A Brief Overview of Large Models

Yuhao Cheng

https://yuhaocheng.github.io/

May 15, 2023

Outline

1. Introduction
2. Development
3. Basic Knowledge
4. Advanced Knowledge
5. Basic Large Models
6. More Models and Application

Contents

1. Introduction
2. Development
3. Basic Knowledge
4. Advanced Knowledge
5. Basic Large Models
6. More Models and Application

Introduction

What's the Large Model?

- The number of model's parameters is large. Nevertheless, the definition of *"large"* changes with the development of the hardware.

Why the large models attract much more attention than before?

- The intuitive performance of recent models is better than before.
- Think about the AlexNet, which firstly gets a great quantitative performance on ImageNet.

Contents

1 Introduction

2 Development

3 Basic Knowledge

4 Advanced Knowledge

5 Basic Large Models

6 More Models and Application

Development

Development of the LLM

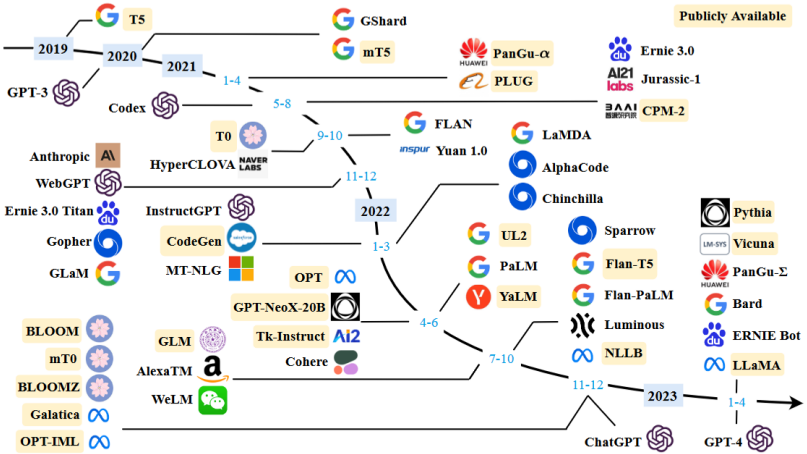


Figure 1: The overview of recent LLM[31].

Development

Model Size

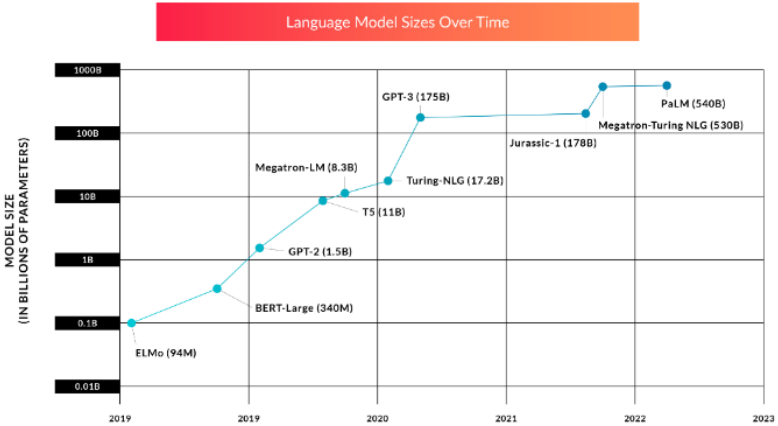


Figure 2: The increase of the model size[11].

Development

Arena

Rank	Model	Elo Rating	Description
1	vicuna-13b	1169	a chat assistant fine-tuned from LLaMA on user-shared conversations by LMSYS
2	koala-13b	1082	a dialogue model for academic research by BAIR
3	oasst-pythia-12b	1065	an Open Assistant for everyone by LAION
4	alpaca-13b	1008	a model fine-tuned from LLaMA on instruction-following demonstrations by Stanford
5	chatglm-6b	985	an open bilingual dialogue language model by Tsinghua University
6	fastchat-t5-3b	951	a chat assistant fine-tuned from FLAN-T5 by LMSYS
7	dolly-v2-12b	944	an instruction-tuned open large language model by Databricks
8	llama-13b	932	open and efficient foundation language models by Meta
9	stablelm-tuned-alpha-7b	858	Stability AI language models

Figure 3: The Elo table of current LLM[2].

Contents

1 Introduction

2 Development

3 Basic Knowledge

4 Advanced Knowledge

5 Basic Large Models

6 More Models and Application

Basic Knowledge

Overview

Before we go through the advanced knowledge used in recent large model, we will firstly see some basic things which makes the Neural Network.

- Conventional Neural Network. The basic component of the neural network.
- Attention Mechanism. The basic theory of transformers.
- Diffusion Process. The core idea of the diffusion networks.

Conventional Neural Network

Overview

The CNN is first large model at that ear. The AlexNet[13] is split into two GPUs to train and inference.

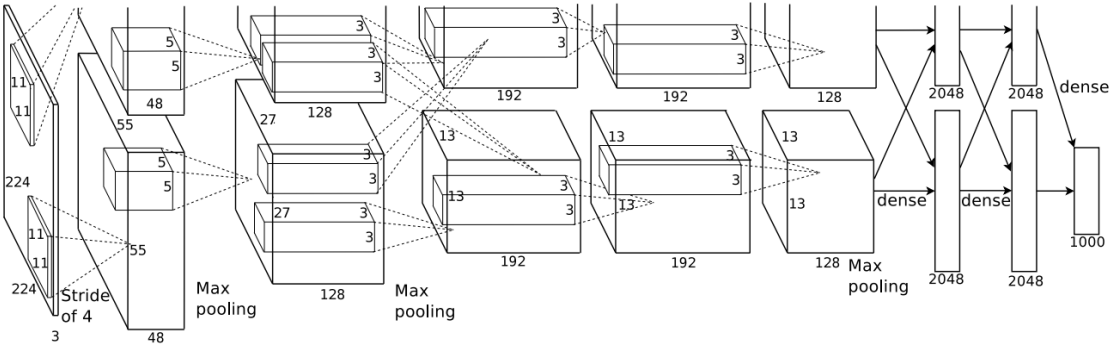


Figure 4: The architecture of the AlexNet.

Conventional Neural Network

Details

The CNN is the network made up by the conventional kernel. Each kernel is a square of learnable weights and different kernel consists the different channels. One channel's weights are sharable on the whole image.

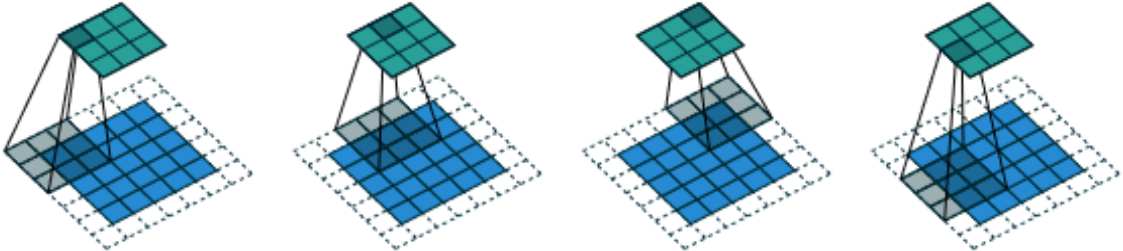


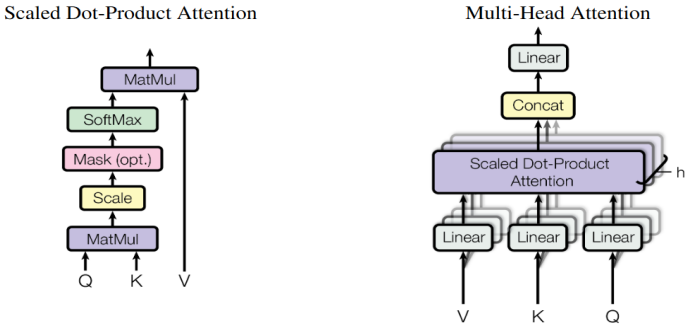
Figure 5: The schematic figure of conventional process.

Attention Mechanism

Attention computation

The bloom of the attention, or in the other words, the transformer architecture, is from [27].

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (1)$$



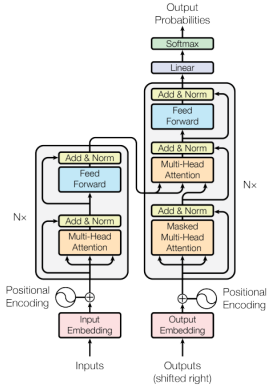


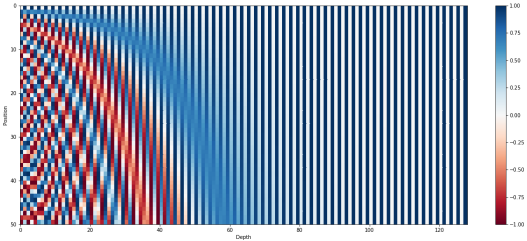
Figure 6: The architecture of the transformer

Attention Mechanism

Positional Encoding

Only having the self-attention will make the model not know the sequential information of each position. So in order to solve this problem, researchers use the **Positional Embedding** to make the model get the sequential information.

$$
\begin{aligned}
PE_{(pos,2i)} &= \sin\left(pos/10000^{2i/d_{\text{model}}}\right) \\
PE_{(pos,2i+1)} &= \cos\left(pos/10000^{2i/d_{\text{model}}}\right)
\end{aligned}
\tag{2}
$$



Diffusion Process

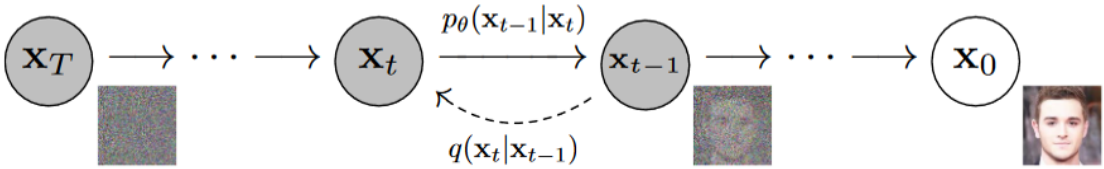


Figure 7: The diffusion process

$$p_\theta\left(\mathbf{x}_{0:T}\right) := p\left(\mathbf{x}_T\right) \prod_{t=1}^{T} p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right), \quad p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) := \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right)\right) \tag{3}$$

$$q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right) := \prod_{t=1}^{T} q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right), \quad q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) := \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right) \tag{4}$$

Diffusion Process

Reverse Phase

The reverse phase is **recover the image** from the noise image.

$$p_\theta\left(\mathbf{x}_{0:T}\right) := p\left(\mathbf{x}_T\right) \prod_{t=1}^{T} p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right), \quad p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) := \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right)\right)$$

$$p_\theta\left(\mathbf{x}_0\right) := \int p_\theta\left(\mathbf{x}_{0:T}\right) d\mathbf{x}_{1:T}$$

$$p\left(\mathbf{x}_T\right) = \mathcal{N}\left(\mathbf{x}_T; \mathbf{0}, \mathbf{I}\right)$$

Diffusion Process

Diffusion Phase

The diffusion phase is making the image into a noise image by adding the noise step by step. It gradually adds Gaussian noise to the data according to a variance schedule $\beta_1, ..., \beta_T$:

$$q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right) := \prod_{t=1}^{T} q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right), \quad q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) := \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right)$$

The training loss of this method is:

$$L = \mathbb{E}_q[\underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_T \mid \mathbf{x}_0\right) \| p\left(\mathbf{x}_T\right)\right)}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)\right)}_{L_{t-1}} \underbrace{- \log p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)}_{L_0}]$$

Contents

1 Introduction

2 Development

3 Basic Knowledge

4 Advanced Knowledge

5 Basic Large Models

6 More Models and Application

Advanced Knowledge

Overview

This part will introduce some new techs used in super large model

- Generative Pre-trained Transformer(GPT)
- Prompt & Instruct Learning
- Reinforcement learning from Human Feedback(RLHF)
- Low Rank Adaptation(LoRA)

Generative Pre-trained Transformer(GPT)

Generative Pre-trained Transformer(GPT)[21] is proposed by OpenAI in 2018. It proposes using the self-supervised learning to train the large Language model.

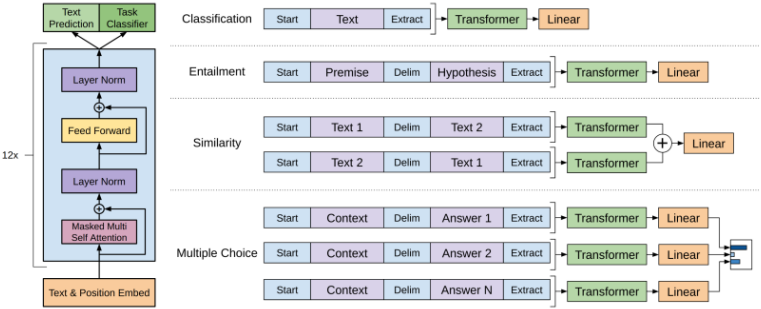


Figure 8: (left)Transformer architecture and training objectives used in this work. (right) Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Prompt & Instruct Learning

Overview

Prompt means providing models some clues to help it understand users better. However, they are not super-new staff. It firstly comes at 2021

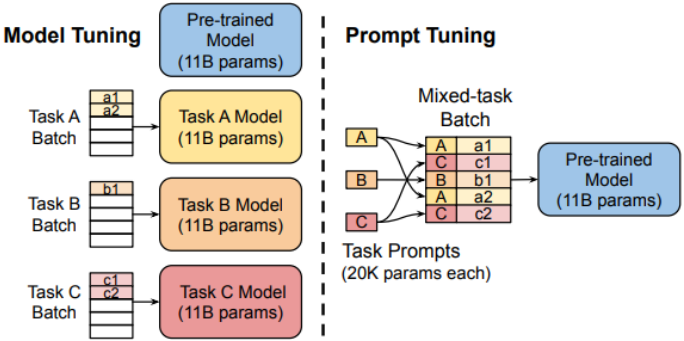


Figure 9: The difference between the traditional fine-tuning and prompt[14].

Prompt Learning

Prompt learning is proposed in [25]. It use the prompt to transfer different task to the Masked-style, which can be used to fine-tune the BERT.(1) A number of patterns encoding some form of task description are created to convert training examples to close questions; for each pattern, a pretrained language model is finetuned. (2) The ensemble of trained models annotates unlabeled data. (3) A classifier is trained on the resulting soft-labeled dataset.

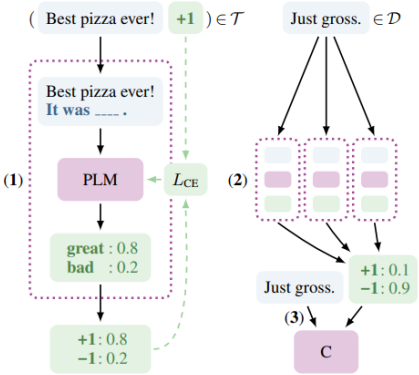


Figure 10: PET for sentiment classification.

Instruct Learning

After the prompt learning, researchers propose instruct learning[28] to improve the generality of the NLP models.

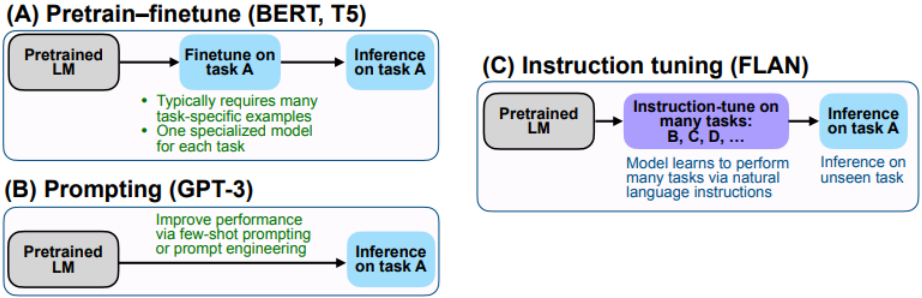


Figure 11: Comparing instruction tuning with pretrain-finetune and prompting.

Prompt Learning and Instruct Leaning Example

The original sentence: **I love this movie.** Prompt Learning:

- Predict the masked part in **I love this movie. The movie is _.**
- Labels may be *great, excellent*.

Instruct Learning:

- Judge the label of the this sentence: **I love this movie. The movie is great.**
- Labels may be **A is Good**, **B is Normal** and **C is Bad**.

In a word, the Prompt Learning is ability of Completion, while the Instruct Learning is understanding.

RLHF
Overview

The Reinforcement learning from Human Feedback(RLHF)[19] is one of the key technology in recent large model fine-tuning. And as the result of using this technology, the ChatGPT achieves the great performance. In general, the RLHF can be divided into three steps:

1. Pre-training Models;
2. Getting the reward model;
3. Using reinforcement learning to fine-tune models.

RLHF

Overview

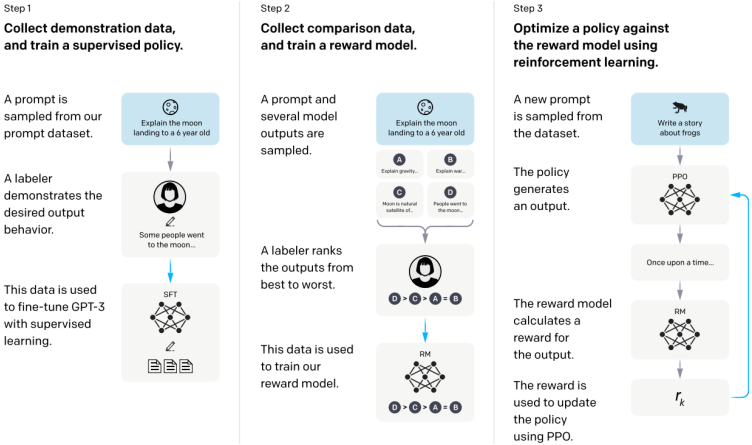


Figure 12: The RLHF process introduced in InstructGPT[19]

RLHF

Pre-training Models

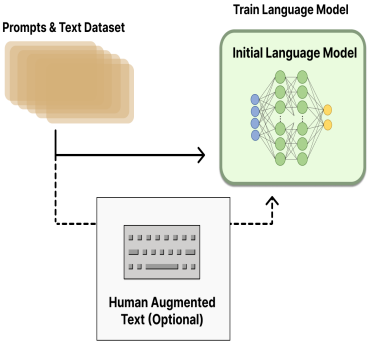


Figure 13: RLHF-Step1: Get pre-training models

We need have some pre-trained Large Models. For this part, we may need use others' or use huge number of GPUs to train some models.

RLHF

Getting the reward model

We need to use some labor to rank the LM results and use the rank as the input of the reward model to get the reward score for the next step.

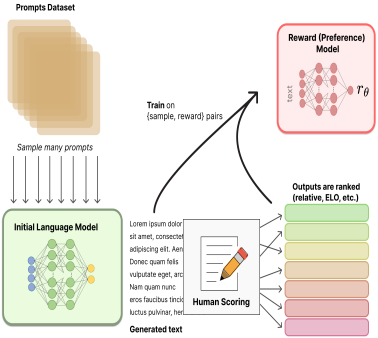


Figure 14: RLHF-Step2: Get the reward model

RLHF

Using reinforcement learning to fine-tune models

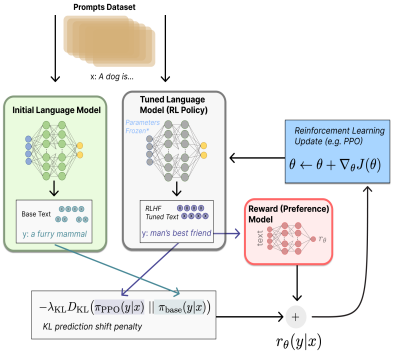


Figure 15: RLHF-Step3: Use the reinforcement learning to fine-tune

The last step will define the Policy, Action of updating the LM, and the reward is from the Step2.

Low Rank Adaptation(LoRA)

Overview

LoRA[9] is a useful method to fine-tune the large model without using huge computation power. Because **full fine-tuning** of all the parameters of the huge model is not always possible. And when inference, we can directly add these trained parameters into the pre-trained weights.
For example, d equals to 1288 while the r equals to 1 or 2.

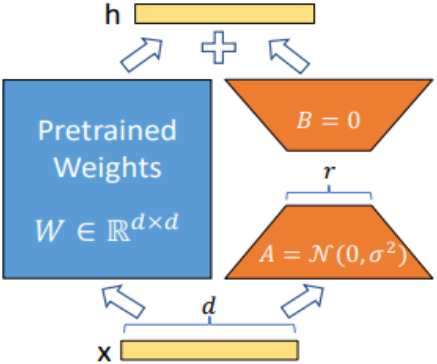


Figure 16: The LoRA method.

Contents

1 Introduction

2 Development

3 Basic Knowledge

4 Advanced Knowledge

5 Basic Large Models

6 More Models and Application

Basic Large Models

Overview

I want to summarize these models first, because they are so similar. And different models may just have tiny difference. So I will summarize them first, and then go through them quickly. In this part, I will introduce these stuff:

- Model architecture.
- Training Strategy

After that, I will separate Large models into two parts: **Large Language Model** and **Vision Fundamental Model** to quickly introduce some representative works. Note that, recently, there is a boom of large models, so maybe I miss some work.

Basic Large Models

Model architecture

Basically, all of large models use the **Encoder-Decoder Architecture**. In this way, Encoder is to project the low-level information to high-dimension space, and Decoder is to recover the abstract information to human-readable signals.

At present, most of the encoder and decoder ara made up by transformers. However, there still are some differences, we can split them into these parts:

- Full Encoder-decoder;
- Encoder-only;
- Decoder-only.

	Full	Encoder Only	Decoder Only
Model Name	T5[23]	BERT[5]	GPT-1[21]

Table 1: The example of different model structure.

Basic Large Models

Model architecture

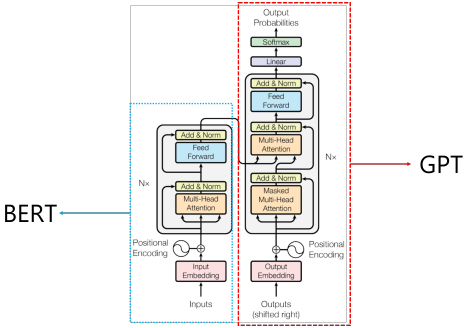


Figure 17: The simple illustration of T5, BERT and GPT.

Basic Large Models

Training Strategy

For the training, we have two parts: **From Scratch** and **Fine-tune**. When we get the large models, we will use training **From Scratch**, however, it will cost so much money. In contrast, **Fine-tune** only use comparable small computation resource. For the training from the scratch, I mainly introduce:

- Self-supervised Learning

For the fine-tuning method, I have already introduced, they are Prompt Leaning, RLHF and LoRA.

Self-supervised Learning

Overview

Compared with the supervised learning, the unsupervised and self-supervised may not need the so many labels. In this way, using these kind of training method can save the labor to label the image and makes quickly get the new model on new data.
Some typical unsupervised learning:

- Clustering

For self-supervised learning, we use the data itself as the label and hope the model can learn the intrinsic information in the data. Typically, we have several ways of self-supervised learning:

- Predict the masked parts;
- Predict the next part.

Large Language Model(**LLM**)

Overview

There are lots of great work of large language models, and I will pick some of them to introduce. The reason why I pick them is that they have different features.

- **BERT**. The first-batch of LLM using the transformer, having different training method compared with the GPT-style;
- **T5**. The large language model is proposed by Google.
- **GPT Series**. A series of work proposed by the OpenAI, and ChatGPT is based on the GPT-3.5;

However, there still are some interesting and fascinating models, such as LLaMA, PaLM and so on. As the time limit, I will not introduce them.

BERT

Bidirectional Encoder Representations from Transformers(BERT)[5].

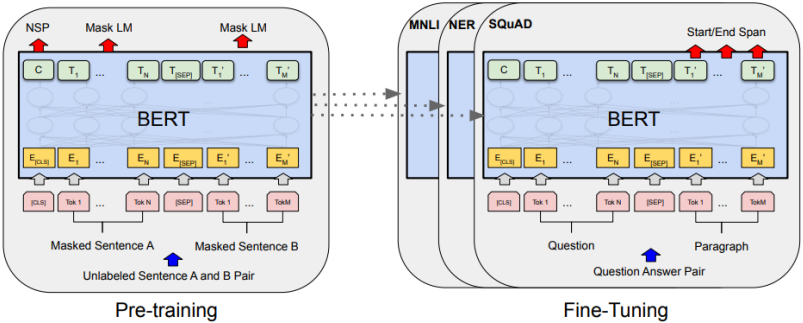


Figure 18: The pre-training and fine-tune of BERT[5].

T5

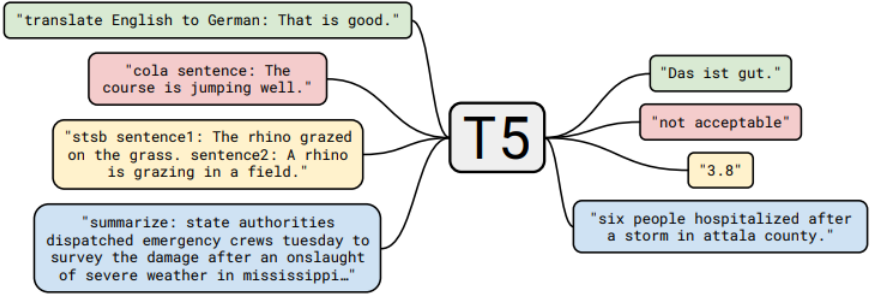


Figure 19: A diagram of our text-to-text framework. Every task we consider—including translation, question answering, and classification—is cast as feeding our model text as input and training it to generate some target text. This allows us to use the same model, loss function, hyperparameters, etc. across our diverse set of tasks. It also provides a standard testbed for the methods included in our empirical survey.

GPT-series

From the GPT-1, they gradually improve it to develop the GPT-2[22], GPT-3[4] and so on. However, they do not have so much update on the core technologies. They just use more data and more weights to train. And the performance's increasing also proves the advantages of the large model.

	Release	#Parameters	Model Store	Pre-trained Data
GPT-1	2018.6	0.117B	0.468GB	$\sim$ 5GB
GPT-2	2019.2	1.5B	6GB	40GB
GPT-3	2020.5	175B	700GB	45TB

Table 2: The comparison of GPT-series. The model size is computed if the weights are store in Float32. $1B = 10^9.10^9 \times 4bits = 4 \times 10^9 bytes = 4 \times 10^6 KB = 4 \times 10^3 MB = 4GB$

Large Language Model

Summary

Why the LLM is so charming that almost everyone pay attention on?

- In-context Learning
- Chain-of-thoughts

Vision Fundamental Model(**VFM**)

Overview

There are lots of models for computer vision. They are called **Backbone** or **Vision Fundamental Model**.

- **ResNet.** The most famous CNN.
- **ViT.**[6] The first transformer-based computer vision model.
- **MAE.** The masked training method of the computer vision task.
- **Stable Diffusion.** The diffusion model is used in the image generation, which attracts people's attention.
- **CLIP.**CLIP (Contrastive Language-Image Pre-Training)[20] is a neural network trained on a variety of (image, text) pairs.

ResNet

ResNet is a simple structure but solves the gradient vanish problem, it just add a skip connection between the input and the output of the conventional layers. As the development of DL, this simple structure is also used in the transformers.

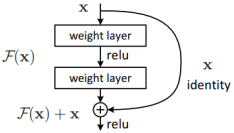


Figure 20: The residual block[8].

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

ViT

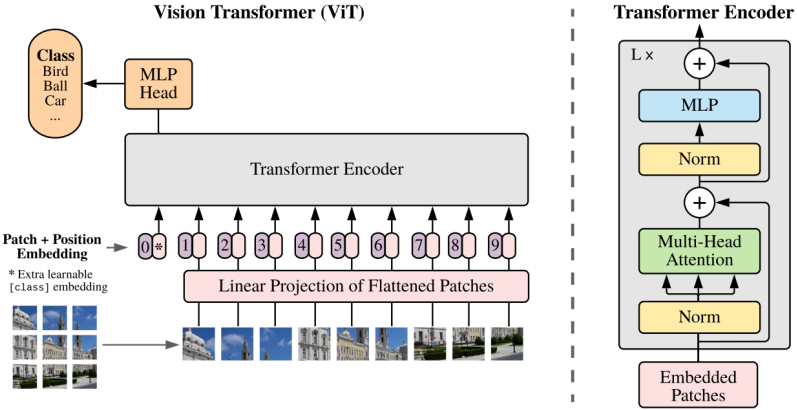


Figure 21: The overview of the ViT [6].

MAE

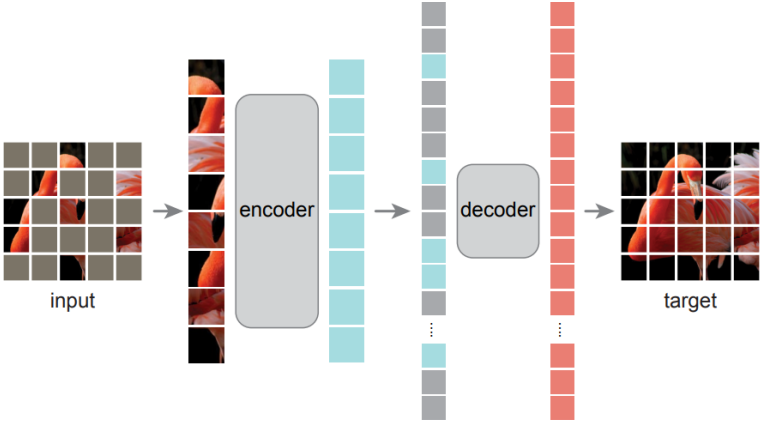


Figure 22: The overview of the MAE[7].

Stable Diffusion

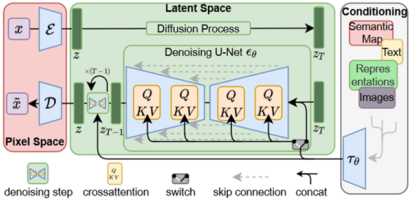


Figure 23: The architecture of the stable diffusion.

The stable diffusion[24] is one of the most famous diffusion recently.
Using the encoder to map the feature to the latent space
Combining the transformer with the diffusion to get the better conditions.

CLIP

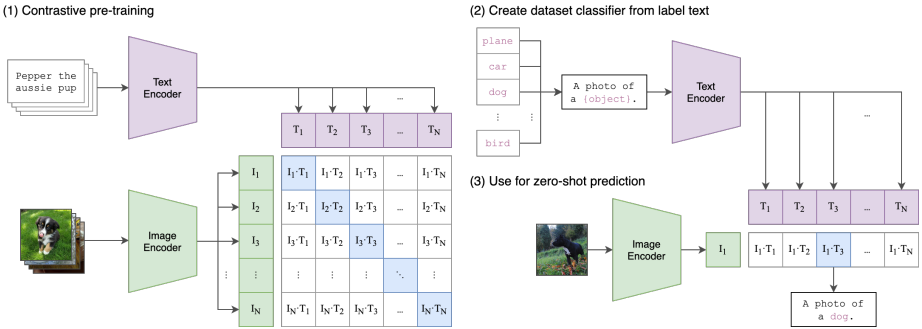


Figure 24: The overview of the Contrastive Language-Image Pre-Training(CLIP)

Contents

1. Introduction
2. Development
3. Basic Knowledge
4. Advanced Knowledge
5. Basic Large Models
6. More Models and Application

More Models and Application

Overview

We have introduce so many large models, and we definitely have the following question: How can we use it? Although the large model is powerful, they still have limited application scenes. In this part, I will firstly introduce some new application of large models, and then try to summarize the ways using the large model.

Bird's Eye View(BEV)

The BEV is widely known by us when it comes on 2021 Tesla AI day. And Tesla uses this model to merge all cameras into one bird's eye view, which can solve some hard problems in AD.

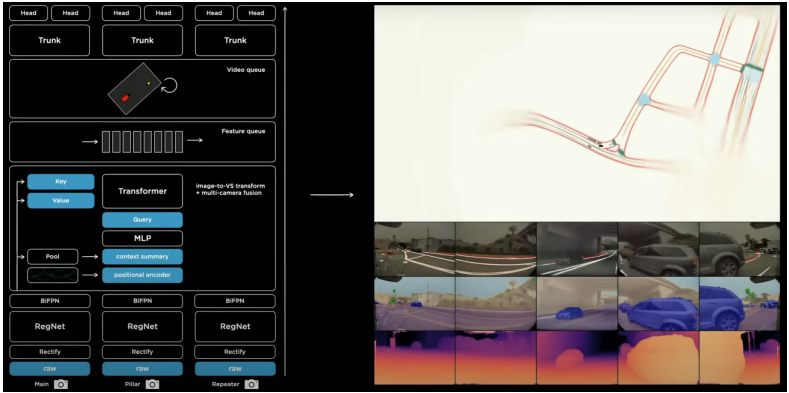
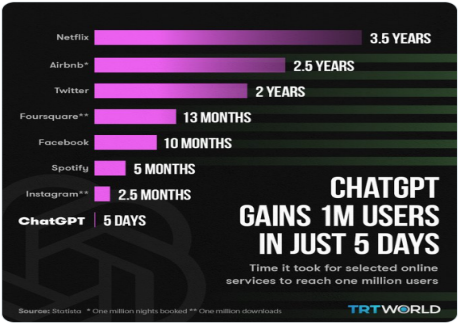


Figure 25: The overview of the Tesla solution.

ChatGPT

This application is developed by the OpenAI and the number of users increase so fast. And it can help us do the tedious works in our daily work. We trained this model using Reinforcement Learning from Human Feedback (RLHF), using the same methods as InstructGPT, but with slight differences in the data collection setup.



Bing AI

As the one of the investor is OpenAI, MicroSoft firstly combines the ChatGPT with their own searching engine–Bing. At present, everyone can use the Bing AI to search the information on the Internet. Meanwhile, the MicroSoft provides different style to satisfy various kinds of aims.

Figure 26: The Bing AI page.

DriveGPT

After the bloom of GPT, the driving GPT is proposed by HAOMO.ai

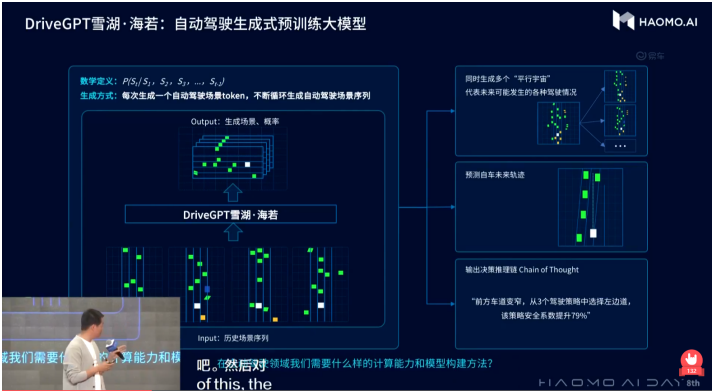
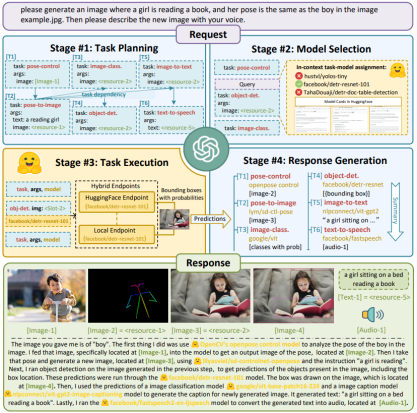


Figure 27: The DriveGPT [1].

AutoGPT

Auto-GPT[3] is an experimental open-source application showcasing the capabilities of the GPT-4 language model. This program, driven by GPT-4, chains together LLM "thoughts", to autonomously achieve whatever goal you set. As one of the first examples of GPT-4 running fully autonomously, Auto-GPT pushes the boundaries of what is possible with AI.

JARVIS



The HuggingGPT[26] uses the LLM as a controller make full use of the downstream task models.

Figure 28: The pipeline of huggingGPT solves problems

Graphical Generation

The bloom of the diffusion makes the image generation more vivid than that of GANs, so there are organizations using this technology to develop some interesting products, such as stability.ai and Midjourney.

ControlNet

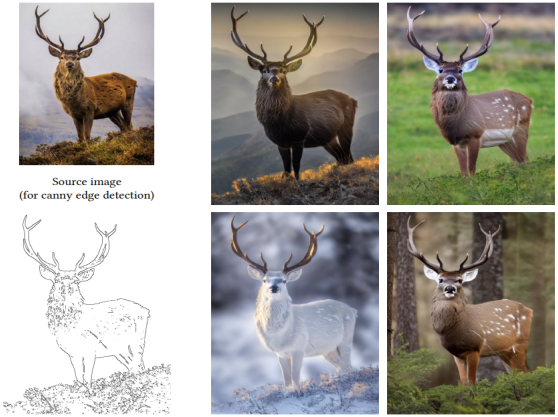
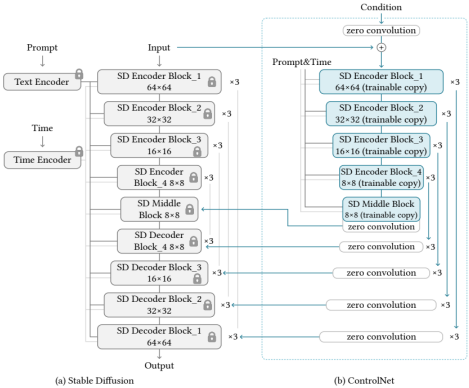


Figure 29: The demo of the ControlNet.

ControlNet



The ControlNet[30] is add an extra network to control stable diffusion to generate the images basing on the need of users.

Figure 30: The overview of the control net used in Stable Diffusion.

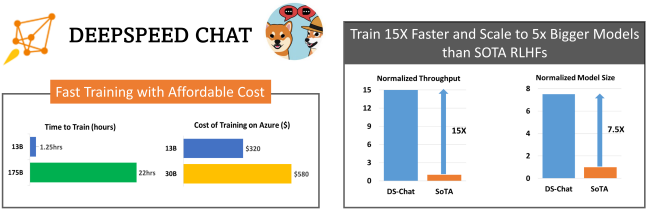
XXX Anything

As the power of large models, researchers can use them to solve the zero-shot learning.

- Segmenting Anything[12]
- Inpainting Anything[29]
- Relate Anything[16]
- OVSeg[15]
- Grounded Segmenting Anything[10]
- Segment Everything Everywhere All at Once[32]

DeepSpeed

DeepSpeed[17] empowers ChatGPT-like model training with a single click, offering 15x speedup over SOTA RLHF systems with unprecedented cost reduction at all scales



Easy-Breezy Training

A complete end-to-end RLHF training experience with a single click

High Performance System

Hybrid Engine achieves 15X training speedup over SOTA RLHF systems with unprecedented cost reduction at all scales

Accessible Large Model Support

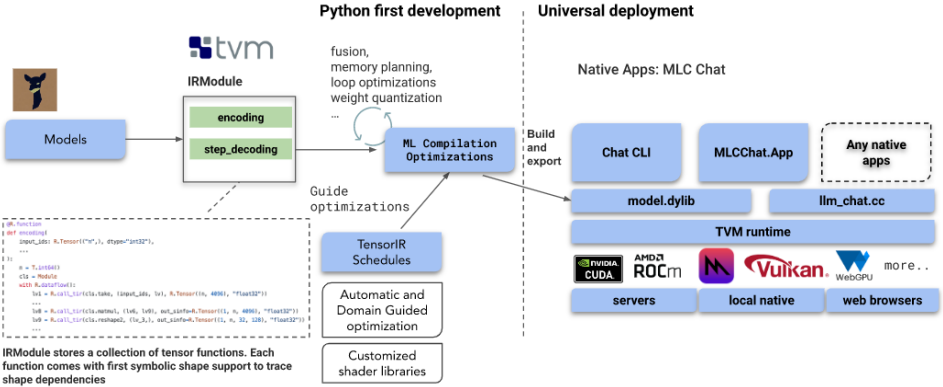
Training ChatGPT-Style models with tens to hundreds of billions parameters on a single or multi-GPUs through ZeRO and LoRA

A Universal Acceleration Backend for RLHF

Support InstructGPT pipeline and large-model finetuning for various models and scenarios

MLC LLM

MLC LLM[18] is a universal solution that allows any language models to be deployed natively on a diverse set of hardware backends and native applications, plus a productive framework for everyone to further optimize model performance for their own use cases.



Thanks

Reference I

Haomo.ai live.
`https://live.yiche.com/live/266120.html`.
2022-05-10.

Arena, 2023.

Auto-GPT.
Autogpt, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al.
Language models are few-shot learners.
Advances in neural information processing systems, 33:1877–1901, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova.
Bert: Pre-training of deep bidirectional transformers for language understanding.
arXiv preprint arXiv:1810.04805, 2018.

Reference II

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al.
An image is worth 16x16 words: Transformers for image recognition at scale.
arXiv preprint arXiv:2010.11929, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick.
Masked autoencoders are scalable vision learners.
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.
Deep residual learning for image recognition.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Reference III

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen.
Lora: Low-rank adaptation of large language models.
arXiv preprint arXiv:2106.09685, 2021.

IDEA-Research.
Grounded-segment-anything, 2023.

Vinay Iyengar.
The promise and perils of large language models, August 2022.
Accessed on: 2022-09-20.

Reference IV

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick.
Segment anything.
arXiv:2304.02643, 2023.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton.
Imagenet classification with deep convolutional neural networks.
Communications of the ACM, 60(6):84–90, 2017.

Brian Lester, Rami Al-Rfou, and Noah Constant.
The power of scale for parameter-efficient prompt tuning.
arXiv preprint arXiv:2104.08691, 2021.

Reference V

Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu.
Open-vocabulary semantic segmentation with mask-adapted clip.
arXiv preprint arXiv:2210.04150, 2022.

Luodian.
Relateanything, 2023.

Microsoft.
Deepspeed, 2023.

mlc ai.
mlc-llm, 2022.

Reference VI

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al.
Training language models to follow instructions with human feedback.
Advances in Neural Information Processing Systems, 35:27730–27744, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al.
Learning transferable visual models from natural language supervision.
In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al.
Improving language understanding by generative pre-training.
2018.

Reference VII

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al.
Language models are unsupervised multitask learners.
OpenAI blog, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu.
Exploring the limits of transfer learning with a unified text-to-text transformer.
The Journal of Machine Learning Research, 21(1):5485–5551, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer.
High-resolution image synthesis with latent diffusion models.
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

Reference VIII

Timo Schick and Hinrich Schütze.
Exploiting cloze questions for few shot text classification and natural language inference.
arXiv preprint arXiv:2001.07676, 2020.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang.
Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface.
arXiv preprint arXiv:2303.17580, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin.
Attention is all you need.
Advances in neural information processing systems, 30, 2017.

Reference IX

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le.
Finetuned language models are zero-shot learners.
arXiv preprint arXiv:2109.01652, 2021.

Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, and Zhibo Chen.
Inpaint anything: Segment anything meets image inpainting.
arXiv preprint arXiv:2304.06790, 2023.

Lvmin Zhang and Maneesh Agrawala.
Adding conditional control to text-to-image diffusion models.
arXiv preprint arXiv:2302.05543, 2023.

Reference X

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al.
A survey of large language models.
arXiv preprint arXiv:2303.18223, 2023.

Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee.
Segment everything everywhere all at once.
arXiv preprint arXiv:2304.06718, 2023.