + d_{model}) * 4}_{\text{The exact formula}} \approx \underbrace{4 * d_{model}^2}_{\text{The approximate formula}}$$

$$Why\ (d_{model} * d_{qkv} + d_{qkv}) * n_{heads} = (d_{model} * d_{model} + d_{model}):$$
$$(d_{model} * d_{qkv} + d_{qkv}) * n_{heads} =$$
$$d_{model} * \underbrace{d_{qkv} * n_{heads}}_{d_{model},\ since\ d_{qvk} = d_{model}/n_{heads}} + \underbrace{d_{qkv} * n_{heads}}_{d_{model}} = d_{model} * d_{model} + d_{model}$$

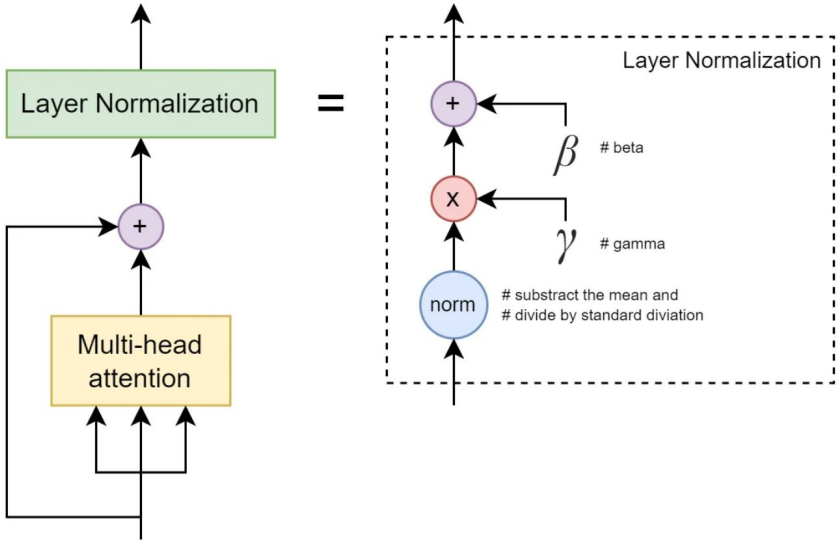
Compute # Parameters of GPT model: FFN



d_ff: FFN dimension
output dim of first linear layer,
input dim of second linear layer,
d_ff=4*d_model is a common practice

$$N_{feedforward} = \overbrace{(d_{model} * d_{ff} + d_{ff})}^{\text{The first linear layer}} + \overbrace{(d_{ff} * d_{model} + d_{model})}^{\text{The second linear layer}} =$$
$$d_{model} * d_{ff} + d_{ff} * d_{model} + d_{model} + d_{ff} =$$
$$\underbrace{2 * d_{model} * d_{ff} + d_{model} + d_{ff}}_{\text{The exact formula}} \approx \underbrace{2 * d_{model} * d_{ff}}_{\text{The approximate formula}}$$

Compute # Parameters of GPT model: LayerNorm



$$N_{layernorm} = d_{model} * 2$$

Element-wise operation

LayerNorm's parameter count is much lower compared to MHA and FFN, and can be omitted when estimating the parameter count of a transformer.

GPT 1,2,3 Model Size:

	GPT-1	GPT-2	GPT-3
d_model	768	1600	12288
N_attention≈ 4*d_model^2	4*768^2= 2,359,296	4*1600^2= 10,240,000	4*12288^2= 603,979,776
N_ffn≈ 8*d_model^2	8*768^2= 4,718,592	8*1600^2= 20,480,000	8*12288^2= 1,207,959,552
N_block≈ N_attention+ N_ffn	12*768^2= 7,077,888	12*1600^2= 30,720,000	12*12288^2= 1,811,939,328
Len_model	12	48	96
Parameters≈ N_block*Len_model	12*7,077,888≈ 85M	48*30,720,000≈ 1.47B	96*12*12288^2≈ 173.95B

Here we make an approximate count, without counting:

1. bias in linear layer
2. layernorm
3. embedding layers

Run the code (very important!)

https://github.com/karpathy/minGPT/tree/master/mingpt

https://github.com/yuxuan-lou/minGPT/tree/master

https://colab.research.google.com/drive/1asudePTx_9n91_TgxZyFwjA4sA4A599-?usp=sharing

GPT-3: Resources

GPT-3 DEMO, 800+ apps based on GPT-3
https://gpt3demo.com