Large language models

Herman Kamper

to-do TOC

A high-level introduction

Intro to large language models by Andrej Karpathy [slides]

GPT is just a transformer language model

Not an encoder-decoder model.

At inference time, it predicts one word at a time

Picture **to-do**

At training time, it predicts all the next words

Picture **to-do**

Uses masking to make sure that what happens during training matches what happens during inference.

The basic architecture, I think, is still (Radford et al., 2018).

From a language model to an assistant

Three steps:

1. Pretraining: Standard next-word prediction task as above
2. Finetuning
3. Reinforcement learning from human feedback (RLHF)

2. Finetuning

Get high-quality data where we add e.g. <user> and <assistant> prompts:

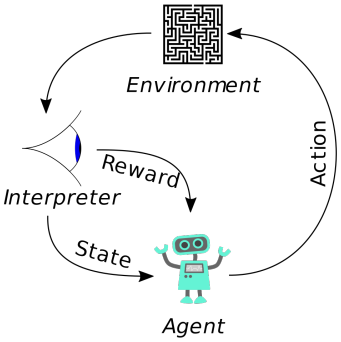
```
<user>
Can you help me with this code? It seems like there is
a bug.
print("hello world)

<assistant>
It looks like you forgot to close the string passed to
the function print. You have to add a closing quote to
properly terminate the string. Here is the corrected
function:

print("hello world")

Let me know if I can help with anything else!
```

You can think of this as a supervised task, but really it is still exactly the same task as used during pretraining. So the dataset is just swapped out, and trianing is continued.

It is a little bit amazing that all of the knowledge gained during pretraining isn't wiped out.

3. Reinforcement learning from human feedback (RLHF)

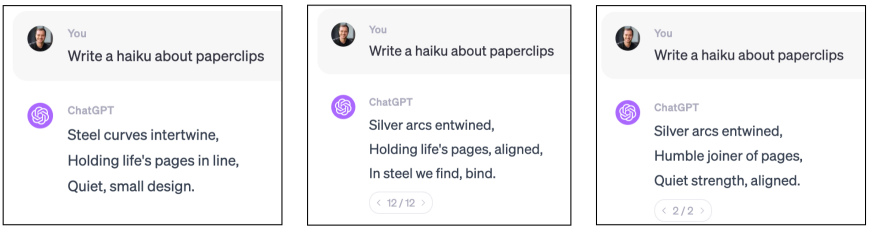
Reinforcement learning:[1]



What reward function should we use?

We learn a reward function by asking humans to rank possible outputs:[2]

Simple example: it's much easier to spot a good haiku than it is to generate one:



In this way they get even more assistant-like behaviour from the model (Ziegler et al., 2020).

But Rafailov et al. (2023) and others say that it isn't really necessary to use reinforcement learning to update the model with the learned reward function: you can do it directly through supervised learning.

[1] Figure from Wikipedia.
[2] Figure by Andrej Karpathy.

PPO links

Acknowledgments

....

Further reading

...

References

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training", 2018.

D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, "Fine-tuning language models from human preferences," *arXiv*, 2020.

R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," in *NeurIPS*, 2023.