llmIntroNotebook

March 22, 2023

1 Resources

1.1 Popular Models:

- ChatGPT (what everyone is using): https://chat.openai.com/auth/login
- OpenAI playground (what you can use when ChatGPT is down): https://platform.openai.com/playground
- LLaMa repo: https://github.com/facebookresearch/llama
- Alpaca repo: https://github.com/tatsu-lab/stanford_alpaca
- BLOOM repo: https://huggingface.co/bigscience/bloom forked from https://github.com/bigscience-workshop/Megatron-DeepSpeed
- Bard (on waitlist as of March 2023): https://bard.google.com/

1.2 Helpful Links

- How GPT (and LLMs generally) work: https://dugas.ch/artificial_curiosity/GPT_architecture.html
- Colab notebook to train own LLM: https://colab.research.google.com/github/huggingface/notebooks/blob/n
- HuggingFace demo with different LLMs: https://huggingface.co/spaces/olivierdehaene/chat-llm-streaming

2 Intro

2.1 ChatGPT (OpenAI) Response to "Current State of Large Language Models in 2023"

As of 2023, large language models have continued to evolve and advance in various ways, with several notable developments and applications:

1. Continual improvement in model size and performance: Language models have continued to increase in size and complexity, with recent models such as GPT-4 and T5 exceeding trillions of parameters. These models have achieved state-of-the-art results on a range of language tasks, including language generation, translation, and understanding.

2. Expansion of multilingual capabilities: Many large language models are now trained to understand and generate text in multiple languages, including low-resource languages, making them more accessible and useful for a wider range of users worldwide.

3. Enhanced fine-tuning capabilities: Fine-tuning allows for the adaptation of pre-trained models to specific domains or tasks, such as conversational AI, question answering, and sentiment

analysis. Recent advances in fine-tuning techniques have made it easier to fine-tune models for specific applications with fewer training examples.

4. Advancements in model compression and optimization: As models grow larger, there is a need to compress and optimize them for efficient deployment in resource-constrained environments. Recent techniques such as pruning, distillation, and quantization have made it possible to reduce the size and computational cost of large models without significant loss in performance.

5. Increased attention to ethical considerations: With the growing influence of large language models, there has been increasing attention to ethical considerations surrounding their use, including issues related to bias, privacy, and transparency. Researchers and practitioners are working to address these concerns through improved data collection, model training, and deployment practices.

Overall, large language models have continued to push the boundaries of natural language processing and have demonstrated significant potential for a wide range of applications. However, ongoing efforts are needed to ensure their responsible and ethical use.

2.1.1 Analysis of response

This response is a great example of why ChatGPT isn't going to replace people directly, it (and other Large Language Models) need fine tuning for specific use cases.

This is a 'vanilla' response. It's so generic that it says effectively nothing. If someone was unfamiliar with LLM and Natural Language Processing/Understanding, they may feel this ChatGPT's response is suitable. They may even publish an article about it and present themselves as an expert.

Unfortunately, such articles are then used to further train LLMs, reinforcing this 'Cycle of Stupidity'(tm).

2.1.2 Split in LLMs Business Models - A lot of a little or a little of a lot

Broadly speaking, in technology there are two business models: 1. Get a lot of users by charging a little-to-nothing for the service 2. Charge a little number of users a lot of money

ChatGPT and it's ilk are type 1. Type 2 are not discussed generally because it requires a lot of work, and they need to be private.

The problem here is two fold. First, the more accessible type 1 models optimize toward broad appeal. Meaning they will produce results understandable and applicable for the general population. Second, the more money and time a business can dedicate to creating its own AI routines, the more competitive advantage it gains.

We now have a situation where AI routines are created by highly specialized engineers, but there is little opportunity for new engineers to learn how the routines work. Leading to 'black box' operations similar to what we saw when Enterprise Systems first started in the 80s.

Additionally, the average person will think AI is as dumb if not dumber then they are. This could create a false sense of security. That said, as long as non-technical people are in charge of businesses decisions related to AI, they will also tend toward broadly appealing AI outcomes, so we likely won't see large scale AI implementation in major firms for quite some time.

3 How Large Language Models (LLMs) Work

Great summary here: https://dugas.ch/artificial_curiosity/GPT_architecture.html Although focusing on ChatGPT, the concepts are applied in most LLMs.

3.1 General process:

Any text based Deep Learning, AI, Machine Learning process needs to translate text into numbers, predict numeric outputs from numeric input, then translate the output numbers back into text.

```
graph TD
    A(Block Text) -->|Tokenizing| B(Text as Small-Phrases)
    B -->|Encoding + Embedding| C(Phrases as Numeric Input Matrix)
    C -->|Neural Network Operations| D(Predicted Numeric Output Matrix)
    D -->|Decoding| E(Predicted Output Text)
```

3.2 Tokenizing: Split large text into small terms/phrases

This article is a fantastic introduction to tokenizing, encoding, and embedding: https://datajenius.com/2022/03/13/a-deep-dive-into-nlp-tokenization-encoding-word-embeddings-sentence-embeddings-word2vec-bert/

"Tokenizing" is essentially a fancy way to say we split large text into smaller parts.

3.3 Encoding and Embedding: Make terms in to numbers

Generally, Artifical Intelligence / Machine Learning processes are all extensions of a few statistical / linear algebra techniques.

Advancements in AI/ML are often faster ways to apply said techniques and novel ways to transform non-numerical data into numerical data.

The exact processes used are not that important to understand in practice, but conceptually are not very difficult to learn, and provide the foundation for many AI/ML operations, so I highly encourage it.

Again, please read this article, so much will make sense once you get through it: https://datajenius.com/2022/03/13/a-deep-dive-into-nlp-tokenization-encoding-word-embeddings-sentence-embeddings-word2vec-bert/

3.4 Neural Network / 'Traditional' Machine Learning Operations:

To understand the foundations of Neural Networks and Deep Learning I highly encourage you to take these courses and complete all exercises: https://www.coursera.org/specializations/machine-learning-introduction

Working through that specialization is no small feat, but a whole world will open up to you if you can finish it.

But for the purpose of this article, going into techinical detail about what numerical operations are performed will only cause confusion.

If reading my summaries of LLM processes is frustrating to you, that is a good thing. It means you want a deeper understanding. The Coursera Specialization linked above will give you what you need - if you are willing to put in the time and effort.

3.5 Decoding: Numbers Back to Text

Last stop, taking all the stuff we did with numbers and reverse the process used to encode text to numbers. Nothing particularly complicated here; however, Deep Learning engineers do have to consider how quickly and accurately *decoding* can be executed given the original *encoding* process.

3.6 What if I don't want to learn this stuff?

Luckily, you don't have to. OpenAI and many other LLM companies make their money because people don't have the motivation and/or time to learn, build, implement, and use their own LLM.

The primary way the 'monetize' LLMs is through their Application Programming (or Process) Interface (API). The API allows an end user to pass in text and receive the model output without understanding any deep learning.

Below is an example.

4 OpenAI API

Basic config settings. Use API key from OpenAI to fine tune and run your own models

Get API key from here (requires account): https://platform.openai.com/account/api-keys

Note: OpenAIs API reference is… not good. https://platform.openai.com/docs/api-reference/ But you can guess the python methods as they generally correspond to the descriptions in documentation.

For example, completion creation is `Completion.create()` and image creation is `Image.create()`

Below is an example using the completion api. See other notebooks for more OpenAI examples: - Image Generation: https://github.com/thedanindanger/yaads-examples/blob/main/imageGeneration/imageGen.ipynb - Fine Tuning: https://github.com/thedanindanger/yaads-examples/tree/main/LinkedInPostAI

[ ]:
```
%%capture
# Set the API Key
key = open("../.config/openai.key").read()
```

[ ]:
```
# Import libraries
import openai
import pandas as pd
import importlib
```

[ ]:
```
completion_prompt = "The first derivative of x^3+2 is:"
```

[ ]:
```
# Load your API key from an environment variable or secret management service
openai.api_key = key
```

```
response = openai.Completion.create(
        model='text-davinci-003',
        prompt= completion_prompt, #p,
        max_tokens=500,
        temperature=0,
        stream=False,
    )
```

[ ]:
```
print(response["choices"][0]["text"].strip())
```

```
3x^2
```

5 Next Steps

5.1 Try BLOOM and derivitatives:

BLOOMZ trained BLOOM model: https://github.com/bigscience-workshop/xmtf#bloomz - how to download huggingface model: https://huggingface.co/docs/huggingface_hub/v0.13.3/guides/download