Cursos Extraordinarios

Verano 2024

“Inteligencia Artificial y Grandes Modelos de Lenguaje: Funcionamiento, Componentes Clave y Aplicaciones”

Zaragoza, del 3 al 5 de julio

Transformer

- **Arquitectura**
  - Residual layers
  - Layer normalization
  - Multi-head attention
  - Feed forward
  - Positional embedding
  - Embedding de entrada
  - Arquitectura general

Transformer

- **Attention is all you need (Vaswani 2017) > 100k cites**

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N. Kaiser L, Polosukhin, I. (2017). Attention is all you need. Advances in neura l information processing systems, 30, 5998-6008..

  - Arquitectura muy novedosa
  - Muchos de los mejores resultados
    - Traducción
    - Modelado de lenguaje
    - NLP: responder preguntas, resúmenes
    - Language understanding,
    - Speech recognition (ASR), synthesis
    - Reconocimiento de imágenes, segmentación
    - Multimodal image+text
    - DNA folding
    - Drug discovery
    - …

Attention Is All You Need

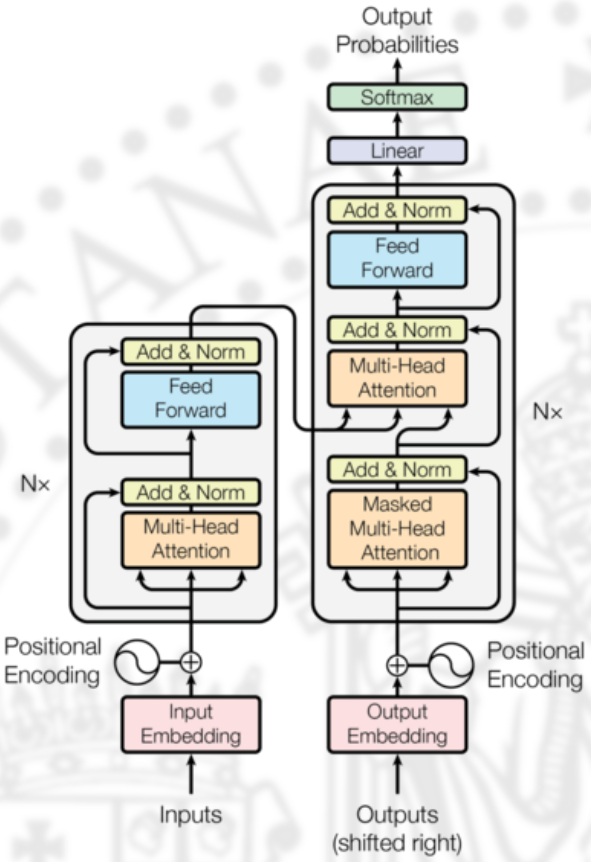
Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Transformer

- **Architecture:**
  - Encoder / Decoder
  - Similar a seq2seq
  - Algunas aplicaciones utilizan solo encoder/decoder



  - Encoder crea una representación útil de la entrada
  - El decoder predice la siguiente palabra a partir de la anterior
  - No utiliza CNNs o LSTMs (de ahí el título)
  - Self attention (multihead) es un nuevo tipo de bloque y la parte más novedosa

Transformer

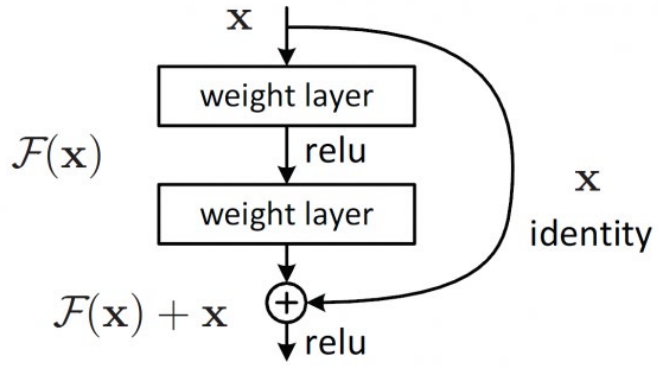
- **Arquitectura**
  - **Residual layers**
  - Layer normalization
  - Multi-head attention
  - Feed forward
  - Positional embedding
  - Arquitectura general

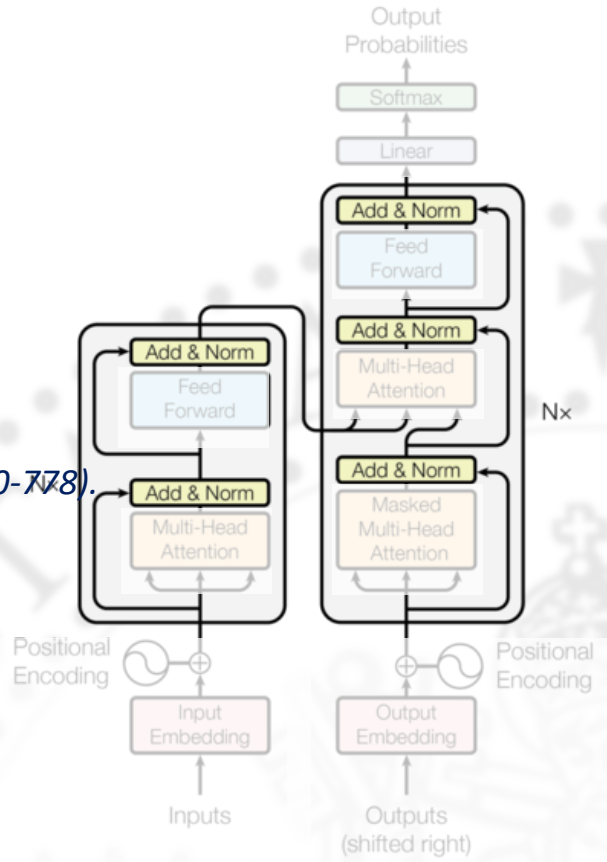
Transformer

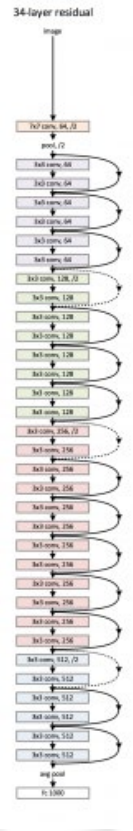
- **Residual layers (121k citas)**

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

  - La idea principal es modificar la representación **incrementalmente**







  - El problema de *vanishing gradients* impedía crear redes muy profundas:
    - En el paper demuestran que pueden entrenar redes de 28 a 152 capas

Transformer

- **Arquitectura**
  - Residual layers
  - **Layer normalization**
  - Multi-head attention
  - Feed forward
  - Positional embedding
  - Embedding de entrada
  - Arquitectura general

Transformer

- **Normalization**
  - Batch normalization (Ioffe 2015)

Ioffe, S. and Szegedy, C., 2015, June. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). pmlr..

  - Layer normalization (Ba 2016)

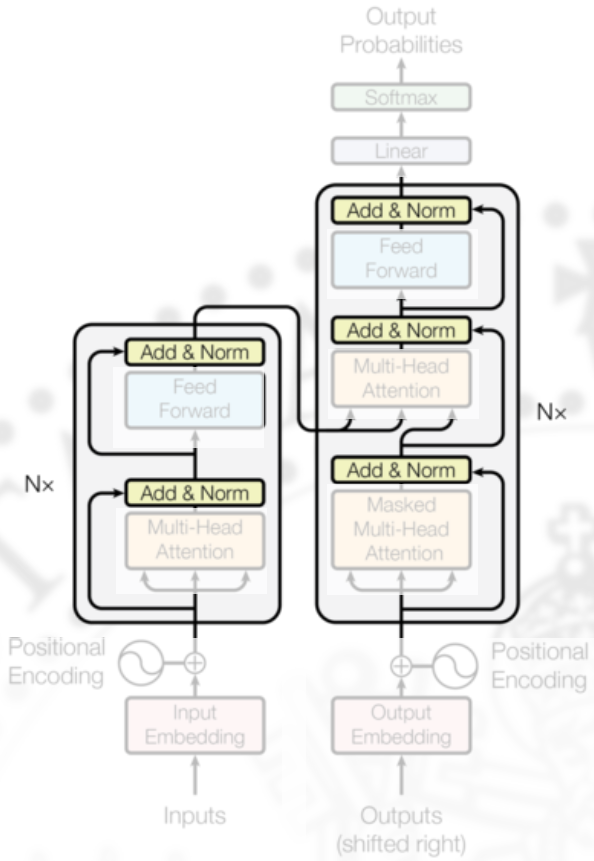
Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. arXiv preprint arXiv:1607.06450.

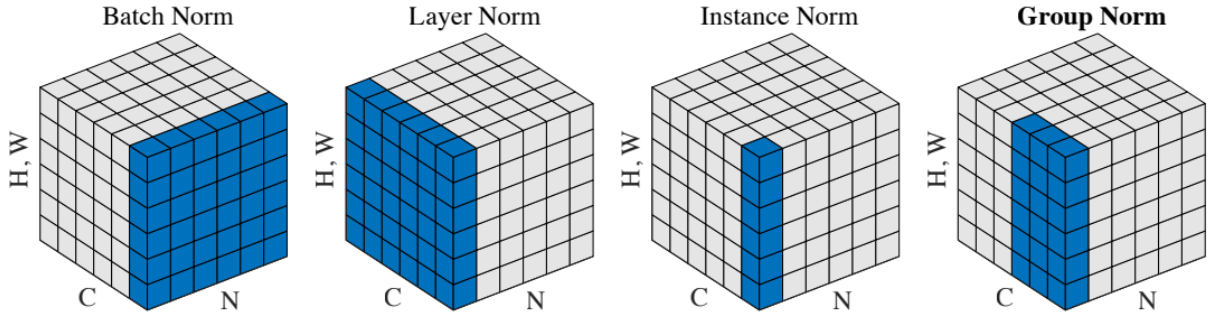
  - Instance Normalization(Wu 2016)

Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2016). Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022.

  - Group Normalization(Wu 2018)

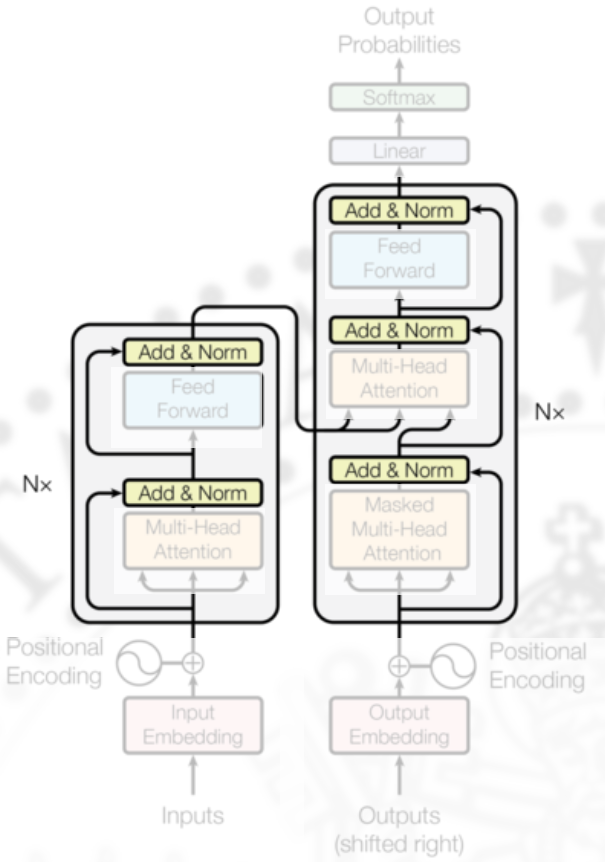
Wu, Y., & He, K. (2018). Group normalization. In Proceedings of the European conference on computer vision (ECCV) (pp. 3-19).

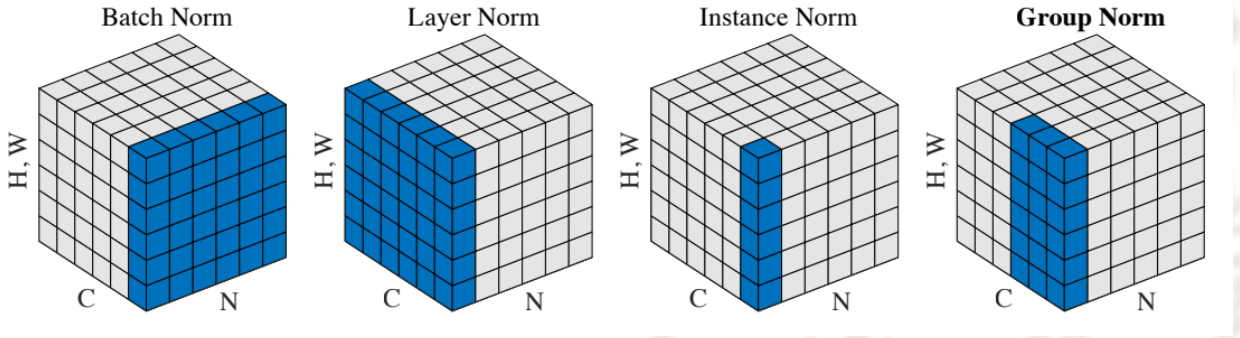




Transformer

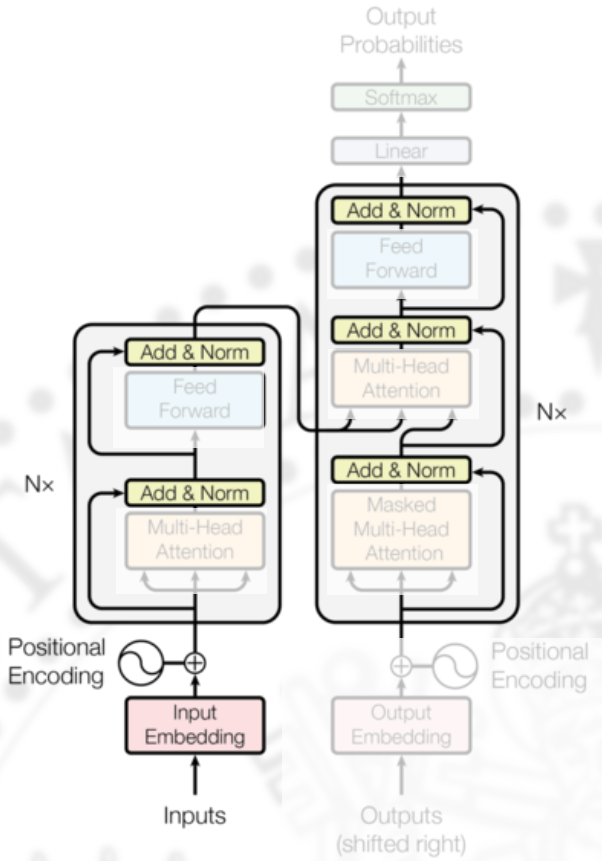
- **Layer normalization**
  - layer normalization es una alternativa a batch normalization, BN
  - El objetivo y mecanismos son similares a BN
  - La diferencia es que la media y la std se calculan sobre la dimensión de los vectores
    - Esto actúa como un control sobre el tamaño/norma de los vectores





Transformer

- **Layer normalization**
  - Pasos
    - 1. Normalizar el vector
      - dimensión de representación/canales
    - 2. Aplicar una transformación lineal
      - reescalar los datos y aplicar sesgo



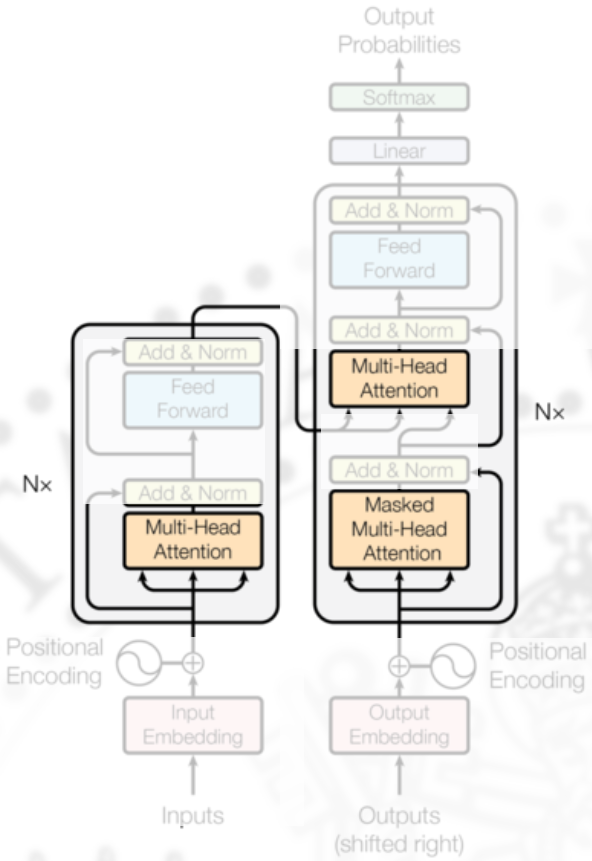

```python
class RMSNorm(torch.nn.Module):
    def __init__(self, d_model=512, eps=1e-8, **kwargs):
        super().__init__()
        self.eps = eps
        self.scale = torch.nn.Parameter(torch.ones(1, 1, d_model))

    def forward(self, x):
        x = x / (torch.sqrt(torch.mean(x**2, dim=-1, keepdim=True)) + self.eps)
        return x * self.scale
```

  - Caso especial: RMSNorm, no se tiene el sesgo/bias en cosideración

Transformer

- **Arquitectura**
  - Residual layers
  - Layer normalization
  - **Multi-head attention**
  - Feed forward
  - Positional embedding
  - Embedding de entrada
  - Arquitectura general

Transformer

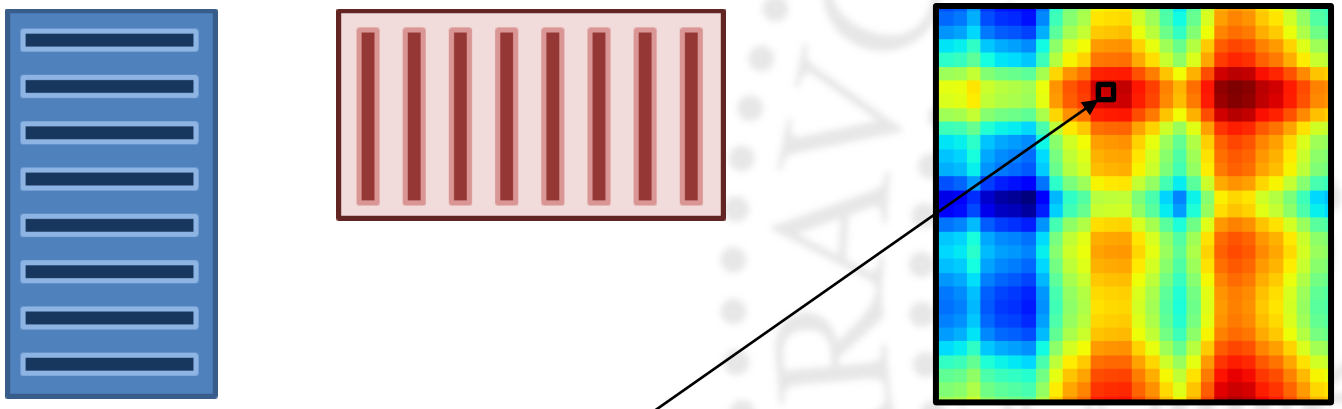


- **Multi-head Attention (MHA)**
  - Self attention
    - Input: tamaño de la secuencia (n x d)
    - Output: tamaño de la secuencia (n x d)
  - Cross attention
    - Input: tamaño de la secuencia (n x d)
      - » señal externa (m x d)
    - Output: tamaño de la secuencia (n x d)
  - La salida es una combinación lineal de las entradas
    - Similar a la convolución, pero con un filtro dinámico (dependiente de las entradas)
      - » El decodificador tiene una restricción causal
  - Heads -> h (de 1 to 96) (módulos MHA trabajando en paralelo)
    - Pasos:
      - Partir el vector en h partes
      - Aplicar atención a cada parte
      - Concatenar

Transformer

- **Self attention**
  - El núcleo del transformer es la operación self-attention
    - Dadas dos secuencias de vectores, k, q, de longitud T y dimensión D
    - Se usa el producto escalar de todos los vectores para identificar el parecido
      - En procesado de señal es similar a la idea de la autocorrelación y la correlación cruzada

$\boldsymbol{q}$ $(T \times D)$ $\cdot$ $\boldsymbol{k}^T$ $(D \times T)$ $=$ $\boldsymbol{C}$ (CrossSimilarity)



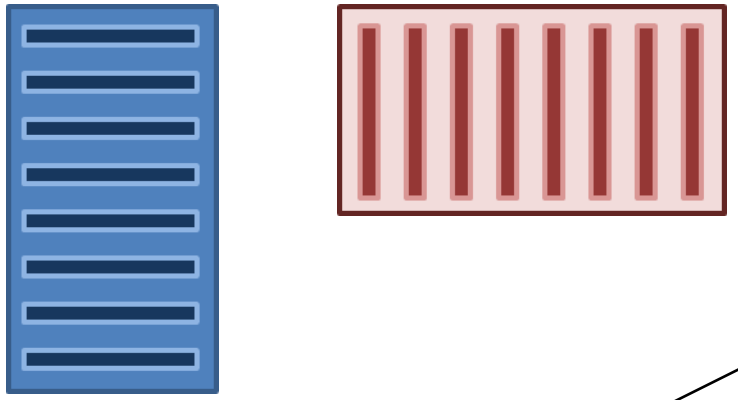
Elemento C_{ij}

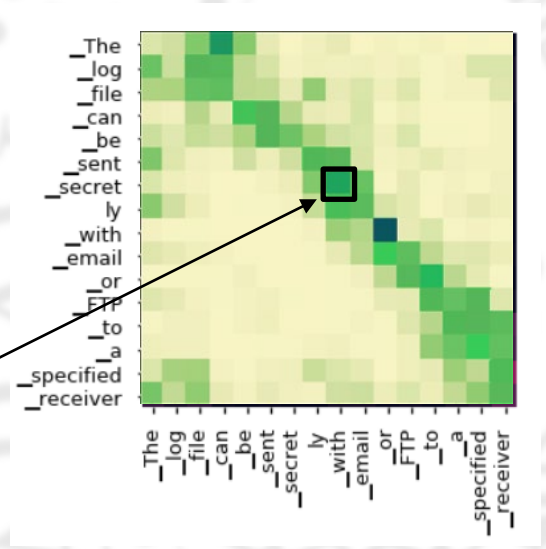
Similitud, vectores: $\boldsymbol{k}_i \cdot \boldsymbol{q}_j^T$

Transformer

- **Self attention**
  - El núcleo del transformer es la operación self-attention
    - Dadas dos secuencias de vectores, k, q, de longitud T y dimensión D
    - Se usa el producto escalar de todos los vectores para identificar el parecido
    - Se usa la operación softmax para mapear la entrada a la salida (matriz de alineamiento)

$$softmax(\underset{(T \times D)}{\boldsymbol{q}} \cdot \underset{(D \times T)}{\boldsymbol{k}^T}) = \boldsymbol{A}$$ (Self attention)





$\sum_j A_{ij} = 1$
(sum 1 cols.)

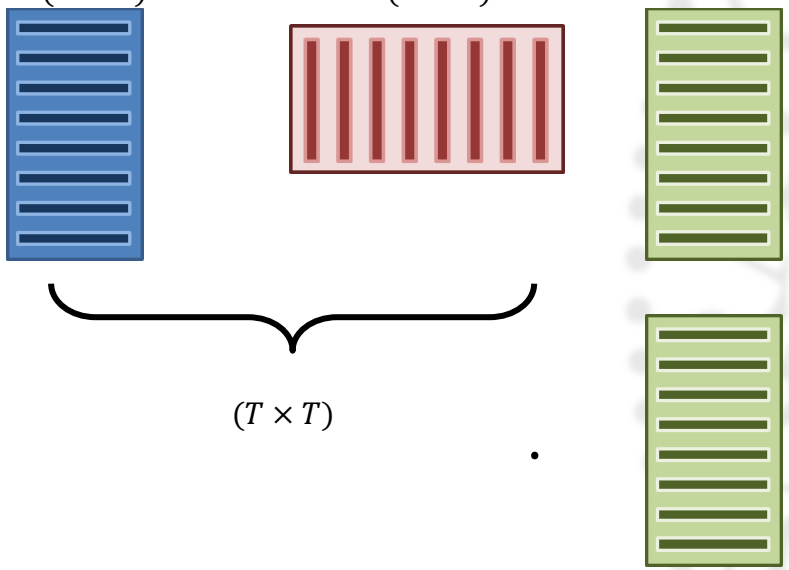
Elemento A_{ij}
Grado de influencia del elemento j de la entrada sobre la salida i

https://nlp.seas.harvard.edu/2018/04/03/attention.html

Transformer

- **Self attention**
  - El núcleo del transformer es la operación self-attention
    - Dadas dos secuencias de vectores, k, q, de longitud T y dimensión D
    - Se usa el producto escalar de todos los vectores para identificar el parecido
    - Se usa la operación softmax para mapear la entrada a la salida (matriz de alineamiento)
    - El mapping se usa para obtener la salida como una combinación lineal de elementos de la entrada

$$softmax(\ \underset{(T\times D)}{\boldsymbol{q}}\ \cdot\ \underset{(D\times T)}{\boldsymbol{k}^T}\ )\cdot \boldsymbol{v} = \boldsymbol{o}\quad \text{(output)}$$

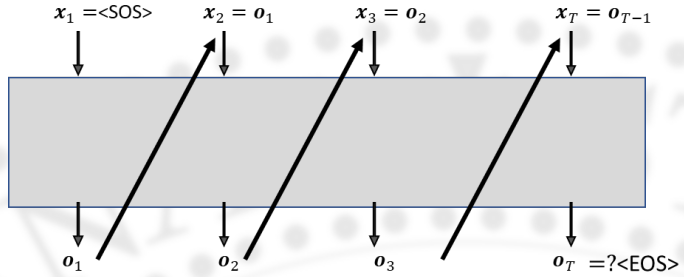


- La secuencia de salida puede ser **reordenada**
- Cada elemento de salida puede tener influencia de **cualquier zona** de la secuencia de entrada

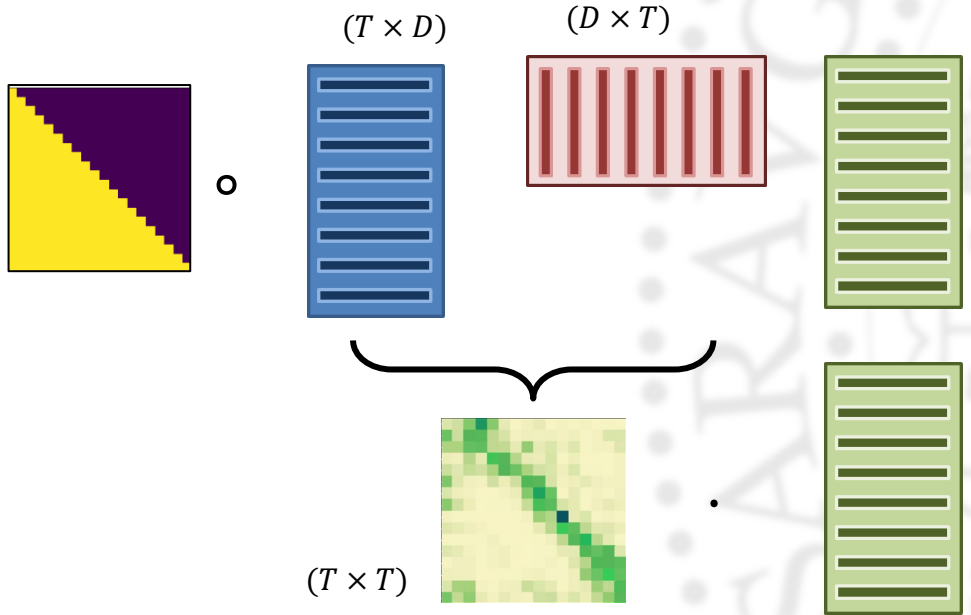
- La salida es la secuencia de entrada multiplicada por el alineamiento/mapping
- El proceso de alineamiento/mapeo puede ser interpretado como un **mecanismo de atención**

Transformer

- **Causal self attention**
  - Forma parte del **decoder** en seq2seq architectures
    - El objetivo es predecir el siguiente token dados los anteriores
    - Para ello hay que modifcar la capa self-attention para que no haga uso de los tokens futuros en el mapping mediante una máscara
    - Se aplica al alineamiento por una mascara de forma que los valores por encima de la diagonal sean 0

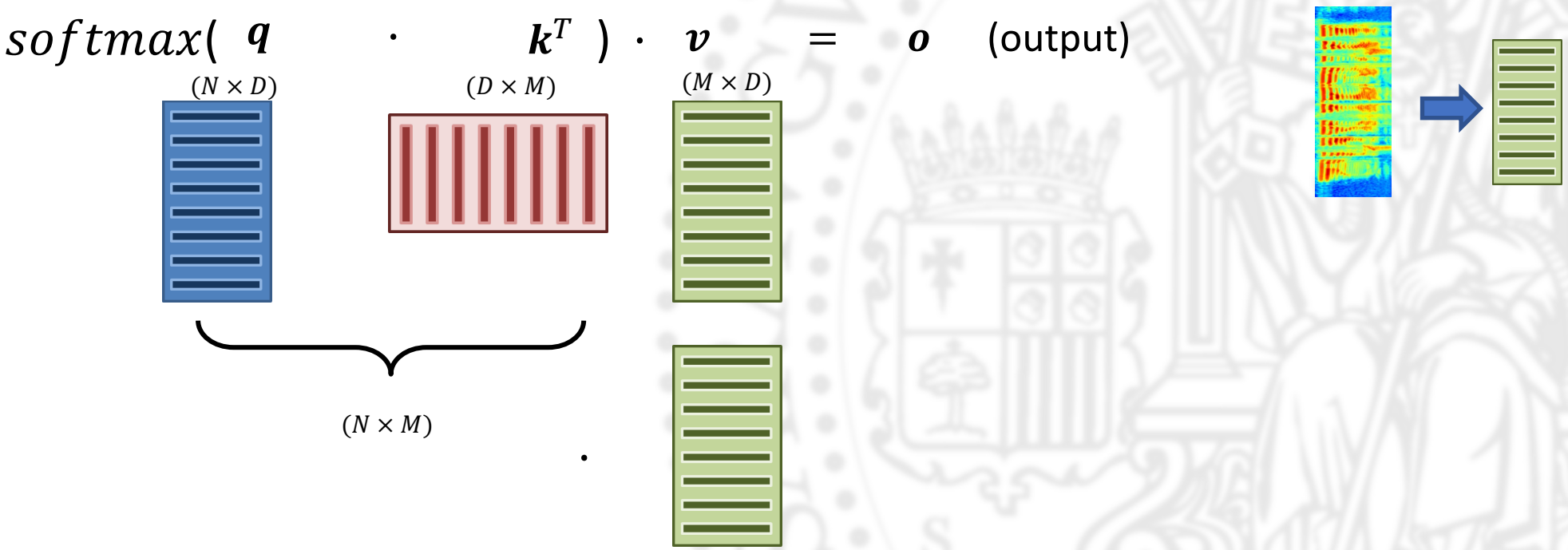


$$softmax(M \circ (\boldsymbol{q} \cdot \boldsymbol{k}^T)) . \boldsymbol{v} = \boldsymbol{o} \quad \text{(output)}$$



Transformer

- **Cross attention**
  - Forma parte del **decoder** en seq2seq architectures
    - En este caso las señales que intervienen son de diferente naturaleza
      - Puede ser una del encoder de texto y la otra del decoder
      - También se usa para mezclar modalidades, imagen-texto audio-texto video texto…

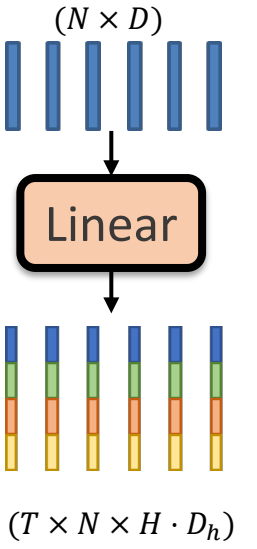


Transformer

- **Multi-Head attention**
  - La dimensión de representación: se parte en h subvectores: $\boldsymbol{k}_h, \boldsymbol{q}_h, \boldsymbol{v}_h$
  - Se aplica self-attention a cada subvector

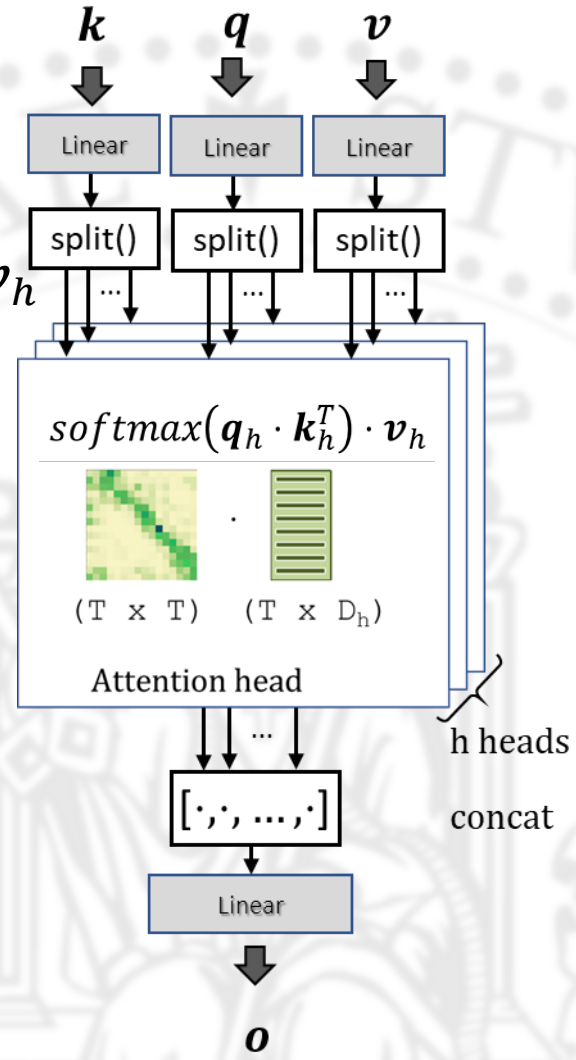
$$Attention(\boldsymbol{q}_h, \boldsymbol{k}_h, \boldsymbol{v}_h) = softmax\left(\frac{\boldsymbol{q}_h \cdot \boldsymbol{k}_h^T}{scale}\right) \cdot \boldsymbol{v}_h$$

    - Constante (scale = 1 / $\sqrt{D_h}$) juega el papel de una temperatura




```
self.linear = nn.Linear(d, h*dh)

q = self.linear(q).reshape(b, n, h, dh)
```

split the ouput in **h subvectors** of dimension dh

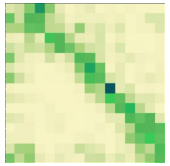


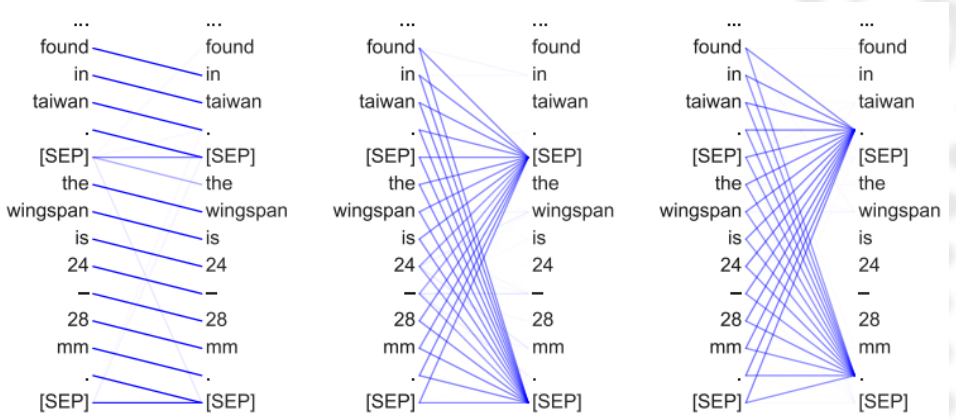
  - Las secuencias que se obtienen de cada head se concatenan

Transformer

- **Self attention**
  - Numerosos estudios analizan las matrices resultantes
    - » Se observa que analiza diferentes niveles de relaciones

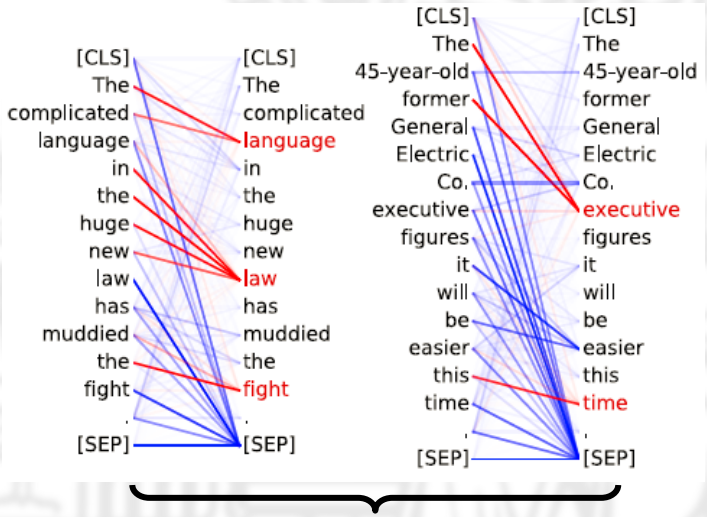
$$softmax(\boldsymbol{q} \cdot \boldsymbol{k}^T) \cdot \boldsymbol{v}$$





Attention to previous item

Attention to end of sentence



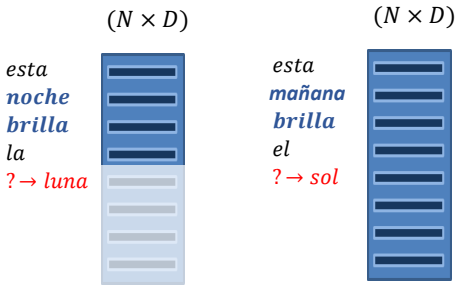
Noun modifiers, e.g. determiners attend to their noun

https://medium.com/synapse-dev/understanding-bert-transformer-attention-isnt-all-you-need-5839ebd396db

https://medium.com/dair-ai/aspects-of-language-captured-by-bert-32bc3c54016f

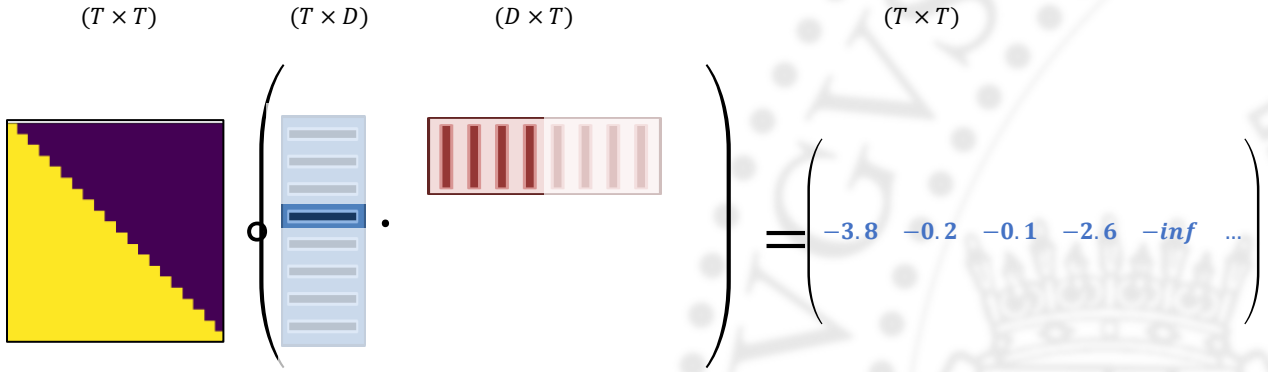
Transformer

- **Ejemplo**



Transformer

- **Ejemplo**



Transformer

- **Ejemplo**

$(N \times D)$ $(N \times D)$

$$softmax\begin{pmatrix} -2.8 & -0.2 & -0.1 & -2.6 & -inf & \dots \end{pmatrix} \cdot$$

$=$

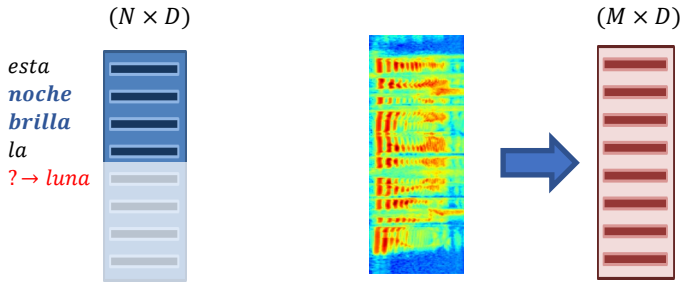
$(N \times N)$ $(N \times D)$ $(N \times D)$

$$\begin{pmatrix} 0.03 & \mathbf{0.45} & \mathbf{0.48} & 0.04 & 0 & \dots \end{pmatrix} \cdot$$

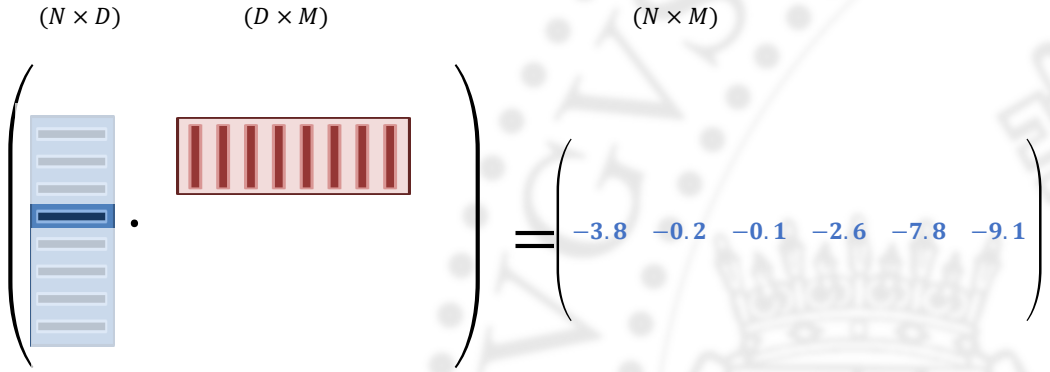
$=$

Transformer

- **Ejemplo**



Cross-atencion



Transformer

- **Ejemplo**