Large Language Models

Junxian He
May 10, 2024
Final Exam

1. Two, double-sided A4-size cheatsheets
2. 2-hour duration
3. Contents cover both before mid-term and after mid-term, while emphasizing more on after mid-term
4. Format similar to mid-term exam, mixed multi-choice and open-ended questions

Will make more formal announcement on Canvas
Recap: Pretraining

Source Data A (maybe a different task)

Train on data A first

Model

Target Data B

Then train on data B

Model

For supervised training, data A is often limited

How can we find large-scale data A to train?
Recap: BERT

Mask language modeling

Construct a synthetic task from raw text only
Can be made very large-scale

Is Bert a language model? Is it a generative model?

Generative Pre-Training (GPT)

Radford et al. Improving Language Understanding by Generative Pre-Training. 2018
Is Next Token Prediction Useful?

Ok, language modeling can be used as pretraining, but is a language model itself useful for some tasks directly?

In the late 1980s the Hong Kong Government anticipated a strong demand for university graduates to fuel an economy increasingly based on services. Sir Sze-Yuen Chung and Sir Edward Youde, the then Governor of Hong Kong, conceived the idea of another university in addition to the pre-existing two universities, The University of Hong Kong and The Chinese University of Hong Kong. Planning for the "Third University", named The Hong Kong University of Science and Technology later, began in 1986. Construction began at the Kohima Camp site in Tai Po Tsai on the Clear Water Bay Peninsula. The site was earmarked for the construction of a new [ ]

Completion

This task seems useless in practice
GPT-2

Next token prediction can unify many tasks

Machine translation:

Chinese: 今天是学期的最后一天。
English:

Completion is very general

This was an early form of prompting, that is widely discussed today

Question answering:

Q: What is the capital of the United States?
A:

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:  

2. cheese => ...........................................  

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:  

2. sea otter => loutre de mer  

3. cheese => ...........................................  

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:  

2. sea otter => loutre de mer  

3. peppermint => menthe poivrée  

4. plush giraffe => girafe peluche  

5. cheese => ...........................................  

In-Context Learning

Brown et al. Language models are few-shot learners. 2020
Almost all text tasks can be expressed with a unified format, no matter whether it is classification or generation.
Large Language Models

Language Model

Large-scale Data
Large-scale Parameters
Large-Scale Compute

Large Language Model
Scaling Law

Just scaling up is the main factor to drive the main AI progress in the past decades

Scale increases exponentially

Kalplan et al. Scaling Laws for Neural Language Models. 2020
Scaling Law

https://vectorinstitute.ai/large-language-models-prompting-and-peft/
How are LLMs Developed?

Text pretraining

GPT-3

Instruction tuning

InstructGPT

code+text pretraining

Code-davinci-002 (gpt3.5 base model)

Alignment by sft/RLHF

Text-davinci-002/text-davinci-003/chatGPT
The LLM Development Stages

Pretraining → Instruction Tuning → Preference Learning (RLHF)

- **Pretraining**
  - 1000s of GPU
  - Months of training
  - Large training data, low quality

- **Instruction Tuning**
  - 1-100 GPUs
  - Days of training
  - Small training data, high quality

- **Preference Learning (RLHF)**
  - 1-100 GPUs
  - Days of training
  - Small training data, high quality
The LLM Development Stages

- Pretraining
  - 1000s of GPU
  - Months of training
  - Large training data, low quality

- Instruction Tuning
  - 1-100 GPUs
  - Days of training
  - Small training data, high quality

- Preference Learning (RLHF)
  - 1-100 GPUs
  - Days of training
  - Small training data, high quality
Code Data in Pretraining

A large amount of code data (e.g. Github repos) is mixed with text data during pretraining

1. Coding ability is important in practice
2. Coding may help improve reasoning
Cross-Lingual Transfer in Pretraining

1. We know that ChatGPT is also good at other languages (e.g. Chinese), even thought it is dominantly optimized on English.

2. The abilities learned on English may easily transfer to other languages with small data from that language.
After Pretraining

1. Fluent text generation
2. In-context learning
3. World knowledge
4. Code understanding and generation
The LLM Development Stages

- **Pretraining**: 1000s of GPU Months of training
  - Large training data, low quality

- **Instruction Tuning**: 1-100 GPUs Days of training
  - Small training data, high quality

- **Preference Learning (RLHF)**: 1-100 GPUs Days of training
  - Small training data, high quality
Instruction Tuning

Also named as Supervised Fine-Tuning (SFT)

The main difference from traditional supervised learning is on DIVERSITY of the data

Prompting is the key to break task boundaries
Instruction Tuning

The chat data is naturally very diverse, covering many tasks.
Why Do we Need Instruction Tuning?

What is the capital of the United States?

LM before SFT

What is the capital of China?
What is the capital of UK?
What is the capital of Canada?

LM after SFT

Washington

It aligns with user’s intents better because we explicitly teach the model SFT is also viewed as a process to align the model with humans.
Difference from Traditional Supervised Learning

1. Instruction tuning typically does not need that much data for normal tasks, it was considered most of abilities are already learned during pretraining, SFT only triggers it out

However, this point only applies to relatively easy tasks.

Pretraining is extremely multi-tasking instruction tuning, pretraining and SFT may not need to have an explicit distinction
Difference from 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.

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

No parameter update

Instruction tuning, by explicitly teaching the model through gradient descent, can generally work better.

Instruction tuning is more efficient at inference time.
Reinforcement Learning from Human Feedbacks (RLHF)

Pretraining → Instruction Tuning → Preference Learning (RLHF)

1000s of GPU
Months of training

1-100 GPUs
Days of training

Large training data, low quality

Small training data, high quality

1-100 GPUs
Days of training

Small training data, high quality
Ouyang et al. Training language models to follow instructions with human feedback. 2022
Standard RL objective, \( r(x,y) \) is the reward model

\[
\text{objective}(\phi) = E_{(x,y) \sim D_{\pi_{RL}^{\phi}}} [r_\theta(x, y)] - \beta \log \left( \frac{\pi_{RL}^{\phi}(y \mid x)}{\pi_{SFT}^{\phi}(y \mid x)} \right) + \gamma E_{x \sim D_{\text{pretrain}}} \left[ \log(\pi_{RL}^{\phi}(x)) \right]
\]

KL divergence with the SFT model

Pretraining task
Why do we need RL here? Why not SFT only?

1. Annotating high-quality responses is expensive and difficult for humans

2. Providing ranking/classification feedbacks is much easier

Some analogy: A swimming coach cannot directly compete with the player, but can provide helpful feedbacks to improve the player

In most cases, we cannot write as good as ChatGPT, but we can tell which one is better from two ChatGPT responses?
Thoughts: How can humans supervise models with super-human intelligence?

This direction is called scalable oversight

Fundamentally, RL is not supervised training, and provides different supervision signals
Open Challenges

• How to supervise stronger-than-human models?
• Models Hallucinate (generated contents are not reliable)
• Training Efficiency — how to use less resources to train a good model?
  ▶ Smaller model (new arch, quantization, pruning…)
  ▶ Smaller data (data evaluation, data quality)
  ▶ Better infra (more efficient implementations)
• Inference efficiency
  ▶ how to deploy models with smaller cost? (Model compression, new arch…)
  ▶ Decoding speedup... (recall how we talked autoregressive decoding is sequential)
• Evaluation — always hard..
• Multimodal — how to fuse different modalities better (arch challenges)
• AI Safety

......
Ending Remarks

**Discriminative**
- Linear Regression
  - Logistic Regression
- Generalized Linear Models
  - Kernel SVM

**Generative**
- Naive Bayes
  - PGM
- Expectation Max
  - MLE
  - MAP
  - HMM

**Deep Learning**
- RNN, CNN, Transformer
  - Backpropagation
- VAEs GANs

**Reinforcement Learning**
The unsupervised learning ones can actually do both, and semi-supervised.