CS410 Text Information Systems, Fall 2020
Technology Review – OpenAI GPT, GPT-2, GPT-3
James Coffey

Introduction

The goal in developing the recent OpenAI GPT-3 (Generative Pre-Training) language model was to develop a general model that was task-agnostic and could perform language tasks with only a few examples presented as input. This goal was motivated by three drivers. First, there is limited applicability of language models if there exists the need for the tedious process of curating a sizeable supervised learning dataset for each language task. Second, supervised learning is prone to over-fitting the training data rather than fundamentally learning the task. Lastly, humans do not require a large dataset to learn from and perform a task with few or no examples.[1]

To get towards the goal of a general task-agnostic language model, many design decisions were taken from the original GPT model to GPT-2 to GPT-3 in the development history that originated in the 2017 Transformer model. Google Brain developed the Transformer model as an alternative to the dominant contemporary sequence transduction models that relied on recurrent neural networks or convolutional neural networks. Instead, the Transformer model performed transduction using an attention mechanism incorporated into an encoder and a decoder.[2] This work led to developing the original OpenAI GPT model in 2018 that dispensed with the encoder and used a decoder only Transformer model. Development of GPT first created a task-agnostic general model using a generative pre-training of model weights on a large corpus of unlabeled text and then fine-tuned the model using supervised learning for specific language tasks.[3] In 2019, OpenAI's GPT-2 model development focused on eliminating the need for subsequent supervised fine-tuning of the general model by focusing on model size's effect on the performance using only unsupervised generative pre-training to develop a general model that was task-agnostic.[4] Lastly, in 2020, OpenAI's GPT-3 furthered performance of only unsupervised generative pre-training by increasing model size and incorporating alternating dense and locally banded sparse attention patterns. This improvement resulted in a general task-agnostic model that performs well alongside many other state-of-the-art models designed for specific language tasks. [1, 5]

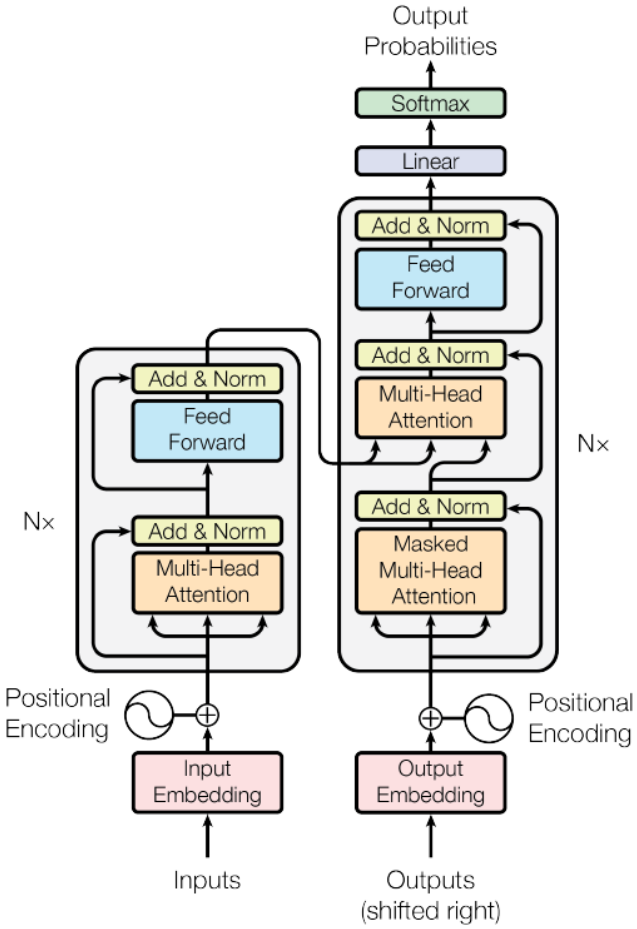

Figure 1 - Architecture of Google Brain's Transformer model [2]

Background

OpenAI used the Transformer model developed in 2017 as the basis for GPT, GPT-2, and GPT-3. Google Brain developed the Transformer to perform sequence transduction using an attention mechanism, the essential innovation in the model that allows the learning of long-range dependencies. It does this by keeping the maximum path length of the forward and backward traversals of the network shorter than competing recurrent neural network and convolutional neural network models while also maintaining a manageable computational complexity per layer of $O(n^2 \cdot d)$ where n is the sequence length and d is the representation dimension. More specifically, attention is mapping a query vector and a set of key vector, value vector pairs to an output vector. The output vector is the weighted sum of the value vector whereby a compatibility function between the query and key vectors computes the weights.[2]

Figure 1 gives the architecture of the Transformer model. Input to the encoder and decoder are tokens embedded as vector representations and are then positionally encoded due to the lack of recurrence and convolution. The encoder comprises a stack of six identical encoder layers where each layer is composed of multi-headed self-attention that feeds into a fully connected feed-forward network using a rectified linear unit (ReLU) activation function. After each sublayer in each encoder layer is a residual connection added and normalized to the attention and feed-forward layers' outputs. Each encoder layer outputs to its corresponding layer on the decoder stack. The decoder comprises six identical layers composed of the same sublayers as the encoder but with masked multi-head attention that attends to offset output embeddings. The decoder's output is to a linear transformation and softmax function to predict the next tokens' probabilities.[2]

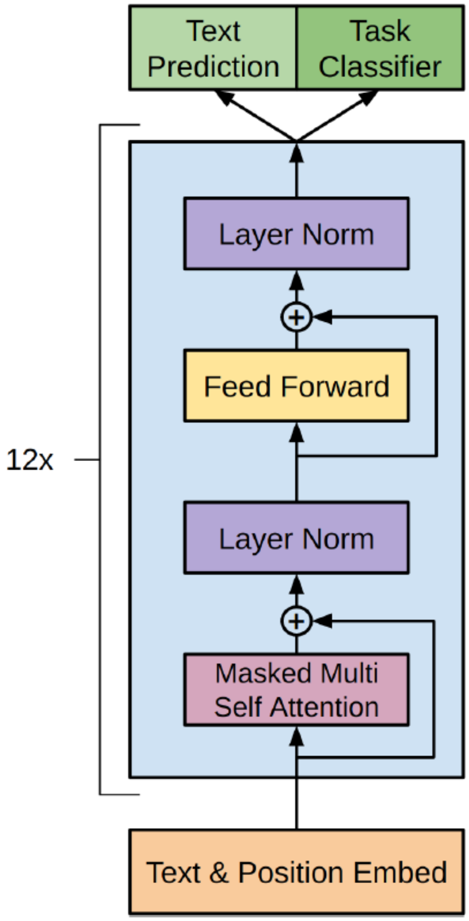

Figure 2 - GPT architecture [3]

Development from GPT to GPT-3

The original GPT model's development was to develop a general model trained on readily available unlabeled text data, which could be later fine-tuned to specific language tasks using smaller labeled supervised learning datasets. It used a decoder only Transformer model with masked multi-head self-attention layers, as shown in Figure 2. Also deviating from the original Transformer model, GPT uses a gaussian error linear unit (GELU) activation function and learned position embeddings rather than the original sinusoidal version. To pre-train the model, the model parameters were updated via stochastic gradient descent to maximize the log-likelihood over unlabeled text data. The unlabeled text data was taken from the BooksCorpus dataset as it had a long contiguous text for the model to learn long-range order. For fine-tuning, structured inputs from several supervised learning language task datasets were converted into token sequences to be read by the pre-trained model. The model parameters were tuned with stochastic gradient descent to maximize the log-likelihood of the supervised labels. At the time, this method was successful, with the authors improving the state-of-the-art performance on nine of the twelve datasets they used.[3]

Developing the GPT-2 model was to eliminate the need for fine-tuning the model to specific language tasks. To that end, the development team at OpenAI decided to keep the overall architecture of the model used for GPT and explored the effect of size on specific language tasks' performance. It was found that the model performed better on language tasks by increasing the number of parameters from the original GPT amount of 117 million (12 decoder layers) to 1.542 billion for GPT-2 (48 decoder layers) with in-between size explored as well. In addition to increased layer and parameter count, GPT-2 had layer normalization moved to the input of sub-blocks, a modified weight initialization, an expanded vocabulary, an expanded context size, and expanded batch size. In training the model, the same generative pre-training technique from GPT was used but on the much larger 8 million documents WebText dataset (compared to the 7000 documents BooksCorpus). This method was successful at the time, with the authors achieving a state of the art zero-shot performance on seven out of the eight datasets they used.[4]

Lastly, the goal of GPT-3 was to increase the performance of the general task-agnostic language model so that it could perform alongside state-of-the-art fine-tuning approaches with few-shot, one-shot, and no-shot prediction modes. To do this, the overall architecture of the model used for GPT-2 was used with two modifications so that longer-range patterns could be learned. First, the number of parameters was increased to 175 billion.[1] Second, alternating dense and locally banded sparse attention patterns were used in layers of the model in a way like that of OpenAI’s 2019 Sparse Transformer. The attention matrix's sparse factorizations help reduce the computational complexity from quadratically growing with sequence length to being $O(n\sqrt{n})$.[5] In training the model, the same generative pre-training technique from GPT-2 was used but on a much larger version of the CommonCrawl dataset modified to reduce repeat information and increase sample diversity. [1]

The result of this development, GPT-3, has been evaluated using few-shot, one-shot, and zero-shot prediction on multiple language tasks. In few-shot, the model sees the natural language task description and a few examples of the task. In one-shot, the model sees one example of the task in addition to the task description. In zero-shot, only the task description is provided. No gradient updates are performed during any of these prediction methods. The language tasks included language modeling, closed book question answering, translation, Winograd-style tasks, common sense reasoning, reading comprehension, SuperGLUE, natural language inference, arithmetic, word scrambling and manipulation, SAT analogies, news article generation, learning and using novel words, and correcting English grammar. In some of these tasks, GPT-3 has nearly matched the performance of fine-tuned models.[1]

Conclusion

As a result of the development from the original GPT to the recent GPT-3 language model, large language models have been demonstrated as a possible important element in general models that are task-agnostic and can perform language tasks with only a few examples presented as input. The success and short cycle time between model improvements (2018, 2019, 2020) indicate that this technique will likely continue to see rapid development. This development could mean less reliance on supervised learning for language tasks. It could also mean more human-like interaction with such models as they require less structured prompts.

References

[1] 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, and Dario Amodei. Language models are few-shot learners, 2020.

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

[3] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

[4] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019.

[5] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019.