Aprendizaje Topológico 2025-II

Spatioformer: A Geo-encoded Transformer

Miguel Ángel Maurin García de la Vega

12 de febrero de 2025

Instituto de Matemáticas UNAM

Contenidos

Overview del Modelo Transformer

El modelo Transformer fue originalmente ideado para tareas de procesamiento de lenguaje natural. Tomaremos como motivación la tarea de generación de texto del modelo GPT. La idea de este modelo es recibir una secuencia de palabras y predecir la palabra siguiente.

Overview del Modelo Transformer

El modelo Transformer fue originalmente ideado para tareas de procesamiento de lenguaje natural. Tomaremos como motivación la tarea de generación de texto del modelo GPT. La idea de este modelo es recibir una secuencia de palabras y predecir la palabra siguiente. Comenzaremos con esta versión para exponer los componentes clave del modelo: **Embedding, Attention, MLP y Unembedding**.

Overview del Modelo Transformer

- El input se divide en unidades llamadas **tokens**, que podemos pensar como palabras.

Overview del Modelo Transformer

- El input se divide en unidades llamadas **tokens**, que podemos pensar como palabras.
- Cada token es asociado con un **vector** en un espacio de cierta dimensión, el cual codifica su significado.

Overview del Modelo Transformer

- El input se divide en unidades llamadas **tokens**, que podemos pensar como palabras.
- Cada token es asociado con un **vector** en un espacio de cierta dimensión, el cual codifica su significado.
- Estos vectores pasan por el mecanismo de **atención**, donde procesan el contexto de las palabras cercanas.

Overview del Modelo Transformer

- El input se divide en unidades llamadas **tokens**, que podemos pensar como palabras.
- Cada token es asociado con un **vector** en un espacio de cierta dimensión, el cual codifica su significado.
- Estos vectores pasan por el mecanismo de **atención**, donde procesan el contexto de las palabras cercanas.
- Luego, los vectores se procesan en paralelo a través de una **MLP** (red neuronal multicapa).

Overview del Modelo Transformer

- El input se divide en unidades llamadas **tokens**, que podemos pensar como palabras.
- Cada token es asociado con un **vector** en un espacio de cierta dimensión, el cual codifica su significado.
- Estos vectores pasan por el mecanismo de **atención**, donde procesan el contexto de las palabras cercanas.
- Luego, los vectores se procesan en paralelo a través de una **MLP** (red neuronal multicapa).
- Tras repetir estos bloques varias veces, obtenemos un vector final que genera una **distribución de probabilidad** sobre la siguiente palabra.

Cada uno de estos pasos corresponde a una operación matricial. Nuestro objetivo es entender el rol de cada una de estas operaciones.

Word Embeddings

El primer paso en el modelo Transformer es mapear todas las **palabras únicas** (nuestro vocabulario) a un espacio de mayor dimensión.

Word Embeddings

El primer paso en el modelo Transformer es mapear todas las **palabras únicas** (nuestro vocabulario) a un espacio de mayor dimensión.
Este mapeo se realiza a través de una matriz denominada **Embedding Matrix**, donde cada palabra del vocabulario está representada por un vector en un espacio de dimensión fija.

Word Embeddings

El primer paso en el modelo Transformer es mapear todas las **palabras únicas** (nuestro vocabulario) a un espacio de mayor dimensión.
Este mapeo se realiza a través de una matriz denominada **Embedding Matrix**, donde cada palabra del vocabulario está representada por un vector en un espacio de dimensión fija.

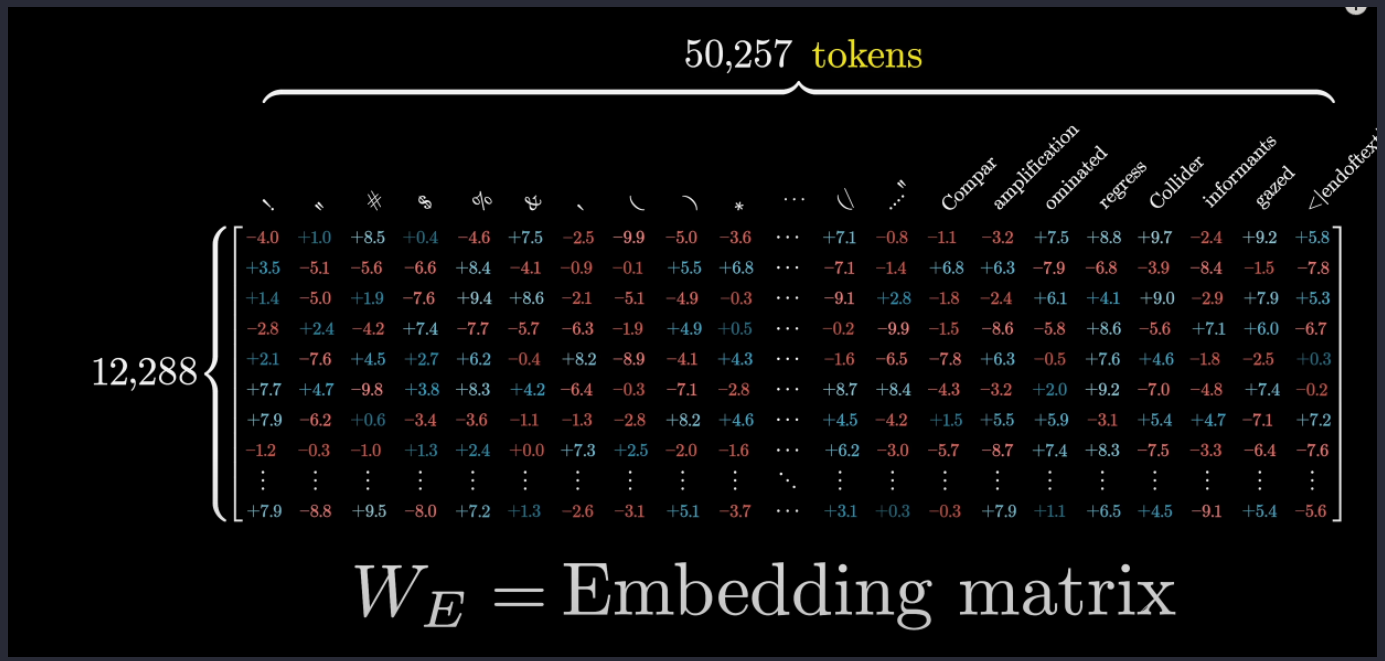
- La **Embedding Matrix** es una matriz de tamaño $d \times V$, donde:
  - V es el tamaño del vocabulario.
  - d es la dimensión del espacio de embedding.

Word Embeddings

El primer paso en el modelo Transformer es mapear todas las **palabras únicas** (nuestro vocabulario) a un espacio de mayor dimensión.
Este mapeo se realiza a través de una matriz denominada **Embedding Matrix**, donde cada palabra del vocabulario está representada por un vector en un espacio de dimensión fija.

- La **Embedding Matrix** es una matriz de tamaño $d \times V$, donde:
  - V es el tamaño del vocabulario.
  - d es la dimensión del espacio de embedding.
- Cada columna de la matriz representa una palabra única en este nuevo espacio vectorial.

Word Embeddings

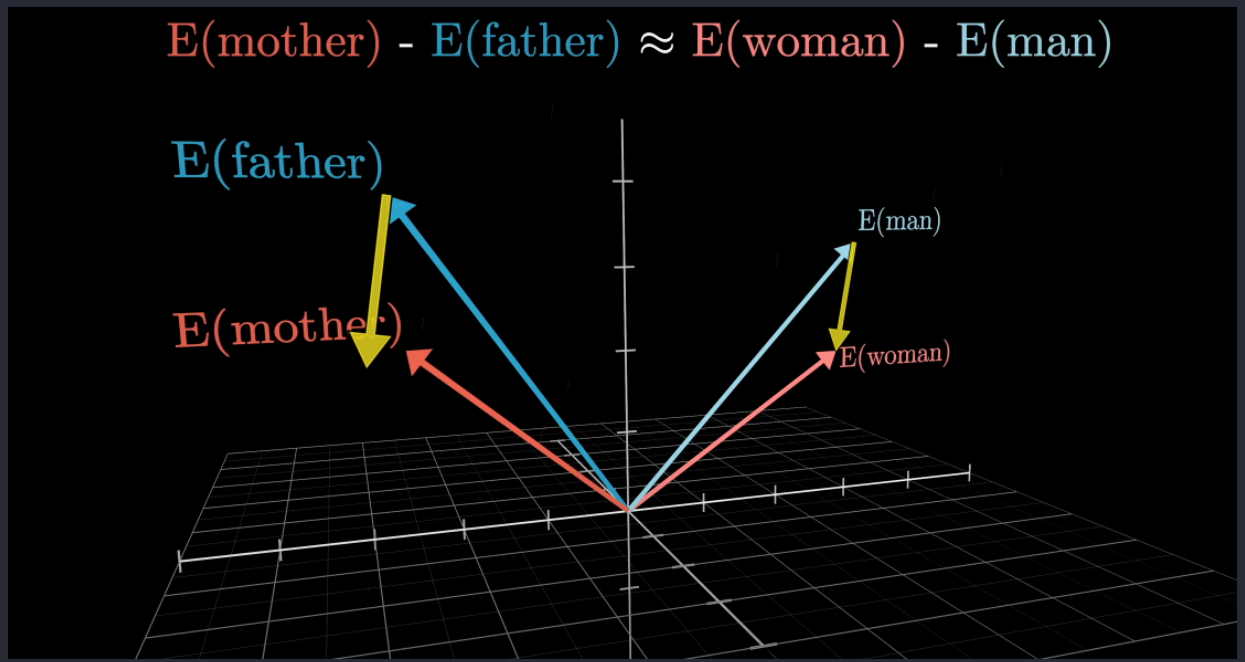


Word Embeddings

Una vez entrenado, las direcciones de los encajes codifican un componente semantico, por ejemplo:

Word Embeddings

Una vez entrenado, las direcciones de los encajes codifican un componente semantico, por ejemplo:



Unembedding Matrix

Después de aplicar las operaciones clave del modelo (**Atención, MLP**), obtenemos una matriz con representaciones actualizadas de los tokens procesados.

Unembedding Matrix

Después de aplicar las operaciones clave del modelo (**Atención, MLP**), obtenemos una matriz con representaciones actualizadas de los tokens procesados.
Nos enfocamos en el último vector, el cual representa la última palabra con su contexto actualizado. Para predecir la siguiente palabra, aplicamos una nueva transformación mediante la matriz de **unembedding**.

Unembedding Matrix

Después de aplicar las operaciones clave del modelo (**Atención, MLP**), obtenemos una matriz con representaciones actualizadas de los tokens procesados.
Nos enfocamos en el último vector, el cual representa la última palabra con su contexto actualizado. Para predecir la siguiente palabra, aplicamos una nueva transformación mediante la matriz de **unembedding**.

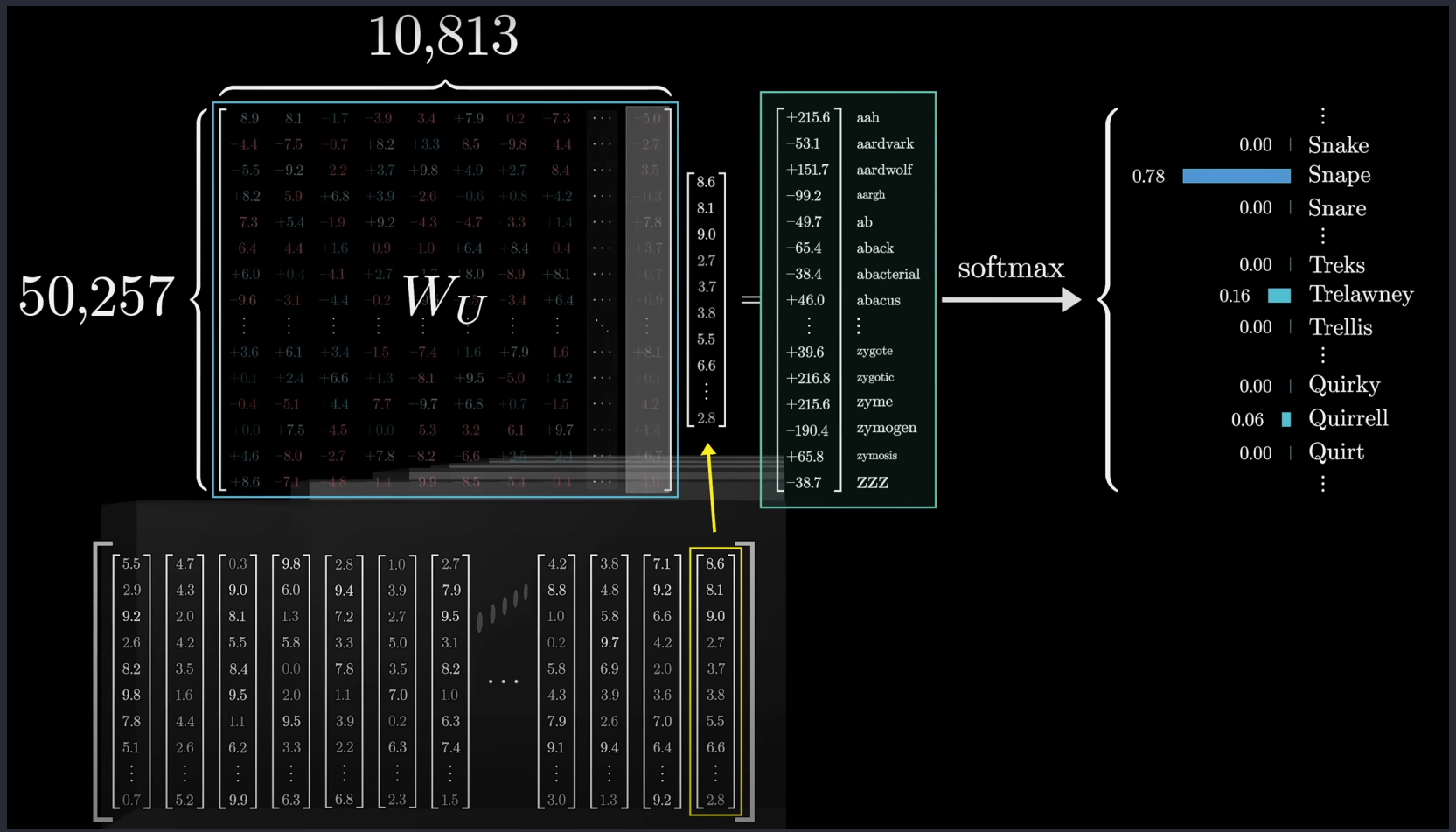
- La **Unembedding Matrix** proyecta este último vector en un nuevo espacio.

Unembedding Matrix

Después de aplicar las operaciones clave del modelo (**Atención, MLP**), obtenemos una matriz con representaciones actualizadas de los tokens procesados.
Nos enfocamos en el último vector, el cual representa la última palabra con su contexto actualizado. Para predecir la siguiente palabra, aplicamos una nueva transformación mediante la matriz de **unembedding**.

- La **Unembedding Matrix** proyecta este último vector en un nuevo espacio.
- Este nuevo espacio representa todas las palabras del vocabulario, asignando una **distribución de probabilidad** sobre ellas. Para esto, usamos la función softmax.

Unembedding Matrix



Motivación del Mecanismo de Atención

El **mecanismo de atención** permite que el modelo actualice los **encajes** (significados) de las palabras según el contexto en el que aparecen.

Motivación del Mecanismo de Atención

El **mecanismo de atención** permite que el modelo actualice los **encajes** (significados) de las palabras según el contexto en el que aparecen.
Ejemplo: Supongamos que la palabra *"banco"* está en nuestro vocabulario.

Motivación del Mecanismo de Atención

El **mecanismo de atención** permite que el modelo actualice los **encajes** (significados) de las palabras según el contexto en el que aparecen.
Ejemplo: Supongamos que la palabra *"banco"* está en nuestro vocabulario.
Un modelo correctamente entrenado, usando el contexto, podrá capturar sus distintos significados:

- **"el banco de sardinas se mueve hacia el norte"** (grupo de peces).

Motivación del Mecanismo de Atención

El **mecanismo de atención** permite que el modelo actualice los **encajes** (significados) de las palabras según el contexto en el que aparecen.
Ejemplo: Supongamos que la palabra *"banco"* está en nuestro vocabulario.
Un modelo correctamente entrenado, usando el contexto, podrá capturar sus distintos significados:

- **"el banco de sardinas se mueve hacia el norte"** (grupo de peces).
- **"el joven se sentó en el banco"** (mobiliario).

Motivación del Mecanismo de Atención

El **mecanismo de atención** permite que el modelo actualice los **encajes** (significados) de las palabras según el contexto en el que aparecen.
Ejemplo: Supongamos que la palabra *"banco"* está en nuestro vocabulario.
Un modelo correctamente entrenado, usando el contexto, podrá capturar sus distintos significados:

- **"el banco de sardinas se mueve hacia el norte"** (grupo de peces).
- **"el joven se sentó en el banco"** (mobiliario).
- **"hizo un deposito al banco"** (institución financiera).

Motivación del Mecanismo de Atención

El **mecanismo de atención** permite que el modelo actualice los **encajes** (significados) de las palabras según el contexto en el que aparecen.
Ejemplo: Supongamos que la palabra *"banco"* está en nuestro vocabulario.
Un modelo correctamente entrenado, usando el contexto, podrá capturar sus distintos significados:

- **"el banco de sardinas se mueve hacia el norte"** (grupo de peces).
- **"el joven se sentó en el banco"** (mobiliario).
- **"hizo un deposito al banco"** (institución financiera).

En el contexto de **predecir la siguiente palabra**, el mecanismo de atención permite que la predicción tome en cuenta todas las palabras previas, ajustando el significado de cada token según su contexto.

Cabezal de Atención Único

Supongamos que tenemos nuestra **matriz de encaje** y recibimos una nueva oración.

Cabezal de Atención Único

Supongamos que tenemos nuestra **matriz de encaje** y recibimos una nueva oración.
¿Cómo incorporamos el contexto para actualizar los encajes?

Cabezal de Atención Único

Supongamos que tenemos nuestra **matriz de encaje** y recibimos una nueva oración.
¿Cómo incorporamos el contexto para actualizar los encajes?
Un ejemplo ilustrativo es el efecto de **adjetivos** modificando **sustantivos**. (No representa realmente la complejidad de lo que ocurre en la capa pero sirve como intuición)

Cabezal de Atención Único

Supongamos que tenemos nuestra **matriz de encaje** y recibimos una nueva oración.
¿Cómo incorporamos el contexto para actualizar los encajes?
Un ejemplo ilustrativo es el efecto de **adjetivos** modificando **sustantivos**. (No representa realmente la complejidad de lo que ocurre en la capa pero sirve como intuición)
Ejemplo: Consideremos la oración:

"El zorro café saltó sobre el perro flojo."

Cabezal de Atención Único

Supongamos que tenemos nuestra **matriz de encaje** y recibimos una nueva oración.
¿Cómo incorporamos el contexto para actualizar los encajes?
Un ejemplo ilustrativo es el efecto de **adjetivos** modificando **sustantivos**. (No representa realmente la complejidad de lo que ocurre en la capa pero sirve como intuición)
Ejemplo: Consideremos la oración:

"El zorro café saltó sobre el perro flojo."

Queremos que las palabras **"café"** y **"flojo"** modifiquen los encajes de **"zorro"** y **"perro"**, respectivamente.

Cabezal de Atención Único

Supongamos que tenemos nuestra **matriz de encaje** y recibimos una nueva oración.
¿Cómo incorporamos el contexto para actualizar los encajes?
Un ejemplo ilustrativo es el efecto de **adjetivos** modificando **sustantivos**. (No representa realmente la complejidad de lo que ocurre en la capa pero sirve como intuición)
Ejemplo: Consideremos la oración:

"El zorro café saltó sobre el perro flojo."

Queremos que las palabras **"café"** y **"flojo"** modifiquen los encajes de **"zorro"** y **"perro"**, respectivamente.
El mecanismo de atención permite que cada palabra ajuste su significado con base en las palabras cercanas en la oración.

Espacio de Consultas y Llaves

En nuestro ejemplo, imaginemos que los **sustantivos** hacen la pregunta:

"¿Tengo algún adjetivo que me modifique?"

Espacio de Consultas y Llaves

En nuestro ejemplo, imaginemos que los **sustantivos** hacen la pregunta:

"¿Tengo algún adjetivo que me modifique?"

Los **adjetivos** pueden responder afirmativamente si están relacionados con el sustantivo.

Espacio de Consultas y Llaves

En nuestro ejemplo, imaginemos que los **sustantivos** hacen la pregunta:

"¿Tengo algún adjetivo que me modifique?"

Los **adjetivos** pueden responder afirmativamente si están relacionados con el sustantivo.
Para modelar esta interacción, utilizamos una operación matricial que genera la **matriz de consultas** (*Query Matrix*).

Espacio de Consultas y Llaves

En nuestro ejemplo, imaginemos que los **sustantivos** hacen la pregunta:

"¿Tengo algún adjetivo que me modifique?"

Los **adjetivos** pueden responder afirmativamente si están relacionados con el sustantivo.
Para modelar esta interacción, utilizamos una operación matricial que genera la **matriz de consultas** (*Query Matrix*).

- Cada palabra genera una consulta (**query**) que representa su búsqueda de contexto relevante.

Espacio de Consultas y Llaves

En nuestro ejemplo, imaginemos que los **sustantivos** hacen la pregunta:

"¿Tengo algún adjetivo que me modifique?"

Los **adjetivos** pueden responder afirmativamente si están relacionados con el sustantivo.
Para modelar esta interacción, utilizamos una operación matricial que genera la **matriz de consultas** (*Query Matrix*).

- Cada palabra genera una consulta (**query**) que representa su búsqueda de contexto relevante.
- La matriz de consultas vive en un espacio de dimensión menor al de los encajes originales.

Espacio de Consultas y Llaves

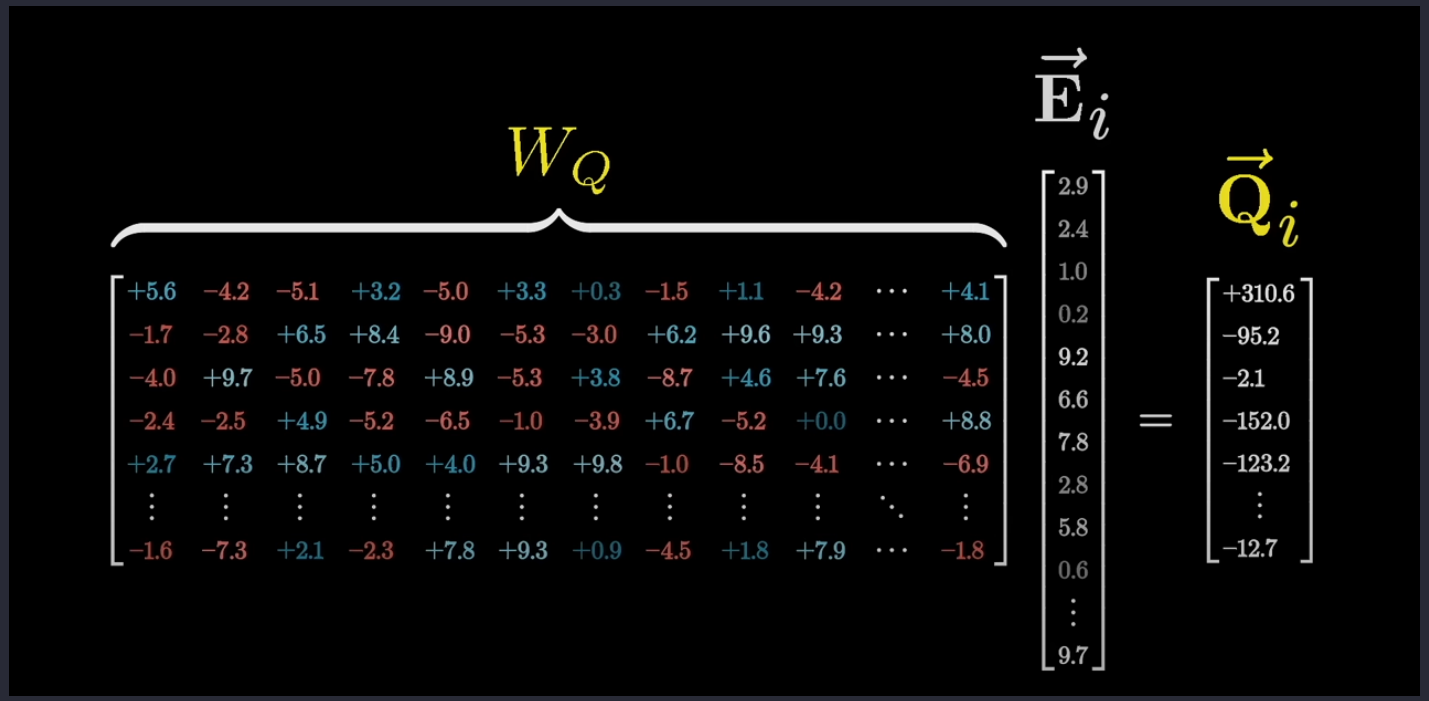
En nuestro ejemplo, imaginemos que los **sustantivos** hacen la pregunta:

"¿Tengo algún adjetivo que me modifique?"

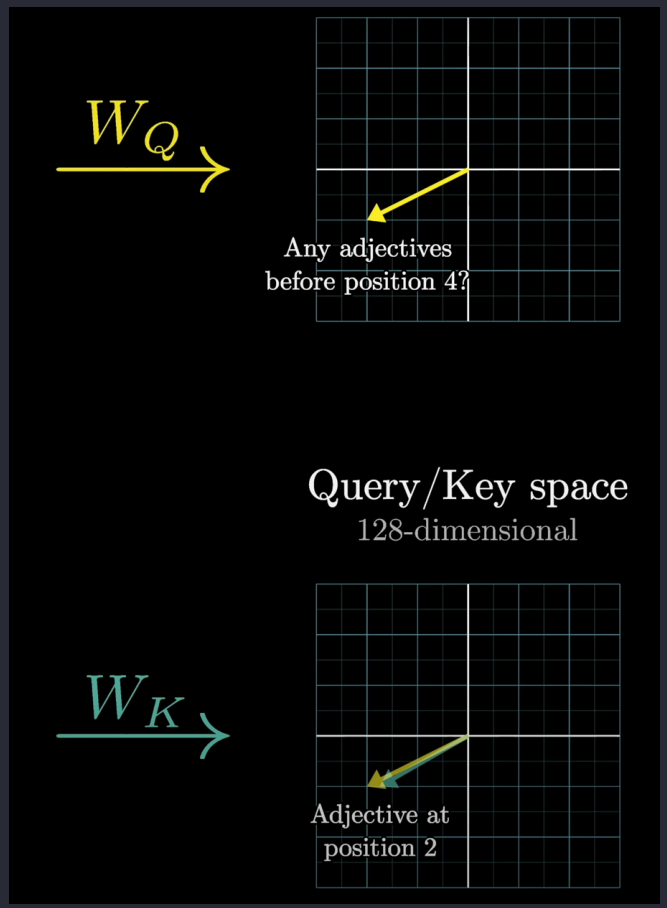
Los **adjetivos** pueden responder afirmativamente si están relacionados con el sustantivo.
Para modelar esta interacción, utilizamos una operación matricial que genera la **matriz de consultas** (*Query Matrix*).

- Cada palabra genera una consulta (**query**) que representa su búsqueda de contexto relevante.
- La matriz de consultas vive en un espacio de dimensión menor al de los encajes originales.
- Esta transformación permite que el modelo aprenda qué palabras deben influenciarse entre sí.

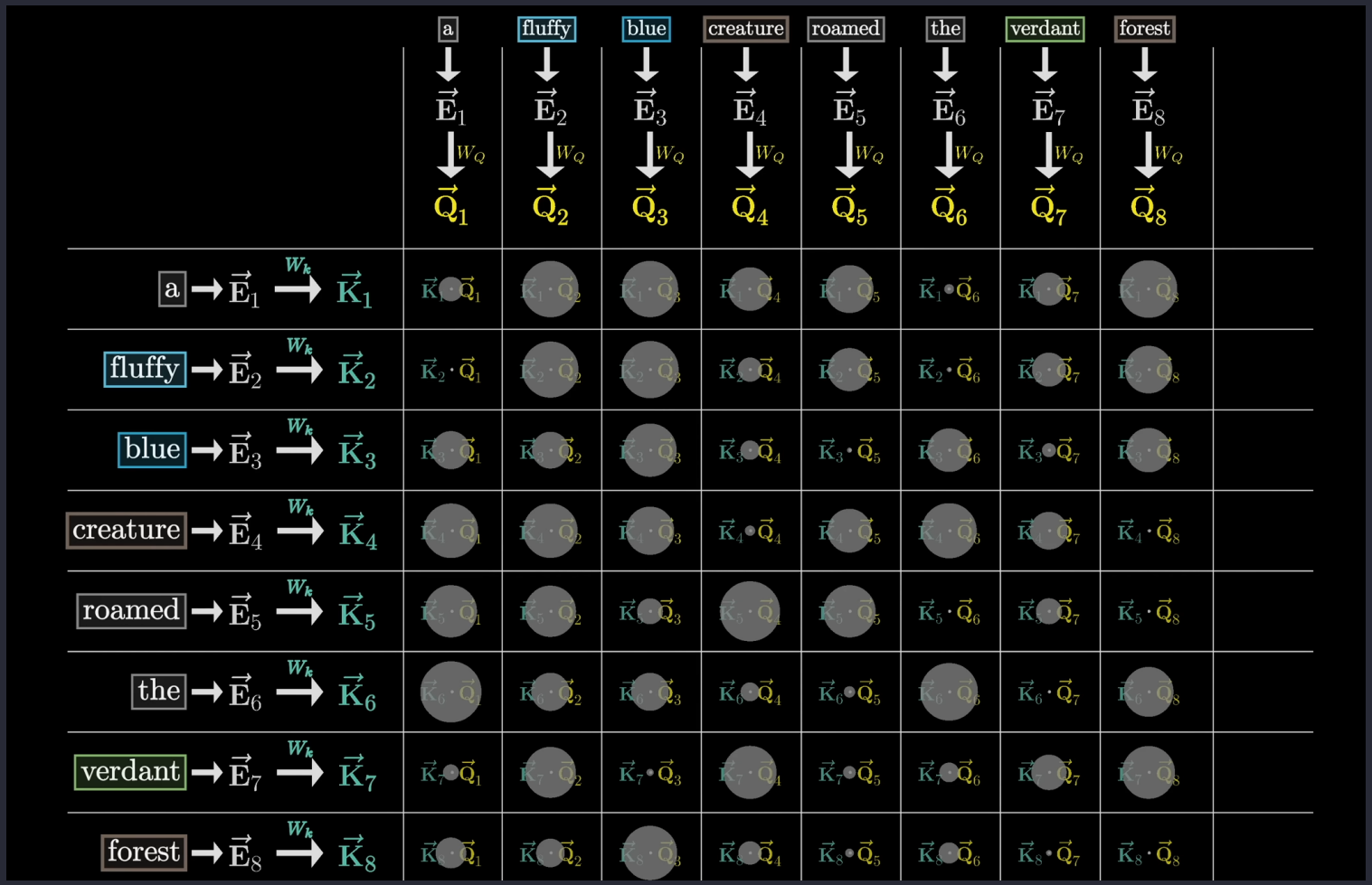
Espacio de Consultas y Llaves



Espacio de Consultas y Llaves



Espacio de Consultas y Llaves

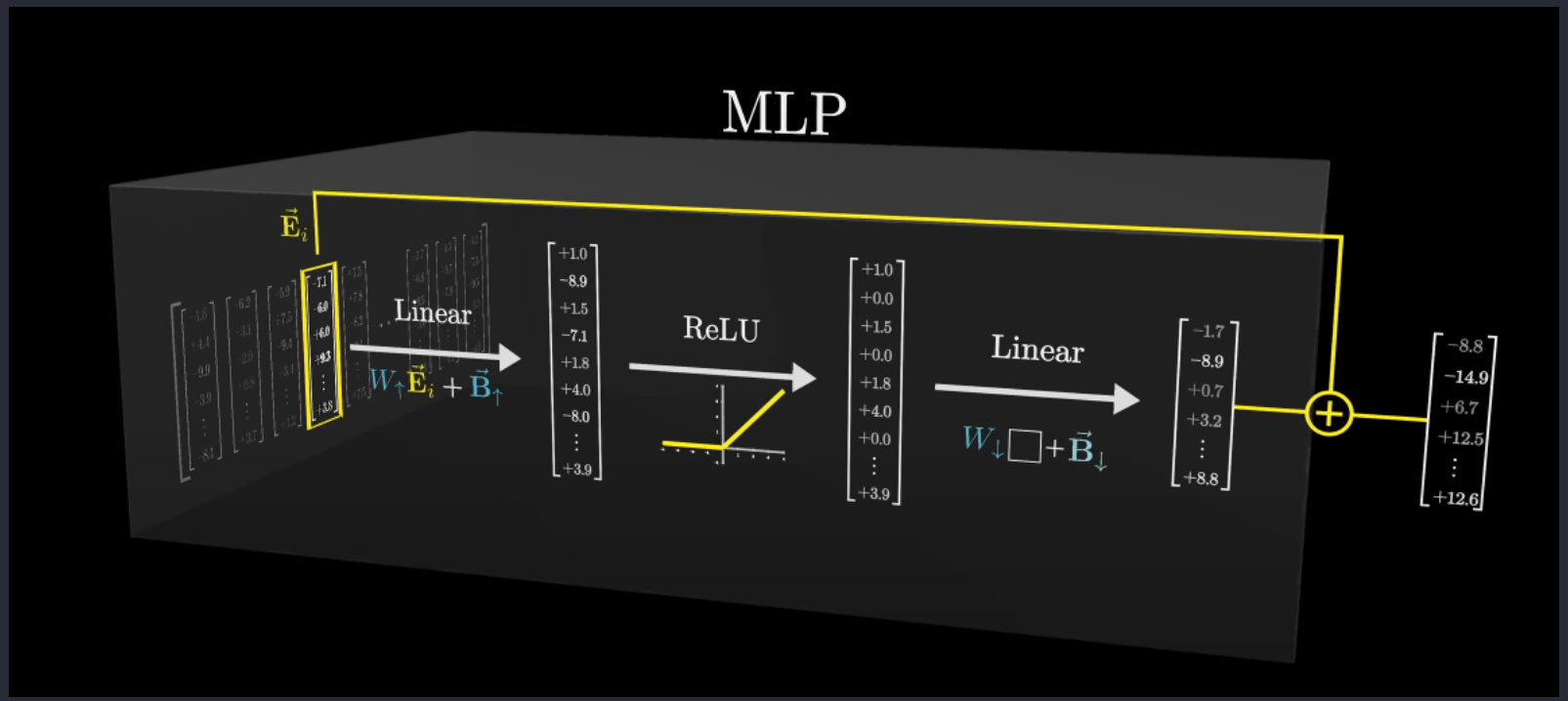


Instituto de Matemáticas UNAM

Formula .^Attention is all you need"

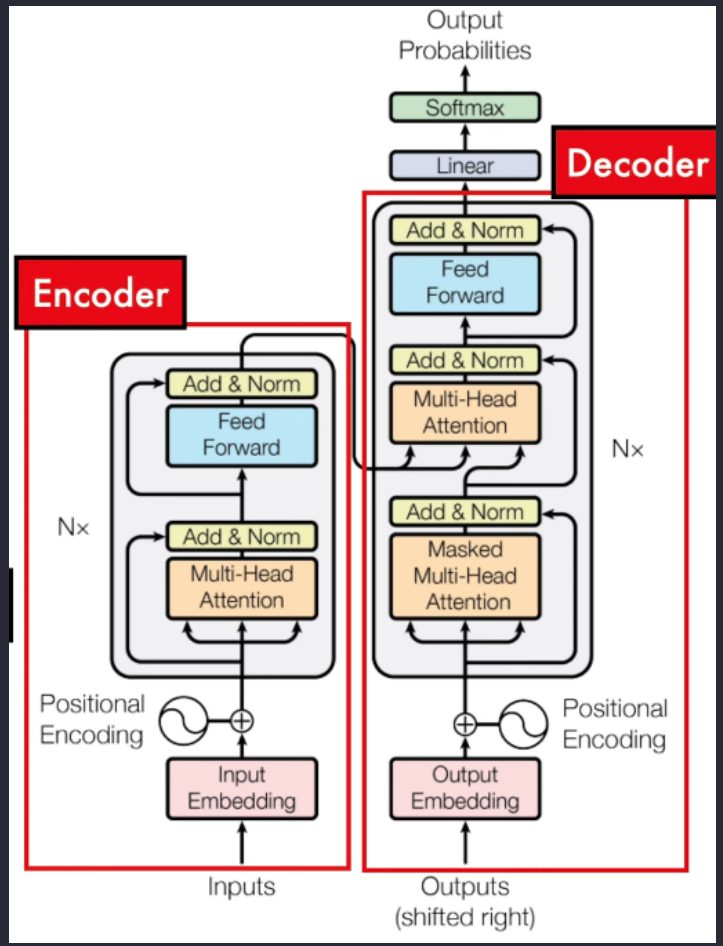
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

MLP


MLP
$\vec{E}_i$
Linear
$W_\uparrow \vec{E}_i + \vec{B}_\uparrow$
+1.0
−8.9
+1.5
−7.1
+1.8
+4.0
−8.0
+3.9
ReLU
+1.0
+0.0
+1.5
+0.0
+1.8
+4.0
+0.0
+3.9
Linear
$W_\downarrow \square + \vec{B}_\downarrow$
−1.7
−8.9
+0.7
+3.2
+8.8
−8.8
−14.9
+6.7
+12.5
+12.6

Contenidos

Diferentes Tipos de Transformers



Esquema ViT (Vision Transformer)

El modelo ViT adapta la arquitectura Transformer para procesar imágenes.

Esquema ViT (Vision Transformer)

El modelo ViT adapta la arquitectura Transformer para procesar imágenes.

1. **División en parches**: La imagen se divide en pequeñas regiones (*patches*), ya que los Transformers solo reciben datos de manera secuencial.

Esquema ViT (Vision Transformer)

El modelo ViT adapta la arquitectura Transformer para procesar imágenes.

1. **División en parches**: La imagen se divide en pequeñas regiones (*patches*), ya que los Transformers solo reciben datos de manera secuencial.
2. **Proyección lineal**: Cada parche pasa por una transformación (*embedding*) para convertirlo en un vector numérico.

Esquema ViT (Vision Transformer)

El modelo ViT adapta la arquitectura Transformer para procesar imágenes.

1. **División en parches**: La imagen se divide en pequeñas regiones (*patches*), ya que los Transformers solo reciben datos de manera secuencial.
2. **Proyección lineal**: Cada parche pasa por una transformación (*embedding*) para convertirlo en un vector numérico.
3. **Positional Encoding**: Se asocia una posición a cada parche, permitiendo que el modelo conserve la estructura espacial de la imagen.

Esquema ViT (Vision Transformer)

El modelo ViT adapta la arquitectura Transformer para procesar imágenes.

1. **División en parches**: La imagen se divide en pequeñas regiones (*patches*), ya que los Transformers solo reciben datos de manera secuencial.
2. **Proyección lineal**: Cada parche pasa por una transformación (*embedding*) para convertirlo en un vector numérico.
3. **Positional Encoding**: Se asocia una posición a cada parche, permitiendo que el modelo conserve la estructura espacial de la imagen.
4. **Token de clasificación**: Se añade un token especial en la posición 0. Este es un parámetro aprendible que se utilizará para hacer la predicción final.

Esquema ViT (Vision Transformer)

El modelo ViT adapta la arquitectura Transformer para procesar imágenes.

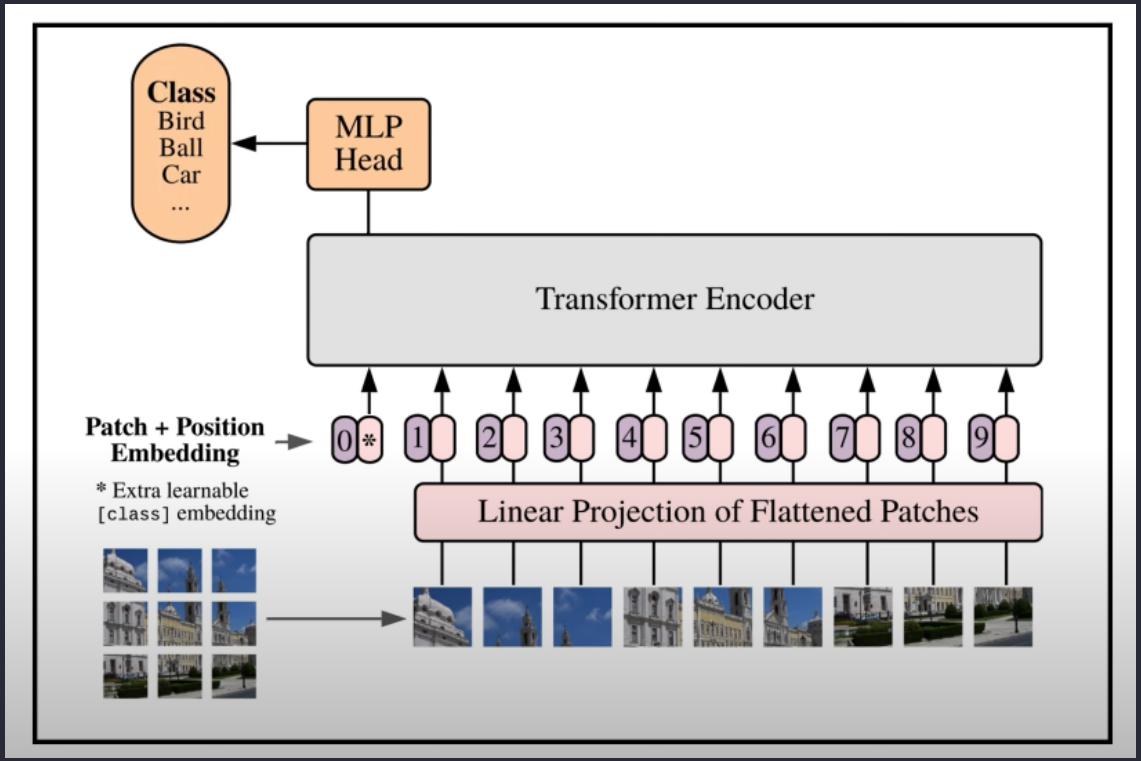
1. **División en parches**: La imagen se divide en pequeñas regiones (*patches*), ya que los Transformers solo reciben datos de manera secuencial.
2. **Proyección lineal**: Cada parche pasa por una transformación (*embedding*) para convertirlo en un vector numérico.
3. **Positional Encoding**: Se asocia una posición a cada parche, permitiendo que el modelo conserve la estructura espacial de la imagen.
4. **Token de clasificación**: Se añade un token especial en la posición 0. Este es un parámetro aprendible que se utilizará para hacer la predicción final.
5. **Transformer Encoder**: Se aplican bloques de *Atención* y *MLP*, obteniendo una representación enriquecida.

Esquema ViT (Vision Transformer)

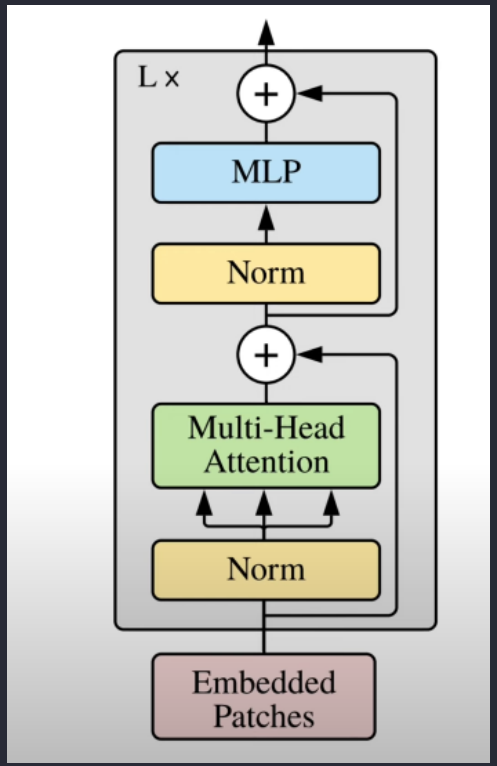
El modelo ViT adapta la arquitectura Transformer para procesar imágenes.

1. **División en parches**: La imagen se divide en pequeñas regiones (*patches*), ya que los Transformers solo reciben datos de manera secuencial.
2. **Proyección lineal**: Cada parche pasa por una transformación (*embedding*) para convertirlo en un vector numérico.
3. **Positional Encoding**: Se asocia una posición a cada parche, permitiendo que el modelo conserve la estructura espacial de la imagen.
4. **Token de clasificación**: Se añade un token especial en la posición 0. Este es un parámetro aprendible que se utilizará para hacer la predicción final.
5. **Transformer Encoder**: Se aplican bloques de *Atención* y *MLP*, obteniendo una representación enriquecida.
6. **Predicción**: Se usa el token de clasificación para generar la salida del modelo.

Esquema ViT (Vision Transformer)



Esquema ViT (Vision Transformer)



Analogia NLP

Analogia NLP

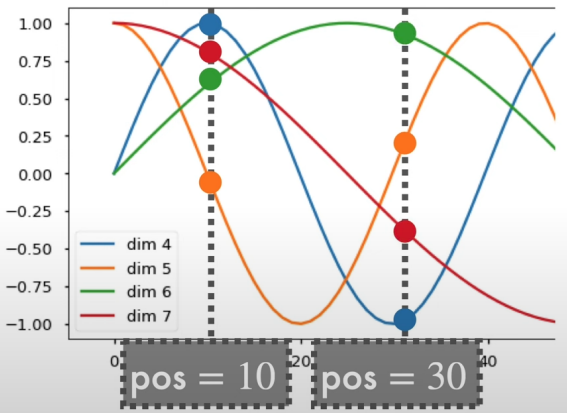
Positional Embeddings

How do we "label" positions?

Hand-crafted position embeddings:

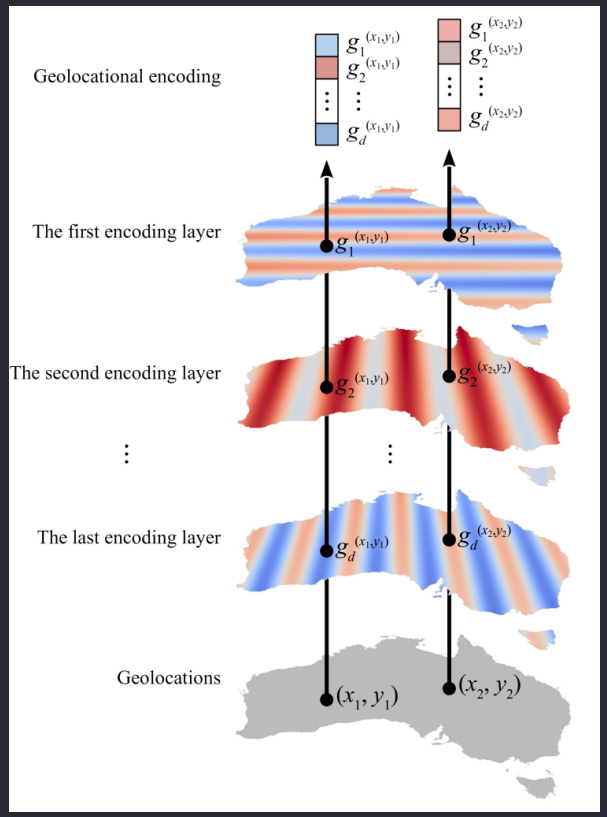
$$\mathrm{PE}(\mathrm{pos}, 2i) = \sin\left(\frac{\mathrm{pos}}{10{,}000^{2i/D}}\right)$$

$$\mathrm{PE}(\mathrm{pos}, 2i+1) = \cos\left(\frac{\mathrm{pos}}{10{,}000^{2i/D}}\right)$$



Contenidos

Spatioformer: Un Transformer Geo-Codificado



Referencia: Spatioformer: A Geo-encoded Transformer for Large-Scale Plant Species Richness Prediction

Instituto de Matemáticas UNAM