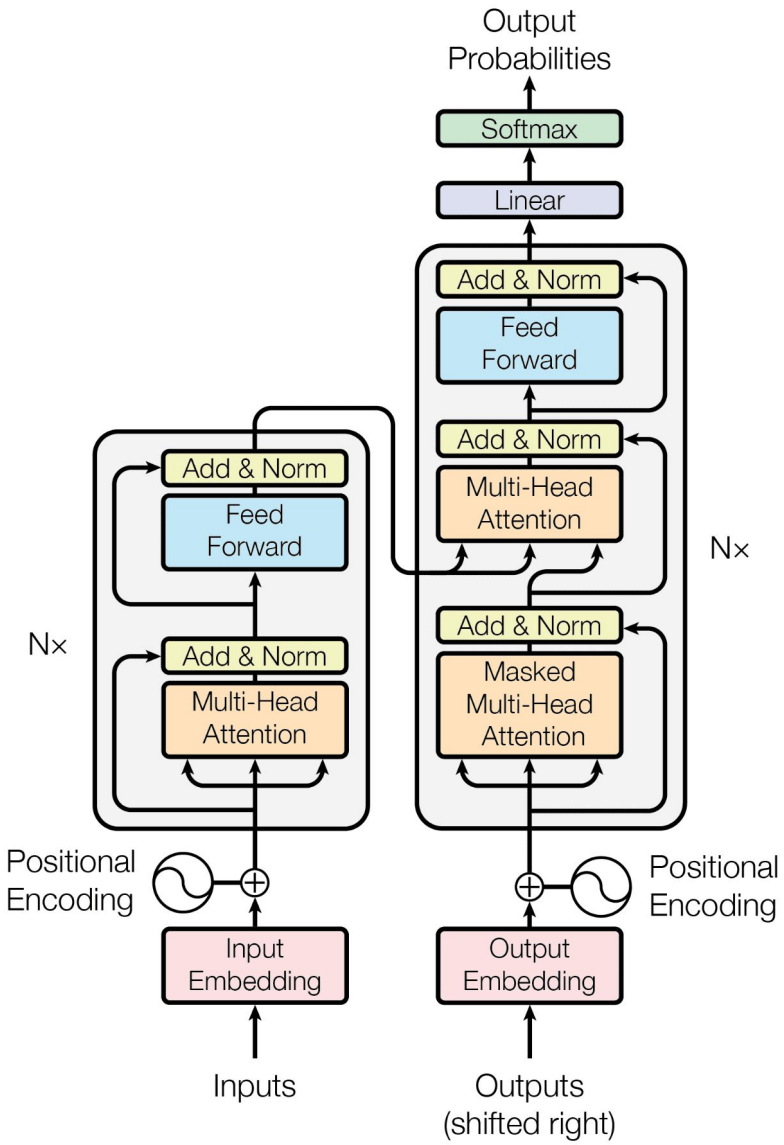
Redes Neuronales para Lenguaje Natural

2023

Grupo de Procesamiento de Lenguaje Natural
Instituto de Computación

(más) Modelos de Lenguaje

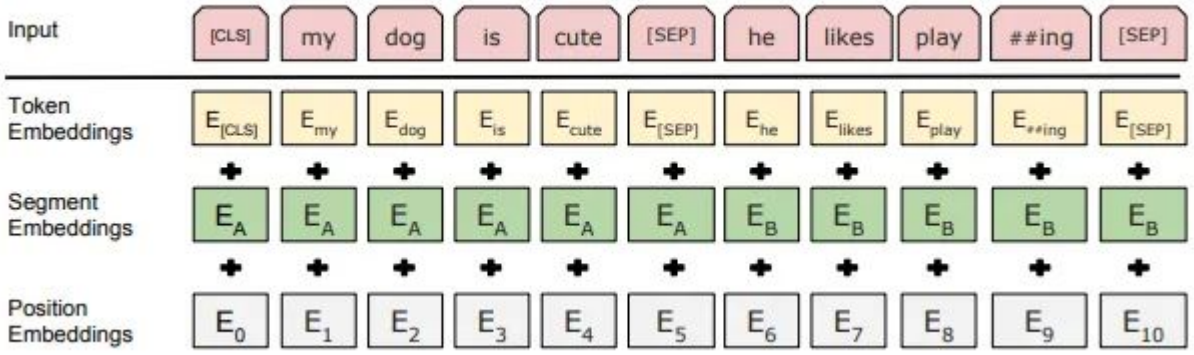
Arquitectura transformer



BERT

Bidirectional Encoder Representations from Transformers

- Utiliza solo el lado del encoder
- Entrenado con dos tareas simultáneas
  - Modelo de lenguaje enmascarado
  - Predicción de oraciones contiguas



Bert: Pre-training of deep bidirectional transformers for language understanding. (2018)
Devlin, J., Chang, M.W., Lee, K. and Toutanova, K.

BART

Bidirectional Auto-Regressive Transformers

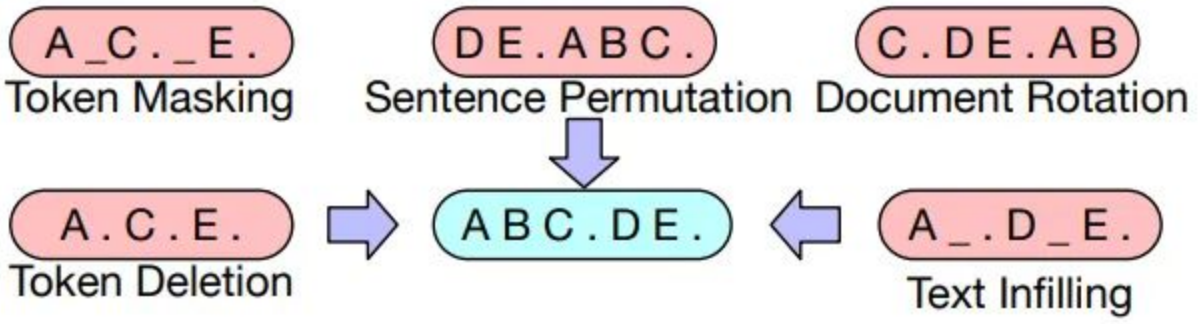
- Utiliza tanto el encoder como el decoder del transformer
- Se entrena de forma autorregresiva
- Hacemos perturbaciones a las oraciones, las procesamos con el encoder, y las tratamos de reconstruir con el decoder

Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. (2019) Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. and Zettlemoyer, L.

BART

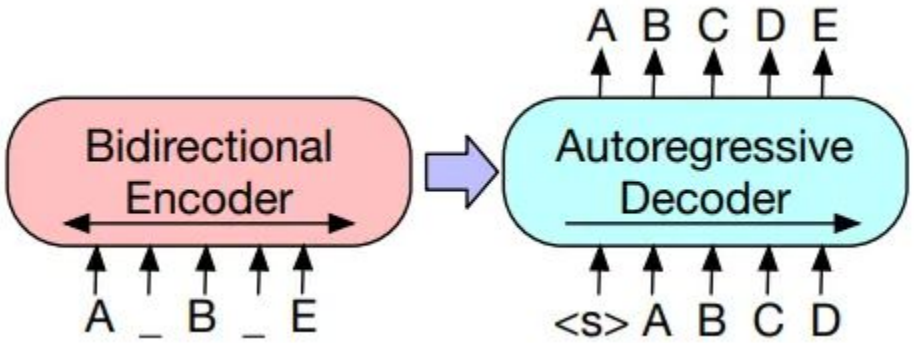
Se pueden realizar varias perturbaciones a la oración original

- Enmascaramiento (como BERT)
- Permutaciones, rotaciones
- Eliminación de tokens
- Rellenado de texto (text infilling)



BART

- El encoder procesa la oración con perturbaciones
- El decoder intenta recuperar la oración original
- Entrenado para que la probabilidad de obtener la oración original restaurada sea máxima



BART

Resultados de BART (cuando salió)

Model	SQuAD 1.1 F1	MNLI Acc	ELI5 PPL	XSum PPL	ConvAI2 PPL	CNN/DM PPL
BERT Base (Devlin et al., 2019)	88.5	**84.3**	-	-	-	-
Masked Language Model	90.0	83.5	24.77	7.87	12.59	7.06
Masked Seq2seq	87.0	82.1	23.40	6.80	11.43	6.19
Language Model	76.7	80.1	**21.40**	7.00	11.51	6.56
Permuted Language Model	89.1	83.7	24.03	7.69	12.23	6.96
Multitask Masked Language Model	89.2	82.4	23.73	7.50	12.39	6.74
BART Base						
w/ Token Masking	90.4	84.1	25.05	7.08	11.73	6.10
w/ Token Deletion	90.4	84.1	24.61	6.90	11.46	5.87
w/ Text Infilling	**90.8**	84.0	24.26	**6.61**	**11.05**	5.83
w/ Document Rotation	77.2	75.3	53.69	17.14	19.87	10.59
w/ Sentence Shuffling	85.4	81.5	41.87	10.93	16.67	7.89
w/ Text Infilling + Sentence Shuffling	**90.8**	83.8	24.17	6.62	11.12	**5.41**

T5

Text-to-Text Transfer Transformer

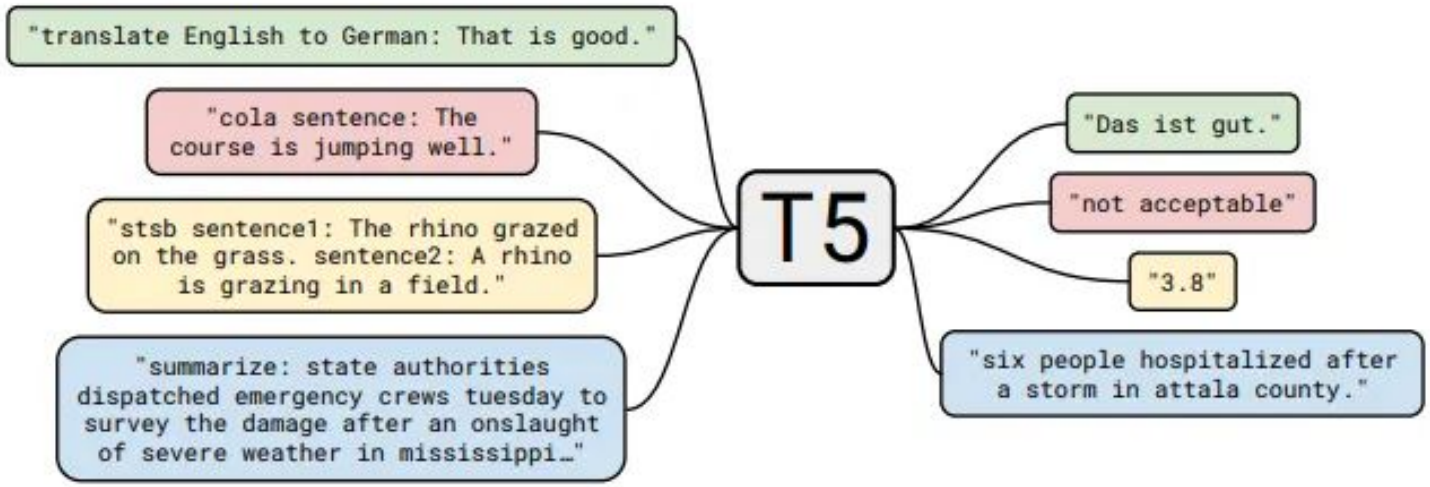
- Arquitectura igual al transformer con pequeñas simplificaciones
- Eliminan bias de la capa de normalización
- Sacan la conexión residual de la capa de normalización
- Cambios en el positional embedding

Exploring the limits of transfer learning with a unified text-to-text transformer. (2020)
Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J.

T5

Entrenamiento:

- Pre-entrenamiento autosupervisado
- Fine-tuning supervisado multitarea, donde todas las tareas se reformulan como problemas texto-a-texto
- Incluso problemas de regresión!

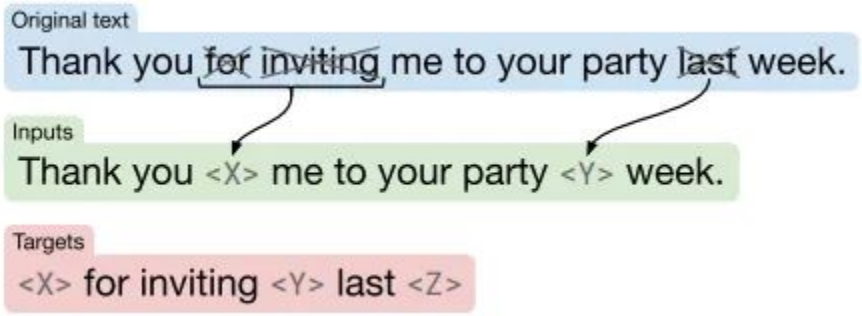


T5

El pre-entrenamiento también se formula como una tarea texto-a-texto

Extraemos palabras (y secuencias de tokens) aleatoriamente sustituidas por marcadores

La salida esperada son esas secuencias asociadas a los marcadores correspondientes



T5

Datos:

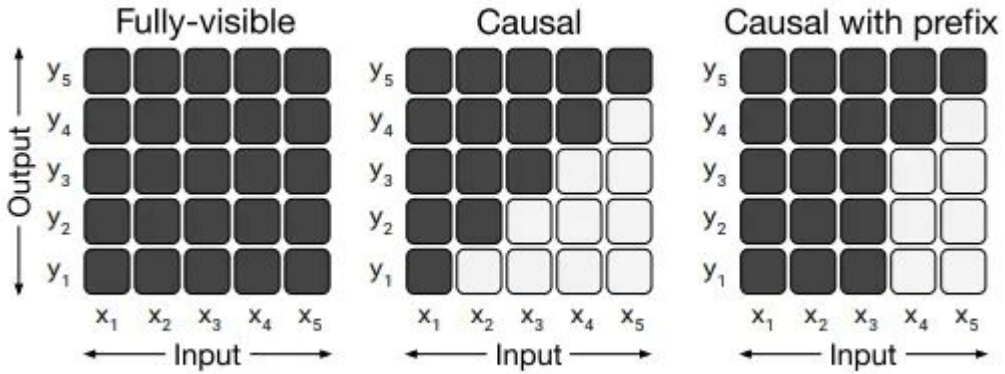
- Dataset C4 de 750 GB
- Hecho a partir del dataset Common Crawl de 20 TB
- Pero eliminan segmentos que contengan:
  - código JavaScript
  - tags HTML
  - llaves { }
  - texto con lenguaje ofensivo
  - otras cosas...

Para el pre-entrenamiento enmascaran 15% del texto aleatoriamente

T5

Permite distintos patrones de enmascaramiento de atención

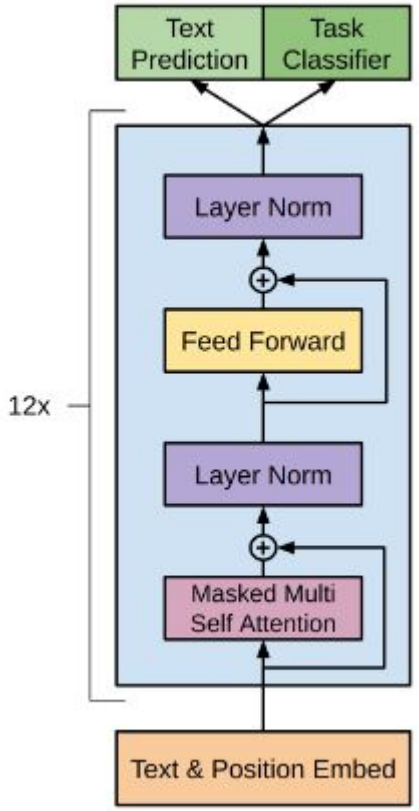
- Bidireccional completo (como BERT)
- Causal (como GPT)
- Causal con prefijo



GPT

Generative Pretrained Transformer

- Utiliza solo el lado del decoder
- Entrenado de manera causal
- Es el más famoso! Y el estado del arte para muchas tareas



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

Language models are few-shot learners. (2020) Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. and Agarwal, S.

LLaMA

Large Language Model Meta AI

Utiliza solo el decoder del transformer

Es de los modelos abiertos más utilizados

Viene en diferentes variantes según cantidad de parámetros:

- Llama2 7B, 13B, 70B
- 4k tokens de contexto

Llama: Open and efficient foundation language models. (2023)
Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F. and Rodriguez, A.

LLaMA

Arquitectura transformer

$$\mathrm{SwiGLU}\left(x, W, V, b, c, \beta\right) = \mathrm{Swish}_{\beta}\left(xW + b\right) \otimes \left(xV + c\right)$$

- Pre-normalización (GPT3)
- Función de activación SwiGLU (PaLM)
- Rotary Position Embeddings (GPTNeo)
- Optimización AdamW
- Varias optimizaciones de eficiencia en memoria y velocidad

LLaMA

Datos:

- English CommonCrawl (67%)
- C4 (15%)
- Github (4.5%)
- Wikipedia (4.5%), varios idiomas
- Gutenberg, Books3 (4.5%)
- ArXiv (2.5%)
- Stack Exchange (2%)

Total aproximado: 4.5 TB

LLaMA

		0-shot	1-shot	5-shot	64-shot
GPT-3	175B	14.6	23.0	-	29.9
Gopher	280B	10.1	-	24.5	28.2
Chinchilla	70B	16.6	-	31.5	35.5
PaLM	8B	8.4	10.6	-	14.6
	62B	18.1	26.5	-	27.6
	540B	21.2	29.3	-	39.6
LLaMA	7B	16.8	18.7	22.0	26.1
	13B	20.1	23.4	28.1	31.9
	33B	**24.9**	28.3	32.9	36.0
	65B	23.8	**31.0**	**35.0**	**39.9**

Table 4: **NaturalQuestions.** Exact match performance.

		0-shot	1-shot	5-shot	64-shot
Gopher	280B	43.5	-	57.0	57.2
Chinchilla	70B	55.4	-	64.1	64.6
LLaMA	7B	50.0	53.4	56.3	57.6
	13B	56.6	60.5	63.1	64.0
	33B	65.1	67.9	69.9	70.4
	65B	**68.2**	**71.6**	**72.6**	**73.0**

Table 5: **TriviaQA.** Zero-shot and few-shot exact match performance on the filtered dev set.

	LLaMA	GPT3	OPT
Gender	70.6	**62.6**	65.7
Religion	79.0	73.3	**68.6**
Race/Color	**57.0**	64.7	68.6
Sexual orientation	81.0	**76.2**	78.6
Age	70.1	**64.4**	67.8
Nationality	64.2	**61.6**	62.9
Disability	**66.7**	76.7	76.7
Physical appearance	77.8	**74.6**	76.2
Socioeconomic status	**71.5**	73.8	76.2
Average	**66.6**	67.2	69.5

Table 12: **CrowS-Pairs.** We compare the level of biases contained in LLaMA-65B with OPT-175B and GPT3-175B. Higher score indicates higher bias.

Otros modelos

- OPT: *Open Pretrained Transformer*

Solo decoder. Varias versiones con los mismos parámetros que GPT.

- PaLM: *Pathways Language Model*

Solo decoder, 540B parámetros. En vez de multi-head attention hacen multi-query attention.

- UL2: *Unifying Language Learning Paradigms.*

Encoder-decoder o solo decoder. Representan el pretraining como diferentes variantes de denoising (como BART o T5).

Model	Foundation	Parameters (B)	MMLU	BBH	DROP	CRASS	HumanEval	Average
Human			89.8	94.4	94.1	98.2		
GPT-4			86.4		80.9		67.0	
Palm 2 Instruct			81.2	69.1	85.0			
ChatGPT			70.0	49.6	64.1	90.5	48.1	64.5
LLaMA	LLaMA	65	62.6	42.6	51.0	54.4	7.3	43.6
Alpaca Lora	LLaMA	65	61.7	45.7	51.0	68.6	23.2	50.0
GPT4 Alpaca Lora	LLaMA	30	58.4	41.3	45.1	79.2	18.9	48.6
OpenAssistant	LLaMA	30	56.9	39.2	46.0	67.2	23.1	46.5
LLaMA	LLaMA	30	57.8	39.3	45.4	68.6	14.0	45.0
GPT4 Alpaca Lora GPTQ						80.3		
Flan-UL2	UL2	20	55.0	44.7	64.3	94.2	0.0	51.6
Flan UL2 Alpaca Lora	UL2	20	45.7	39.2	54.3	91.2	0.0	46.1
Flan UL2 Dolly Lora	UL2	20	52.2	41.8	53.5	90.9	0.0	47.7
OPT IML	OPT	30	38.6	31.3	47.5	67.2	9.1	38.7
OPT	OPT	30	27.3	28.3	19.5	34.7	1.2	22.2
Flan-Alpaca	T5	11	50.9	23.3	62.3	90.2	0.0	45.3
Flan-T5-XXL	T5	11	54.5	43.9	67.2	88.3	0.0	50.8
TK-Instruct	T5	11	41.1	32.9	28.6	31.4	0.0	26.8
T0	T5	11	36.8	10.8	1.6	58.0	0.0	21.5
StableVicuna	LLaMA	13	49.2	37.5	34.3	67.5	15.9	40.9
Vicuna	LLaMA	13	49.7	37.1	32.9	60.9	15.2	39.2
GPT4 Alpaca Lora	LLaMA	13	46.4	36.6	35.4	61.0	14.0	38.7
LLaMA	LLaMA	13	46.2	37.1	35.3	58.8	13.4	38.2
Koala	LLaMA	13	44.6	34.6	28.3	52.6	11.0	34.2

https://declare-lab.net/instruct-eval/

Comparación

MMLU benchmark con preguntas de múltiple opción diseñadas para medir conocimiento en múltiples materias como elementary matemática, historia de Estados Unidos, computación, derecho y más.

BIG-Bench Hard (BBH) agregación de 23 tareas diseñadas para ir más allá de las capacidades de los modelos de lenguaje actuales. Además de tareas clásicas de PLN, tiene tareas para seguir instrucciones complejas como navegación, deducciones lógicas y detección de falacias.

Discrete Reasoning Over Paragraphs (DROP) tarea de comprensión lectora sobre problemas matemáticos, requiere razonamiento discreto, con textos obtenidos de Wikipedia.

Counterfactual Reasoning Assessment (CRASS) diseñado para testear habilidades de razonamiento causal.

HumanEval benchmark creado para evaluar modelos de lenguaje que hayan sido entrenados con código, como generar un programa a partir de un prompt en lenguaje natural.


Fine Tuning

Fine tuning

Los grandes modelos de lenguaje contienen miles de millones de parámetros

Están pre-entrenados con una gran cantidad de texto, por lo que decimos que ya tienen buen conocimiento del lenguaje

¿Cómo puedo adaptarlos a mi tarea específica?

- Zero-shot: Le pido que haga algo
- Few-shot: Le pido que haga algo y le doy un par de ejemplos
- Fine-tuning: Si tengo unos cuantos ejemplos más, puedo ajustar los pesos del modelo para mi tarea

Fine tuning

Ajustar los pesos en general va a tener mucha mejor performance

- Adaptar a una tarea nueva
- Adaptar a un idioma nuevo
- O a un dominio diferente

¿Problemas?

- Necesito tener los datos etiquetados
- Lleva mucho tiempo adaptar todos los pesos del modelo
- También ocupa mucha memoria

LoRA

Low Rank Adaptation

Es una técnica para ajustar modelos (fine-tuning) que permite entrenar solo unos pocos parámetros

Es mucho más rápido y eficiente

Sirve para ajustar cualquier tipo de red neuronal pre-entrenada, no solo modelos de lenguaje

Lora: Low-rank adaptation of large language models. (2021)
Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. and Chen, W.

Repaso de álgebra lineal

Rango de una matriz: cantidad de filas (o columnas) que son linealmente independientes

Para una matriz de tamaño $m*n$, el rango siempre es menor o igual que $min(m,n)$

Dadas dos matrices A con rango a y B con rango b, se cumple que $rango(AB) \leq min(a,b)$

Descomposición por rango: Dada una matriz $A \rightarrow m*n$ de rango r, existen matrices $C \rightarrow m*r$ y $F \rightarrow r*n$ tales que: $A = CF$

LoRA

¿Cómo funciona el método?

Supongamos que en nuestro modelo tenemos una capa con los pesos dados por una matriz $W \rightarrow n*k$

Si quisiéramos hacer un finetuning completo, al final obtendríamos una nueva matriz $W' \rightarrow n*k$

Le llamaremos ΔW a la diferencia entre la matriz inicial y la final luego del finetuning $\Delta W = W' - W$

LoRA asume que este ΔW es una matriz de muy bajo rango (en comparación con W), o sea: $rango(\Delta W) << min(n,k)$

LoRA

¿Por qué se asume que ΔW es de muy bajo rango?

Se está asumiendo que un gran modelo de lenguaje, entrenado sobre montones de texto, aprende en sus pesos features que permiten modelar todo el lenguaje

Cuando queremos ajustarlo a una tarea o a un dominio particular, “solo” tendríamos que aprender lo relevante para ese dominio

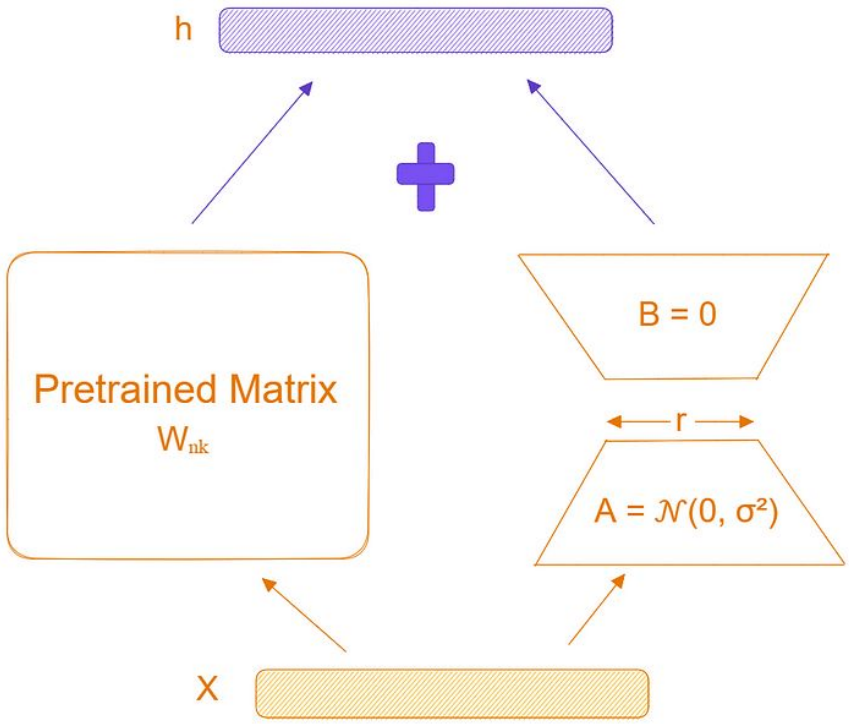
LoRA

Vamos a ajustar la matriz $W \rightarrow n*k$ con otra matriz ΔW de rango r

Entonces existen $A \rightarrow n*r$ y $B \rightarrow r*k$ tales que $\Delta W = AB$

Si elijo un valor de r mucho más chico que n y k, tengo que entrenar muchos menos parámetros

Usualmente se empieza con A inicializado con una gaussiana, y B matriz nula



LoRA

Ventajas:

- En vez de actualizar $n*k$ parámetros, actualizo $r*(n+k)$, que como r es más chico, va a ser mucho menor

- Una vez que terminé de entrenar, puedo sustituir W por $W+\Delta W$ y me queda un nuevo modelo ajustado, sin pérdida de eficiencia

- O en vez de eso, puedo tener el W original y varios ΔW para usarlos en distintas tareas, cambiándolos al momento de ejecución

LoRA

Model & Method	# Trainable Parameters	E2E NLG Challenge BLEU	NIST	MET	ROUGE-L	CIDEr
GPT-2 M (FT)*	354.92M	68.2	8.62	46.2	71.0	2.47
GPT-2 M ($\text{Adapter}^{\text{L}}$)*	0.37M	66.3	8.41	45.0	69.8	2.40
GPT-2 M ($\text{Adapter}^{\text{L}}$)*	11.09M	68.9	8.71	46.1	71.3	2.47
GPT-2 M ($\text{Adapter}^{\text{H}}$)	11.09M	$67.3_{\pm.6}$	$8.50_{\pm.07}$	$46.0_{\pm.2}$	$70.7_{\pm.2}$	$2.44_{\pm.01}$
GPT-2 M (FT^{Top2})*	25.19M	68.1	8.59	46.0	70.8	2.41
GPT-2 M (PreLayer)*	0.35M	69.7	8.81	46.1	71.4	2.49
GPT-2 M (LoRA)	0.35M	$\mathbf{70.4}_{\pm.1}$	$\mathbf{8.85}_{\pm.02}$	$\mathbf{46.8}_{\pm.2}$	$\mathbf{71.8}_{\pm.1}$	$\mathbf{2.53}_{\pm.02}$
GPT-2 L (FT)*	774.03M	68.5	8.78	46.0	69.9	2.45
GPT-2 L ($\text{Adapter}^{\text{L}}$)	0.88M	$69.1_{\pm.1}$	$8.68_{\pm.03}$	$46.3_{\pm.0}$	$71.4_{\pm.2}$	$\mathbf{2.49}_{\pm.0}$
GPT-2 L ($\text{Adapter}^{\text{L}}$)	23.00M	$68.9_{\pm.3}$	$8.70_{\pm.04}$	$46.1_{\pm.1}$	$71.3_{\pm.2}$	$2.45_{\pm.02}$
GPT-2 L (PreLayer)*	0.77M	70.3	8.85	46.2	71.7	2.47
GPT-2 L (LoRA)	0.77M	$\mathbf{70.4}_{\pm.1}$	$\mathbf{8.89}_{\pm.02}$	$\mathbf{46.8}_{\pm.2}$	$\mathbf{72.0}_{\pm.2}$	$2.47_{\pm.02}$

Table 3: GPT-2 medium (M) and large (L) with different adaptation methods on the E2E NLG Challenge. For all metrics, higher is better. LoRA outperforms several baselines with comparable or fewer trainable parameters. Confidence intervals are shown for experiments we ran. * indicates numbers published in prior works.

Model&Method	# Trainable Parameters	WikiSQL Acc. (%)	MNLI-m Acc. (%)	SAMSum R1/R2/RL
GPT-3 (FT)	175,255.8M	**73.8**	89.5	52.0/28.0/44.5
GPT-3 (BitFit)	14.2M	71.3	91.0	51.3/27.4/43.5
GPT-3 (PreEmbed)	3.2M	63.1	88.6	48.3/24.2/40.5
GPT-3 (PreLayer)	20.2M	70.1	89.5	50.8/27.3/43.5
GPT-3 ($\text{Adapter}^{\text{H}}$)	7.1M	71.9	89.8	53.0/28.9/44.8
GPT-3 ($\text{Adapter}^{\text{H}}$)	40.1M	73.2	**91.5**	53.2/29.0/45.1
GPT-3 (LoRA)	4.7M	73.4	**91.7**	**53.8/29.8/45.9**
GPT-3 (LoRA)	37.7M	**74.0**	**91.6**	53.4/29.2/45.1

Table 4: Performance of different adaptation methods on GPT-3 175B. We report the logical form validation accuracy on WikiSQL, validation accuracy on MultiNLI-matched, and Rouge-1/2/L on SAMSum. LoRA performs better than prior approaches, including full fine-tuning. The results on WikiSQL have a fluctuation around $\pm 0.5\%$, MNLI-m around $\pm 0.1\%$, and SAMSum around $\pm 0.2/\pm 0.2/\pm 0.1$ for the three metrics.

Referencias

- A Comparative Analysis of LLMs like BERT, BART, and T5. Zain ul Abideen

https://medium.com/@zaiinn440/a-comparative-analysis-of-llms-like-bert-bart-and-t5-a4a873251ff

- Low Rank Adaptation: A Technical Deep Dive. Nikhil Nagaraj

https://www.ml6.eu/blogpost/low-rank-adaptation-a-technical-deep-dive

- Papers…