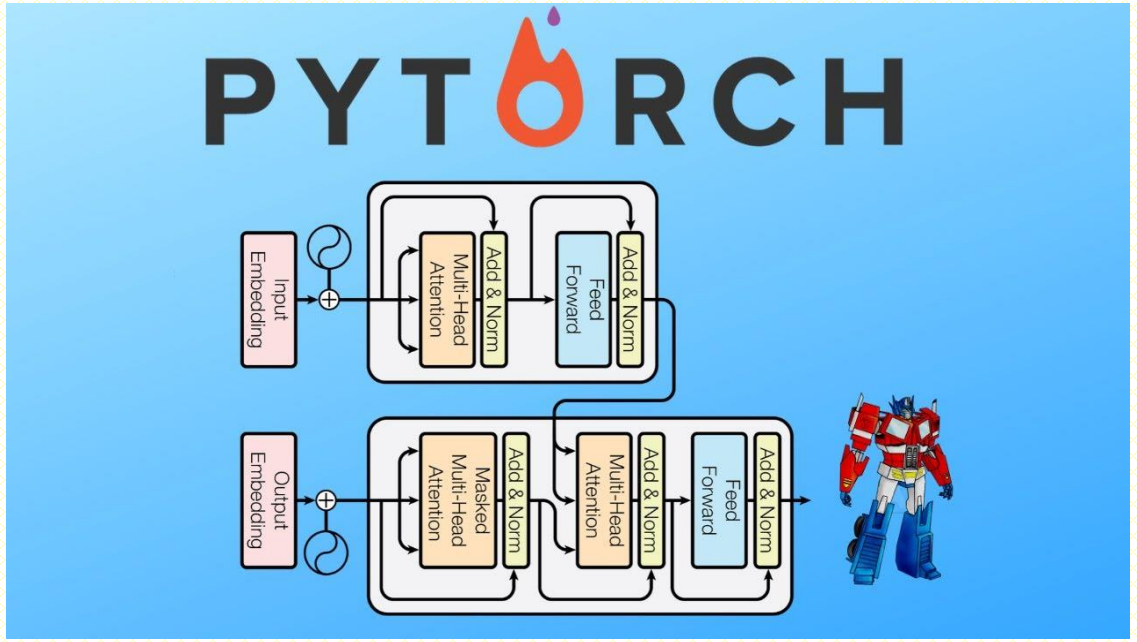
Redes de Transformers en Pytorch

Diego Andrade Canosa

Roberto López Castro

Índice

- Introducción
- Conceptos básicos de Transformers
- Transformers en Pytorch



citic

Introducción

- Los Transformers son un tipo de modelo de aprendizaje automático que ha revolucionado el procesamiento del lenguaje natural.
- En esta presentación, exploraremos los conceptos básicos de los Transformers y cómo se aplican en PyTorch.
- Los Transformers han demostrado un rendimiento destacado en una variedad de tareas, desde la traducción automática hasta la generación de texto y el análisis de sentimientos.
- A medida que avanzamos en la presentación, descubriremos cómo los Transformers en PyTorch han simplificado la implementación y el fine-tuning de estos modelos en aplicaciones del mundo real.

Objetivos de la presentación

- Comprender los conceptos fundamentales de los Transformers y su importancia en el procesamiento del lenguaje natural.
- Explorar los componentes clave de los Transformers y cómo se relacionan entre sí.
- Aprender cómo implementar y utilizar modelos de Transformers en PyTorch.
- Familiarizarse con la biblioteca Transformers de Hugging Face y su integración con PyTorch.
- Conocer el proceso de preprocesamiento de datos y ajuste fino de modelos de Transformers en PyTorch.
- Conocer las ventajas y aplicaciones de los Transformers en PyTorch.

¿Qué son los transformers?

- Modelos de aprendizaje automático muy utilizados en procesamiento del lenguaje natural.
- Introducidos en 2017 como alternativa a los modelos recurrentes.
- Utilizan el mecanismo de atención para capturar relaciones entre palabras.
- Rendimiento destacado en traducción automática, resumen de texto, generación de texto y análisis de sentimientos.
- Estructura codificador-decodificador.
- Estándar en muchas aplicaciones de procesamiento del lenguaje natural.
- En esta presentación, exploraremos los conceptos básicos y su implementación en PyTorch.

Atención (Attention)

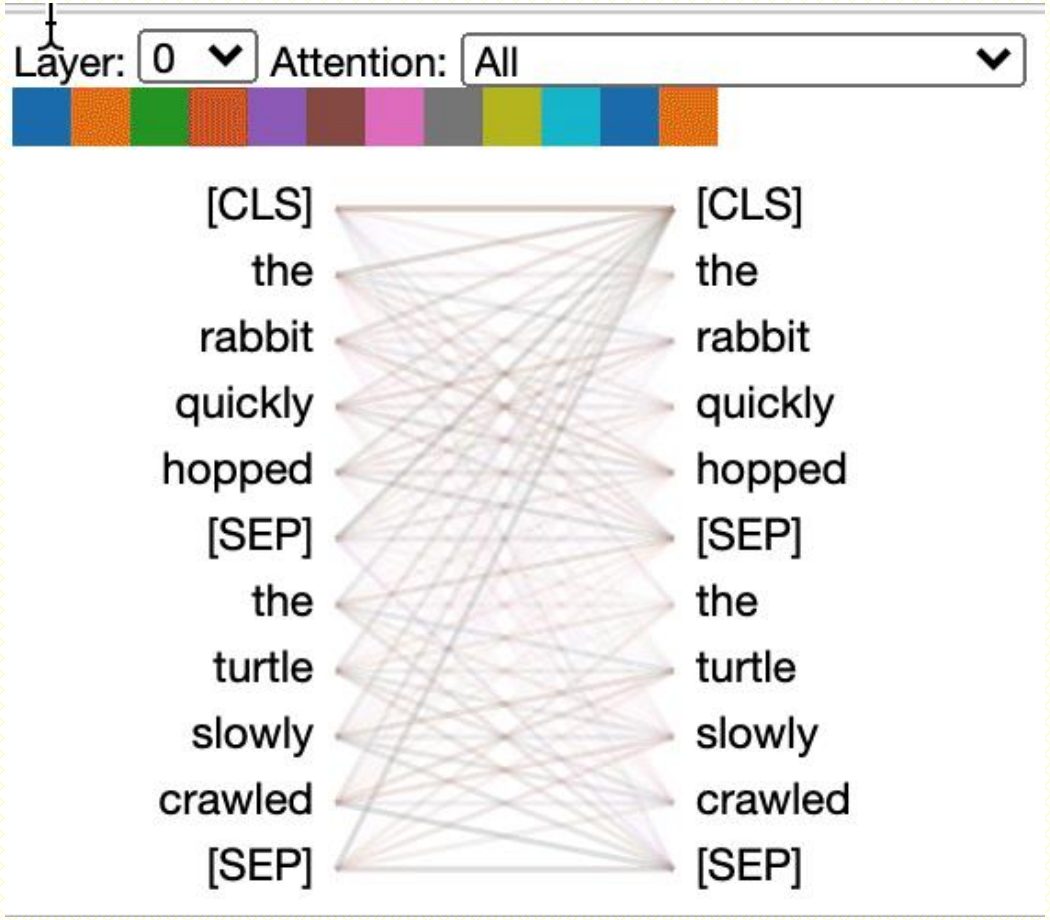
- La atención es un componente fundamental en los Transformers.
- Permite capturar las relaciones entre las palabras en una secuencia de entrada.
- A diferencia de los modelos secuenciales tradicionales, que procesan las palabras en orden, los Transformers calculan las relaciones entre todas las palabras simultáneamente
  - => alto grado de paralelismo vs. RNN
- La atención se basa en el concepto de consultas, claves y valores.
- Las consultas representan las palabras que queremos atender.
- Las claves y los valores representan las palabras con las que se comparan las consultas.

Atención (Attention)

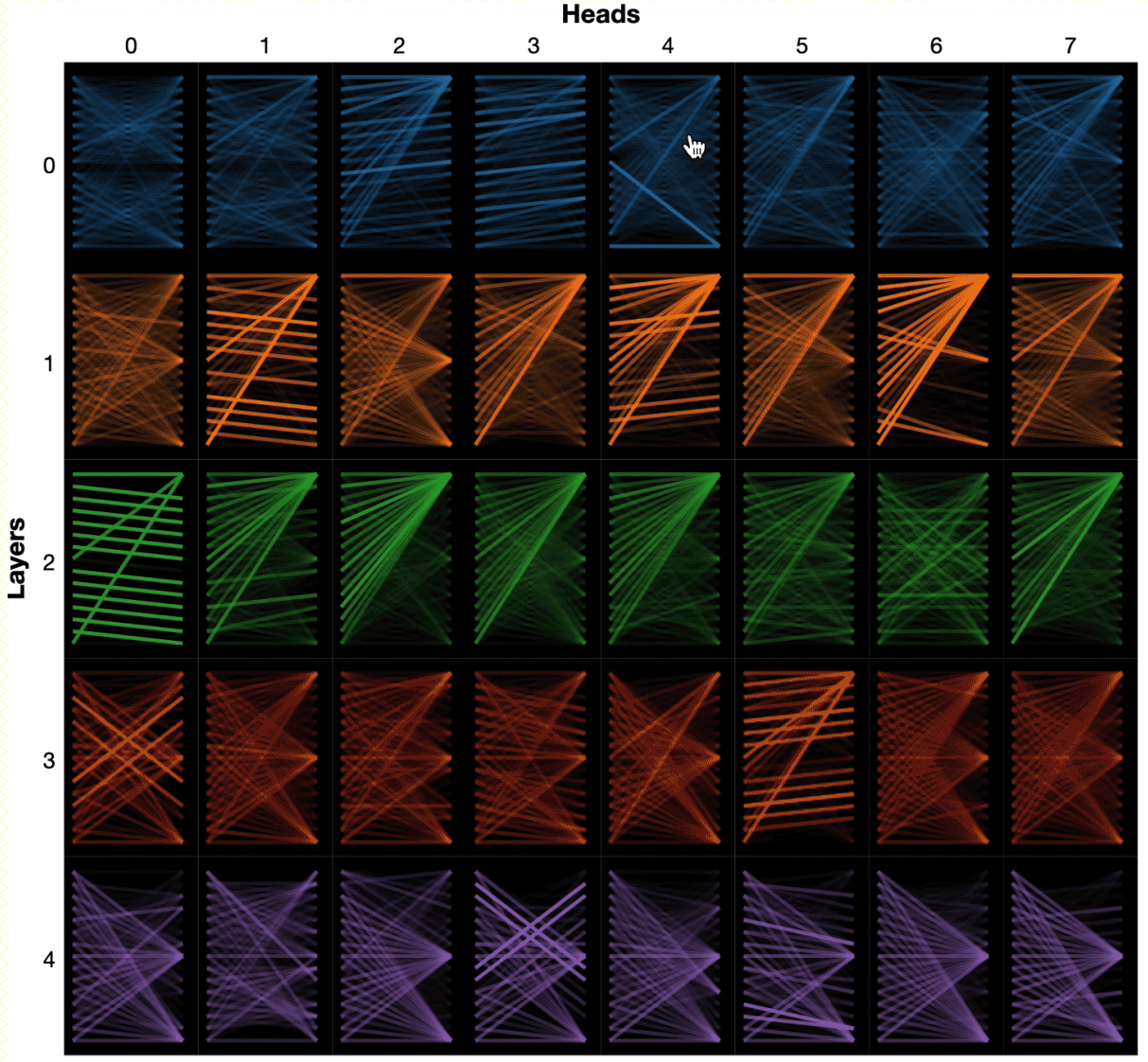
- La atención asigna pesos a las palabras clave según su relevancia para cada consulta.
- Los pesos se utilizan para ponderar los valores y calcular una representación contextualizada de cada palabra de entrada.
- Este enfoque permite que las palabras en una secuencia interactúen entre sí y se capturen las relaciones a largo plazo.
- En los Transformers, se utilizan múltiples cabezas de atención para capturar diferentes tipos de relaciones y mejorar el rendimiento.
- La atención es un componente esencial que impulsa el poder de los Transformers en el procesamiento del lenguaje natural.

VASWANI, Ashish, et al. Attention is all you need. *Advances in neural information processing systems*, 2017, vol. 30.

Atención (Attention)



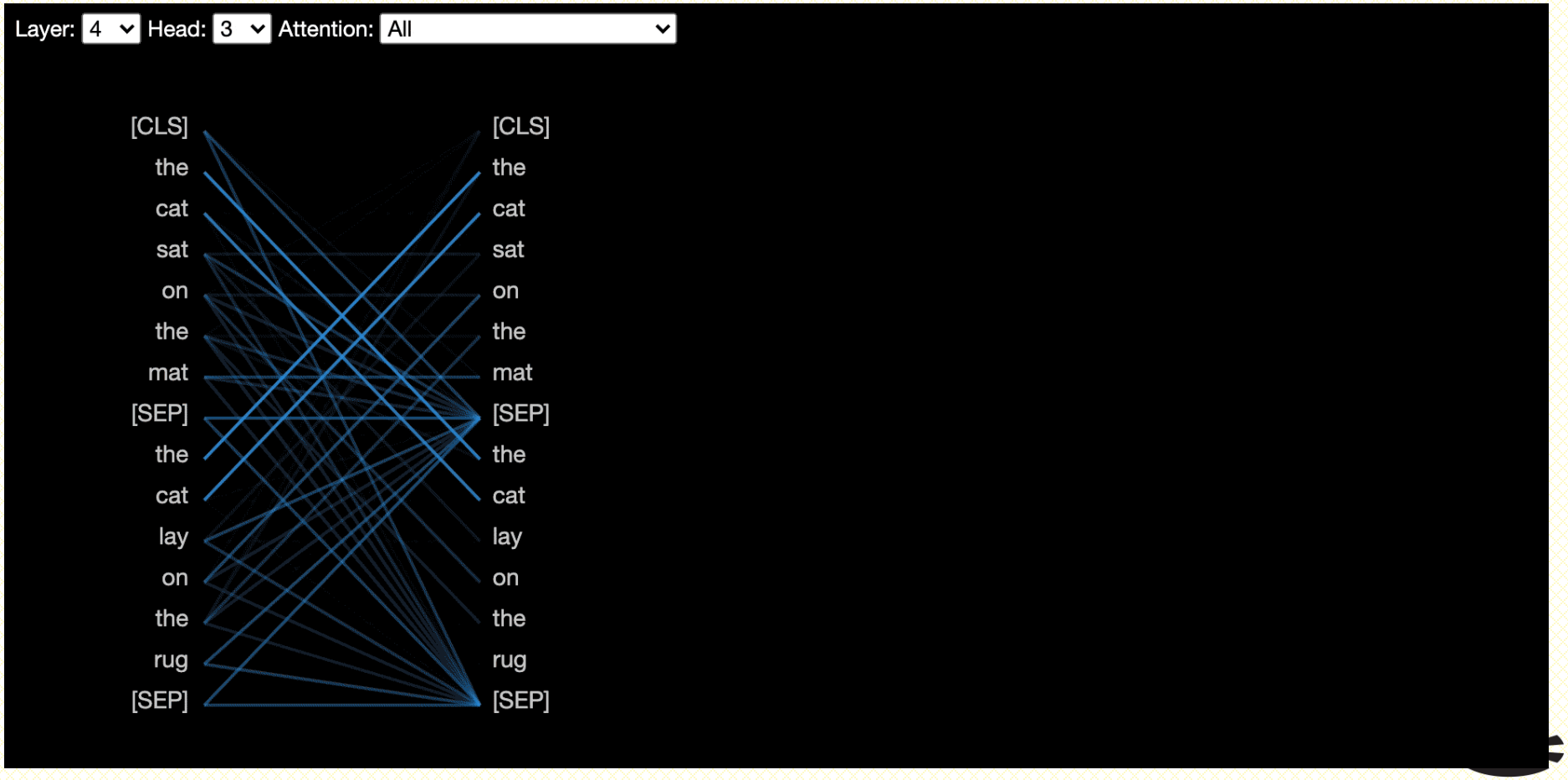
Atención (Attention)



citic

Atención (Attention)

https://github.com/jessevig/bertviz



citic

Componentes principales de los Transformers

- Los Transformers están compuestos por varios componentes clave que trabajan juntos para procesar secuencias de entrada. Algunos de los componentes principales son:
- Capa de atención (Attention Layer):
  - Calcula la atención entre todas las palabras en la secuencia.
  - Utiliza consultas, claves y valores para asignar pesos y obtener una representación contextualizada de cada palabra

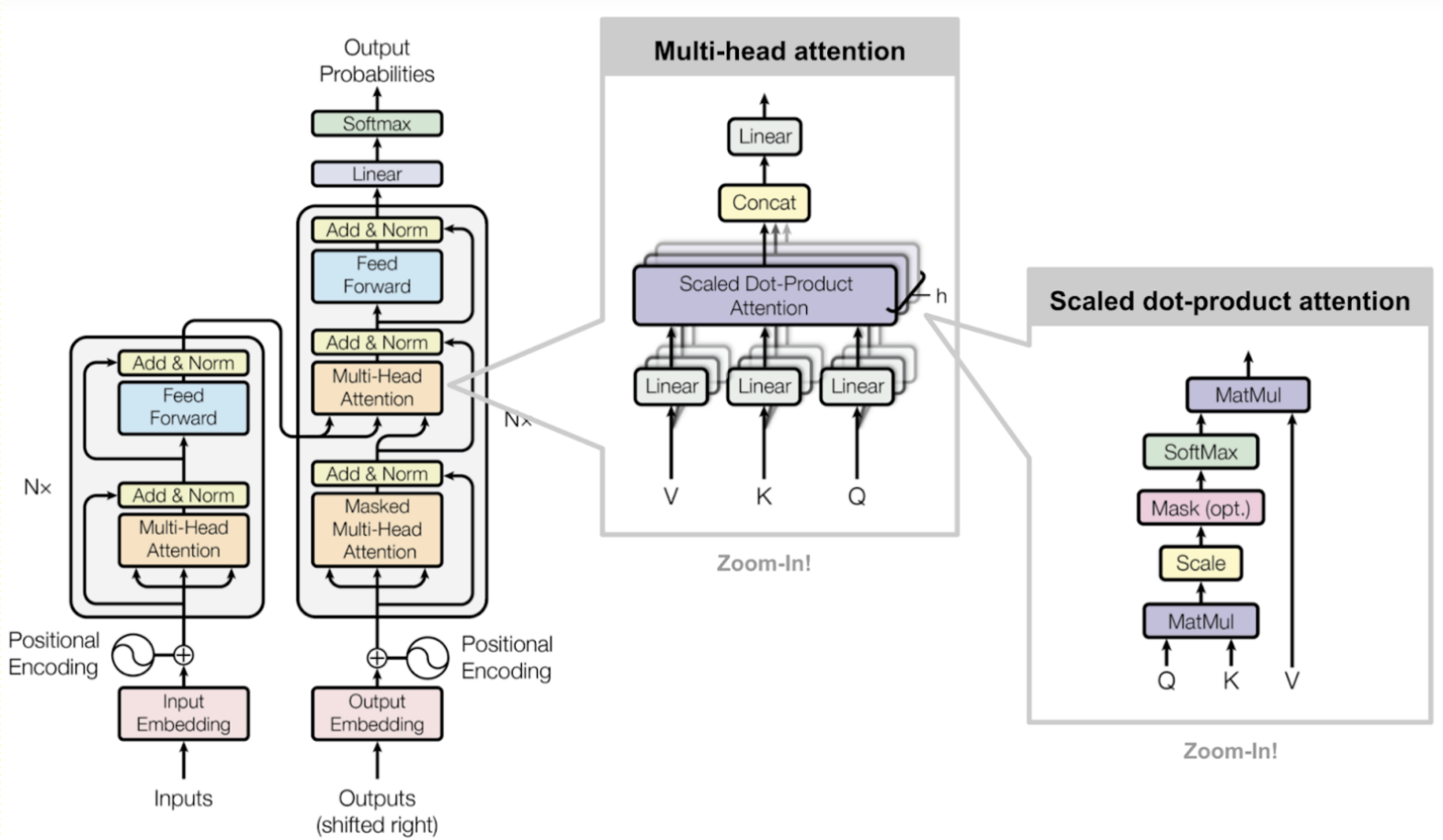
Componentes principales de los Transformers

- Capas de alimentación hacia adelante (Feed-forward Layers):
  - Proporcionan una transformación no lineal después de la capa de atención.
  - Ayudan a capturar relaciones más complejas y a modelar mejor las interacciones entre las palabras.
- Normalización de capas (Layer Normalization):
  - Se aplica después de cada capa para normalizar la salida.
  - Ayuda a estabilizar y acelerar el entrenamiento de los modelos de Transformers.

Componentes principales de los Transformers

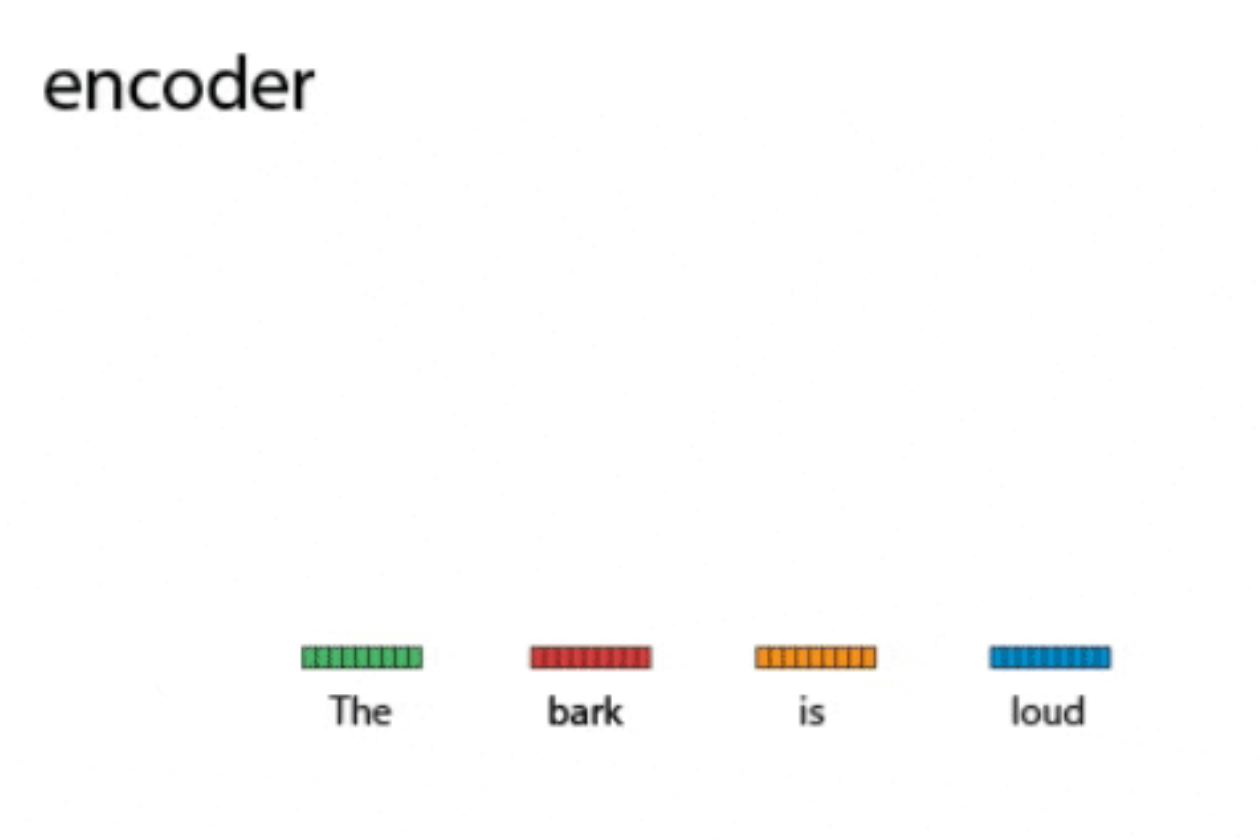
- Conexiones residuales (Residual Connections):
  - Conexiones que se agregan a lo largo de las capas del modelo.
  - Permiten que la información fluya directamente a través del modelo, evitando la pérdida de información.
- Codificador y decodificador:
  - Estructura típica de un modelo de Transformer.
  - El codificador procesa la secuencia de entrada y captura su representación contextualizada.
  - El decodificador genera la secuencia de salida basada en la representación contextual del codificador.
  - Estos componentes se combinan para formar una arquitectura poderosa que captura relaciones a largo plazo y permite el procesamiento eficiente de secuencias en los Transformers.

Componentes principales de los Transformers



citic

Componentes principales de los Transformers



Codificador y decodificador

- Los Transformers se componen de un codificador y un decodificador, que trabajan en conjunto para procesar secuencias y generar resultados.
- El codificador procesa la secuencia de entrada y captura su representación contextualizada.
- Consiste en varias capas de atención y alimentación hacia adelante.
- Cada capa de atención en el codificador se encarga de calcular las relaciones entre todas las palabras en la secuencia de entrada.

Codificador y decodificador

- La salida del codificador es una representación contextualizada de la secuencia de entrada, que captura tanto información local como global.
- El decodificador, por otro lado, genera la secuencia de salida basada en la representación contextual del codificador.
- También consta de múltiples capas de atención y alimentación hacia adelante, pero con algunas diferencias importantes.

Codificador y decodificador

- En el decodificador, se agrega una atención adicional, llamada "atención de máscara", que asegura que cada posición solo pueda atender a posiciones anteriores en la secuencia de salida.
- Esto evita que el modelo tenga acceso a información futura durante la generación de la secuencia.
- El decodificador también tiene una atención adicional, llamada "atención cruzada", que permite que cada posición se atienda a las salidas del codificador, capturando las relaciones entre la entrada y la salida.
- En conjunto, el codificador y el decodificador permiten que los Transformers capturen y generen secuencias de manera efectiva en aplicaciones como la traducción automática y la generación de texto.

Componentes principales de los Transformers

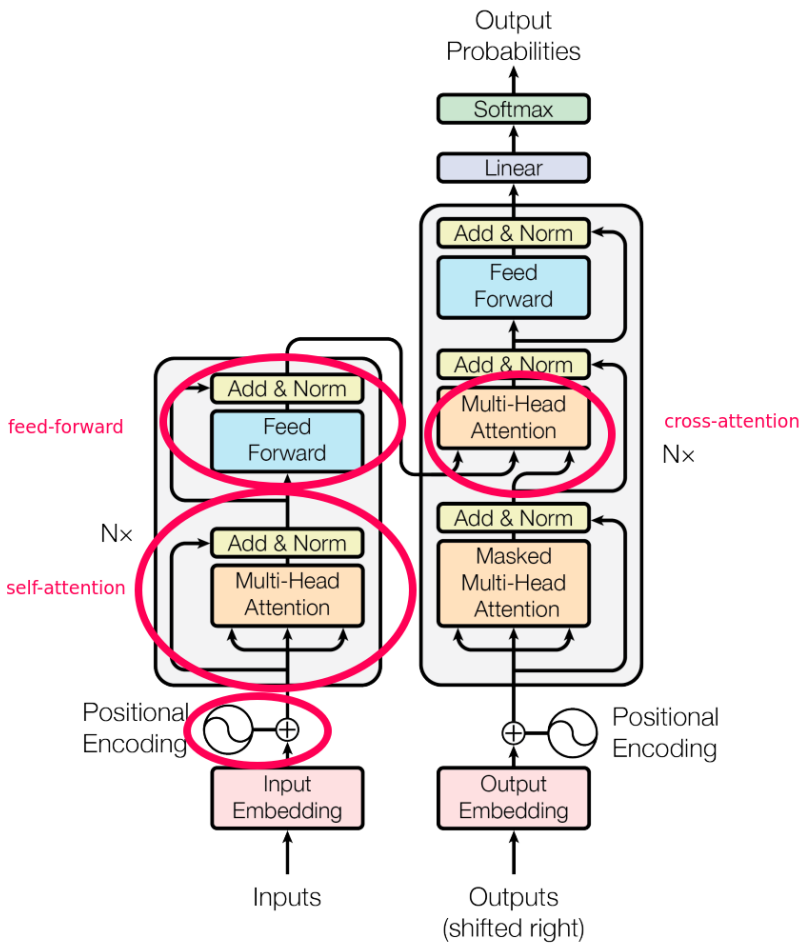


Figure 1: The Transformer - model architecture.

citic

Mecanismos de auto-atención (Self-Attention)

- Un componente fundamental en los Transformers es el mecanismo de auto-atención (self-attention).
- La auto-atención permite que un modelo se relacione con diferentes partes de la misma secuencia para capturar las dependencias y las relaciones contextuales.
- A diferencia de los modelos secuenciales tradicionales, que procesan las palabras en orden, la auto-atención calcula las relaciones entre todas las palabras simultáneamente.
- En la auto-atención, cada palabra en una secuencia tiene tres representaciones: consulta (query), clave (key) y valor (value).

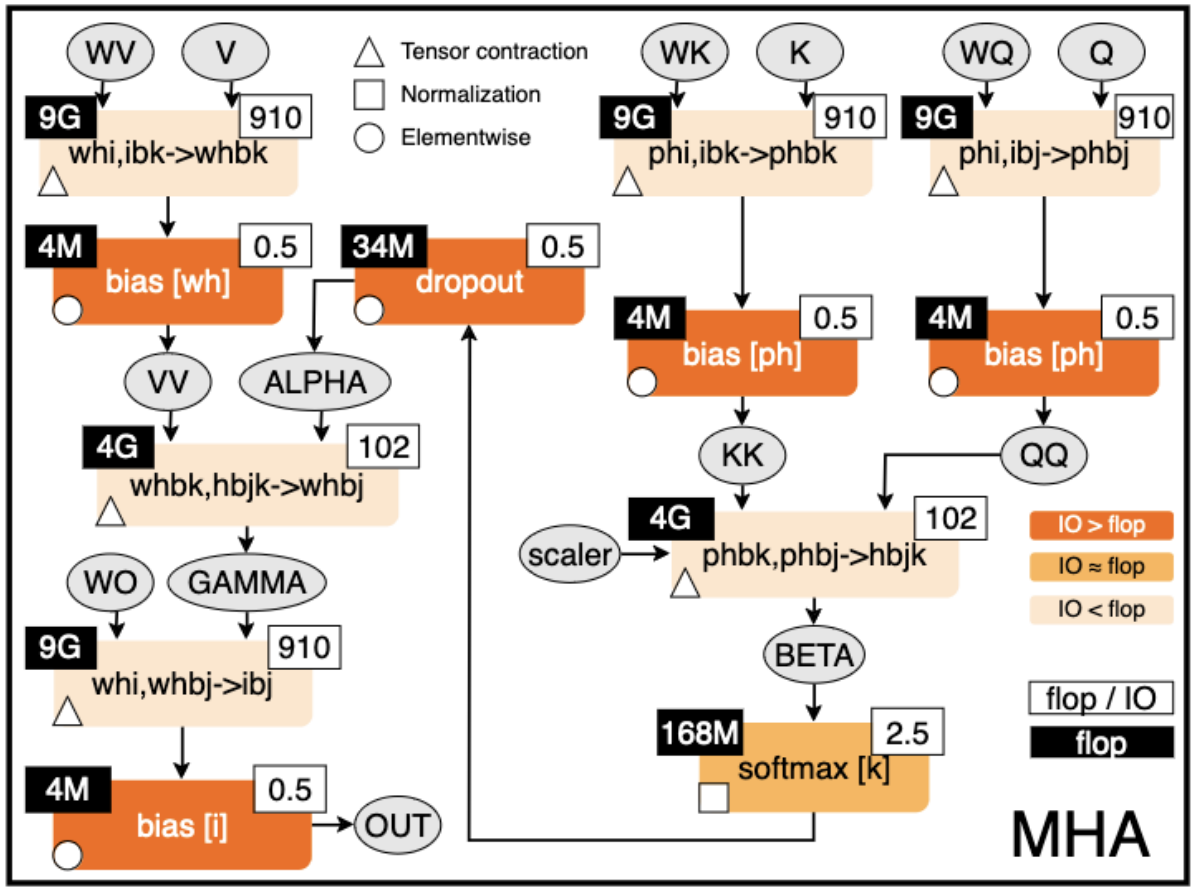
Mecanismos de auto-atención (Self-Attention)

- Las consultas representan las palabras que queremos atender, mientras que las claves y los valores representan las palabras con las que se comparan las consultas.
- La auto-atención calcula la relevancia entre todas las palabras en la secuencia al calcular productos escalares entre las consultas, las claves y los valores.
- Estos productos escalares se transforman en pesos mediante una función softmax, lo que proporciona una distribución de pesos que muestra la importancia relativa de cada palabra para cada consulta.

Mecanismos de auto-atención (Self-Attention)

- Los pesos se utilizan para ponderar los valores y calcular una representación contextualizada de cada palabra de entrada.
- La auto-atención se aplica en paralelo a todas las palabras de entrada, lo que permite que las palabras interactúen y capturen relaciones a largo plazo.
- La capacidad de capturar relaciones complejas y de largo alcance es una de las razones principales por las cuales los Transformers han sido tan efectivos en tareas de procesamiento del lenguaje natural.
- En resumen, la auto-atención es un componente clave que impulsa la capacidad de los Transformers para capturar relaciones contextuales y modelar la dependencia entre las palabras en una secuencia.

Mecanismos de auto-atención (Self-Attention)



(b) Resulting dataflow

citic

Funcionamiento de la atención

- La atención es un mecanismo central en los Transformers que permite capturar las relaciones entre las palabras en una secuencia.
- El funcionamiento de la atención se basa en el cálculo de productos escalares entre las consultas, claves y valores de cada palabra en la secuencia.
- A continuación, se describe el proceso paso a paso:
- Generación de consultas, claves y valores:
  - Cada palabra en la secuencia se utiliza para generar tres representaciones: consulta, clave y valor.
  - Estas representaciones se obtienen a través de transformaciones lineales de la palabra de entrada.
- Cálculo de similitudes:
  - Se calcula el producto escalar entre cada consulta y todas las claves en la secuencia.
  - El resultado de este cálculo refleja la similitud entre la consulta y cada una de las claves.
- Obtención de pesos de atención:
  - Los productos escalares se escalan y se pasan por una función softmax.
  - Esto produce una distribución de pesos que indica la importancia relativa de cada palabra (valor) en relación con la consulta.

Funcionamiento de la atención

- Cálculo de la representación contextualizada:
  - Los pesos de atención se utilizan para ponderar los valores correspondientes.
  - Se realiza una suma ponderada de los valores para obtener una representación contextualizada de la palabra de entrada.
- Repetición para cada palabra de la secuencia:
  - El proceso se repite para cada palabra en la secuencia, generando así una representación contextualizada para cada palabra.
- La atención se puede aplicar en paralelo a todas las palabras de entrada, lo que permite capturar las relaciones entre todas las palabras en la secuencia simultáneamente.
- Este enfoque de atención global es una de las características clave de los Transformers y les permite capturar relaciones a largo plazo de manera efectiva.
- En resumen, la atención en los Transformers se basa en el cálculo de similitudes entre consultas y claves, y utiliza los valores ponderados para generar una representación contextualizada de cada palabra en la secuencia. Este proceso permite que el modelo capture las dependencias y relaciones entre las palabras en un contexto global.

Tokenizer

- En Transformers, un tokenizer es una parte fundamental que se utiliza para convertir texto en secuencias de tokens comprensibles por los modelos de lenguaje preentrenados
- Un tokenizer divide el texto en unidades más pequeñas, como palabras o subpalabras, y las asigna a identificadores numéricos llamados tokens.

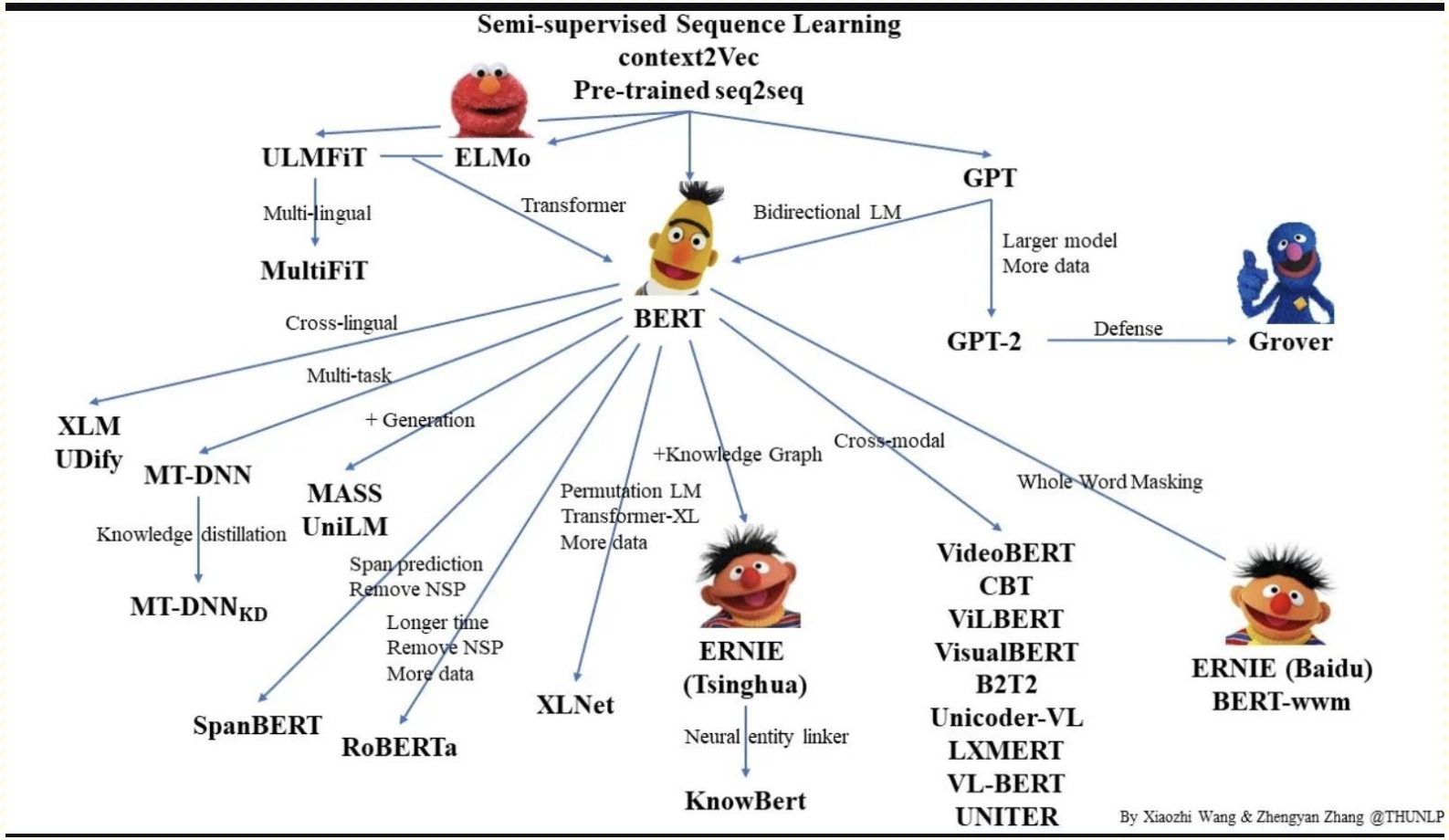
- Además de dividir el texto en tokens, los tokenizers también realizan otras tareas importantes, como agregar caracteres especiales al inicio y final de la secuencia de tokens, realizar el mapeo entre tokens y sus identificadores numéricos correspondientes, y manejar la codificación y decodificación de texto.

Tokenizer

- Los tokenizers en Transformers son específicos del modelo de lenguaje preentrenado que se está utilizando, ya que cada modelo puede tener su propio vocabulario y reglas de tokenización. Los tokenizers se construyen utilizando la biblioteca Tokenizers de Hugging Face, que proporciona una amplia gama de tokenizers preentrenados para diferentes modelos y tareas de procesamiento del lenguaje natural.

- Lab 1: 01_tokenizer_training.ipynb

Transformers



citic

Transformers en PyTorch

- Ofrece una interfaz flexible y herramientas eficientes.
- La biblioteca Transformers de Hugging Face es ampliamente utilizada en PyTorch.
- Proporciona modelos pre-entrenados y herramientas de procesamiento del lenguaje natural.
- Permite implementar modelos personalizados y realizar ajuste fino.
- PyTorch y Transformers simplifican la implementación de modelos de Transformers en procesamiento del lenguaje natural.

- pip install transformers

Transformers en PyTorch

- - El preprocesamiento de datos es una etapa crucial antes de utilizar los Transformers en PyTorch.
- - El objetivo del preprocesamiento es preparar los datos de entrada de manera adecuada para su uso en un modelo de Transformer.
- - Algunas tareas comunes de preprocesamiento incluyen:

- 1. Tokenización:
- - Dividir el texto en unidades más pequeñas llamadas "tokens".
- - Puede ser a nivel de palabras o subpalabras.

- 2. Codificación numérica:
- - Asignar un identificador numérico a cada token.
- - Representar el texto como secuencias de IDs de tokens.

Transformers en PyTorch

- 3. Alineación de secuencias:
- - Asegurar que todas las secuencias tengan la misma longitud.
- - Se pueden truncar o rellenar las secuencias según sea necesario.

- 4. Máscaras de atención:
- - Indicar qué tokens son reales y cuáles son tokens de relleno.
- - Ayudar al modelo a enfocarse en los tokens relevantes durante el procesamiento.

- - La biblioteca Transformers de Hugging Face proporciona herramientas para realizar estas tareas de preprocesamiento de manera eficiente en PyTorch.
- - Un adecuado preprocesamiento de datos es fundamental para garantizar un rendimiento óptimo del modelo Transformer y obtener resultados precisos en tareas de procesamiento del lenguaje natural.

Transformers en PyTorch

- Lab 1: 04_tokenizer_training.ipynb
- Lab 2: 04_languaje_modeling_from_scratch.ipynb

Carga de un modelo pre-entrenado

- La biblioteca Transformers de Hugging Face proporciona una amplia gama de modelos pre-entrenados que se pueden utilizar directamente.

1. Importar las bibliotecas necesarias:

```
from transformers import AutoModel, AutoTokenizer
```

2. Seleccionar el modelo pre-entrenado:
   - Elige el modelo pre-entrenado adecuado para tu tarea específica.
   - Puedes encontrar una lista de modelos pre-entrenados en la documentación de la biblioteca Transformers de Hugging Face.
   - https://huggingface.co/transformers/v3.3.1/pretrained_models.html

Carga de un modelo pre-entrenado

3. Cargar el tokenizer:

```
tokenizer
  =AutoTokenizer.from_pretrained('nombre_del_modelo')
```

4. Cargar el modelo:

```
model = AutoModel.from_pretrained('nombre_del_modelo')
```

Fine-tuning (Ajuste fino)

Lab 3: 04_question_answering.ipynb