Transformers for Natural Language Processing

Build, train, and fine-tune deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, and GPT-3

El ascenso de los Transformadores con motores GPT

Introducción

En 2020, Brown et al. (2020) describieron el entrenamiento de un modelo GPT-3 de OpenAI con 175 mil millones de parámetros que aprendió utilizando enormes conjuntos de datos.

La inteligencia artificial de los motores GPT-3 de OpenAI y su supercomputadora llevó a Brown et al. (2020) a realizar experimentos de "zero-shot".

De esta manera comenzó la era de los motores de inteligencia artificial en la nube.

La arquitectura de los modelos transformadores GPT de OpenAI

GPT-3 está construido sobre la arquitectura de GPT-2.

Los transformadores pasaron del entrenamiento, a la afinación fina, y finalmente a los modelos *zero-shot* en menos de tres años, entre finales de 2017 y la primera parte de 2020.

Un modelo transformador GPT-3 *zero-shot* no requiere afinación fina.

Transformer Model	Paper	Parameters
Transformer Base	*Vaswani* et al. (2017)	65M
Transformer Big	*Vaswani* et al. (2017)	213M
BERT-Base	*Devlin* et al. (2019)	110M
BERT-Large	*Devlin* et al. (2019)	340M
GPT-2	*Radford* et al. (2019)	117M
GPT-2	*Radford* et al. (2019)	345M
GPT-2	*Radford* et al. (2019)	1.5B
GPT-3	*Brown* et al. (2020)	175B

La arquitectura de los modelos transformadores GPT de OpenAI

El tamaño de la arquitectura evolucionó al mismo tiempo:

- El número de capas de un modelo pasó de 6 capas en el Transformer original a 96 capas en el modelo GPT-3
- El número de cabezales por capa pasó de 8 en el modelo Transformer original a 96 en el modelo GPT-3
- El tamaño del contexto pasó de 512 tokens en el modelo Transformer original a 12,288 en el modelo GPT-3

La arquitectura de los modelos transformadores GPT de OpenAI

El orden de la función que define la longitud máxima del camino puede resumirse como se muestra en la Tabla en notación Big O:

Layer Type	Maximum Path Length	Context Size
Self-Attention	O(1)	1
Recurrent	O(n)	100

De la afinación fina a los modelos *zero-shot*

El objetivo era entrenar transformadores con datos no etiquetados.

Dejar que las capas de atención aprendieran un lenguaje a partir de datos no supervisados.

Comenzaron a entrenar modelos transformadores con datos en bruto en lugar de depender de datos etiquetados por especialistas.

El primer paso fue comenzar con el entrenamiento no supervisado en un modelo transformador.

De la afinación fina a los modelos *zero-shot*

- **Afinación Fina (FT)**: Se realiza de la manera que hemos mostrado anteriormente con los ejemplos. Un modelo transformador se entrena y luego se ajusta para tareas posteriores.

- **Pocos Ejemplos (Few-Shot, FS)**: Representa un gran avance. Cuando el modelo necesita hacer inferencias, se le presentan demostraciones de la tarea a realizar como acondicionamiento. El acondicionamiento reemplaza la actualización de pesos, lo que el equipo de GPT excluyó del proceso.

- **Un Solo Ejemplo (One-Shot, 1S)**: Llleva el proceso más allá. El modelo GPT entrenado se presenta con solo una demostración de la tarea posterior a realizar. Tampoco se permite la actualización de pesos.

- **Sin Ejemplos (Zero-Shot, ZS)**: Es el objetivo final. El modelo GPT entrenado se presenta sin ninguna demostración de la tarea posterior a realizar.

De la afinación fina a los modelos *zero-shot*

Motivaciones que llevaron a la arquitectura de los modelos GPT:

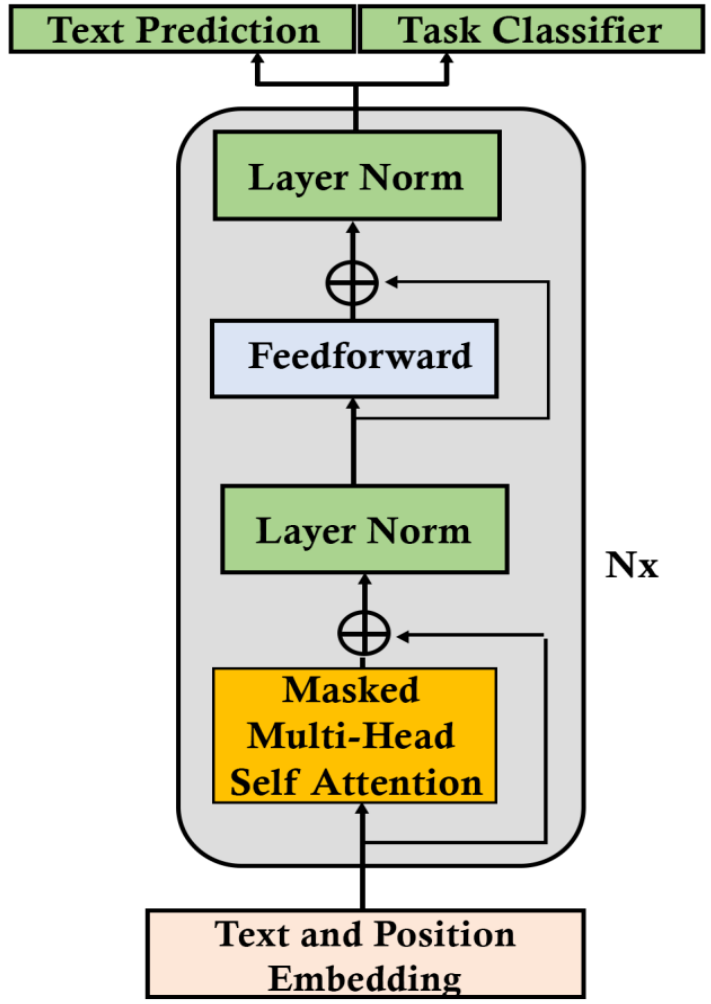
- Enseñar a los modelos transformadores cómo aprender un lenguaje a través de un entrenamiento extenso.
- Enfocarse en el modelado del lenguaje mediante el acondicionamiento del contexto.
- El transformador toma el contexto y genera la finalización de texto de una manera novedosa. En lugar de consumir recursos en aprender tareas posteriores, se centra en comprender la entrada y hacer inferencias sin importar cuál sea la tarea.

Apilamiento de capas de decodificador

Ahora entendemos que el equipo de OpenAI se enfocó en el modelado del lenguaje.

Brown et al. (2020) aumentaron drásticamente el tamaño de los modelos transformadores solo de decodificadores para obtener excelentes resultados.

Los modelos GPT tienen la misma estructura que las pilas de decodificadores del Transformer original diseñado por Vaswani et al. (2017).



Motores GPT-3

Afinación fina de GPT-3

Referencias

- OpenAI and GPT-3 engines: https://beta.openai.com/docs/engines/engines
- BertViz GitHub Repository by *Jesse Vig*: https://github.com/jessevig/bertviz
- OpenAI's supercomputer: https://blogs.microsoft.com/ai/openai-azuresupercomputer/
- *Ashish Vaswani*, *Noam Shazeer*, *Niki Parmar*, *Jakob Uszkoreit*, *Llion Jones*, *Aidan N. Gomez*, *Lukasz Kaiser*, *Illia Polosukhin*, 2017, *Attention is All You Need*: https://arxiv.org/abs/1706.03762
- *Alec Radford*, *Karthik Narasimhan*, *Tim Salimans*, *Ilya Sutskever*, 2018, *Improving Language Understanding by Generative Pre-Training*: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf
- *Jacob Devlin*, *Ming-Wei Chang*, *Kenton Lee*, and *Kristina Toutanova*, 2019, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*: https://arxiv.org/abs/1810.04805

Referencias

• *Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever*, 2019, *Language Models are Unsupervised Multitask Learners*: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

• *Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplany, 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, Dario Amodei*, 2020, *Language Models are Few-Shot Learners*: https://arxiv.org/abs/2005.14165

Referencias

- *Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel R. Bowman*, 2019, *SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems*: https://w4ngatang.github.io/static/papers/superglue.pdf
- *Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel R. Bowman*, 2019, *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding*: https://arxiv.org/pdf/1804.07461.pdf
- OpenAI GPT-2 GitHub Repository: https://github.com/openai/gpt-2
- N. Shepperd's GitHub Repository: https://github.com/nshepperd/gpt-2
- Common Crawl data: https://commoncrawl.org/big-picture/

Transformers for Natural Language Processing

Build, train, and fine-tune deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, and GPT-3