Visual transformers

Blanca Vázquez

3 de abril de 2024

Contexto

Los mecanismos de atención mejoran los modelos de aprendizaje profundo al **centrarse** selectivamente en **elementos de entrada importantes**, mejorando la precisión de la predicción y la eficiencia computacional.

En psicología, la atención es el proceso cognitivo de **concentrarse** selectivamente en una o algunas cosas mientras se ignoran otras.

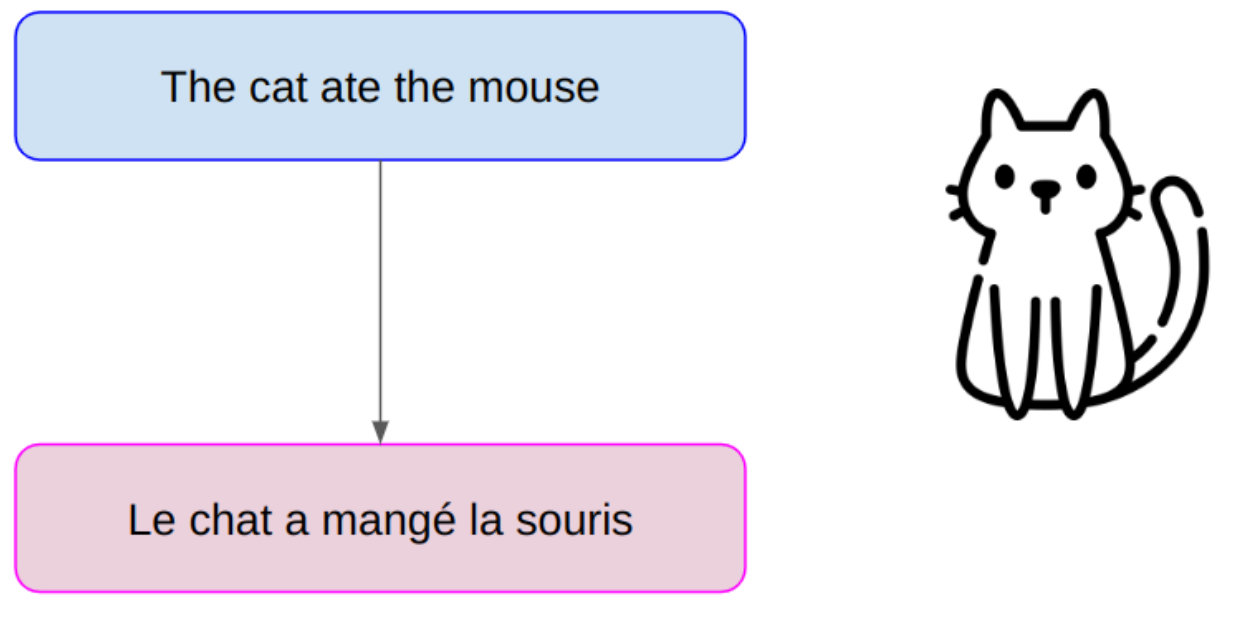
Ejercicio

Retomemos

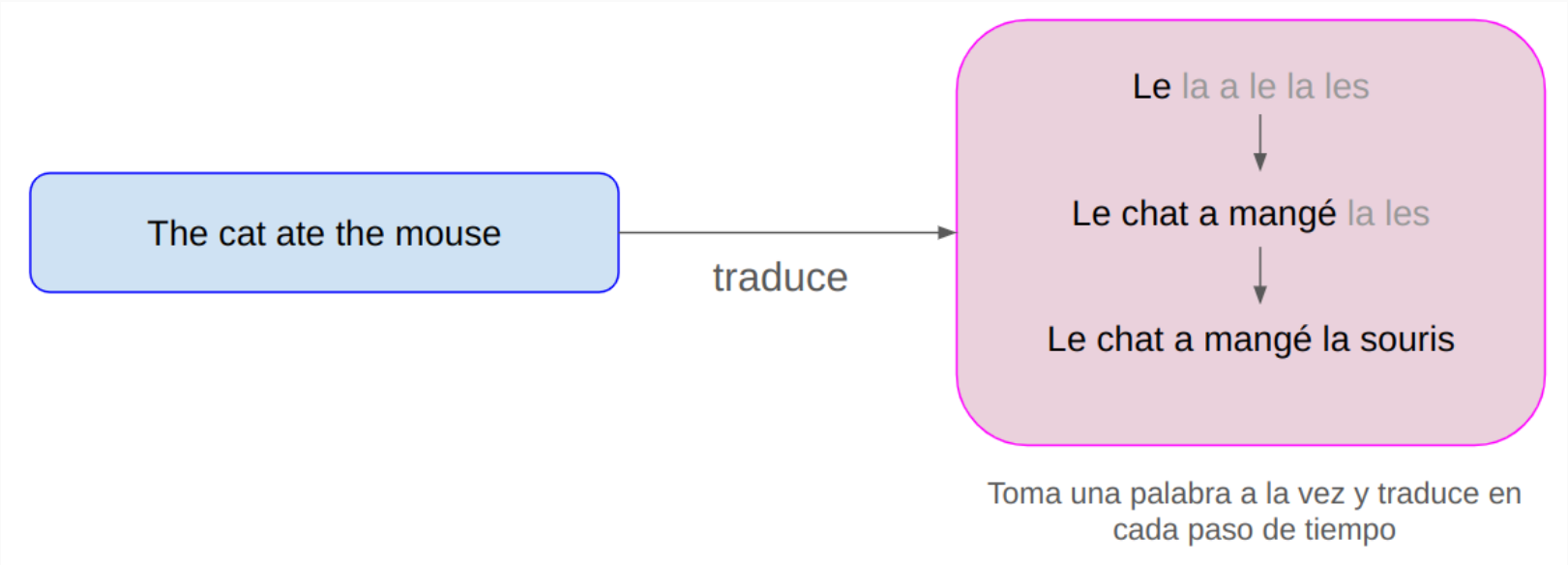
En psicología, la atención es el proceso cognitivo de **concentrarse** selectivamente en una o algunas cosas mientras se ignoran otras.

Una red neuronal también puede imitar las acciones del cerebro humano de forma simplificada. El mecanismo de atención es un intento de implementar la misma acción de **concentrarse** selectivamente en algunas cosas relevantes, mientras otras son ignoradas.

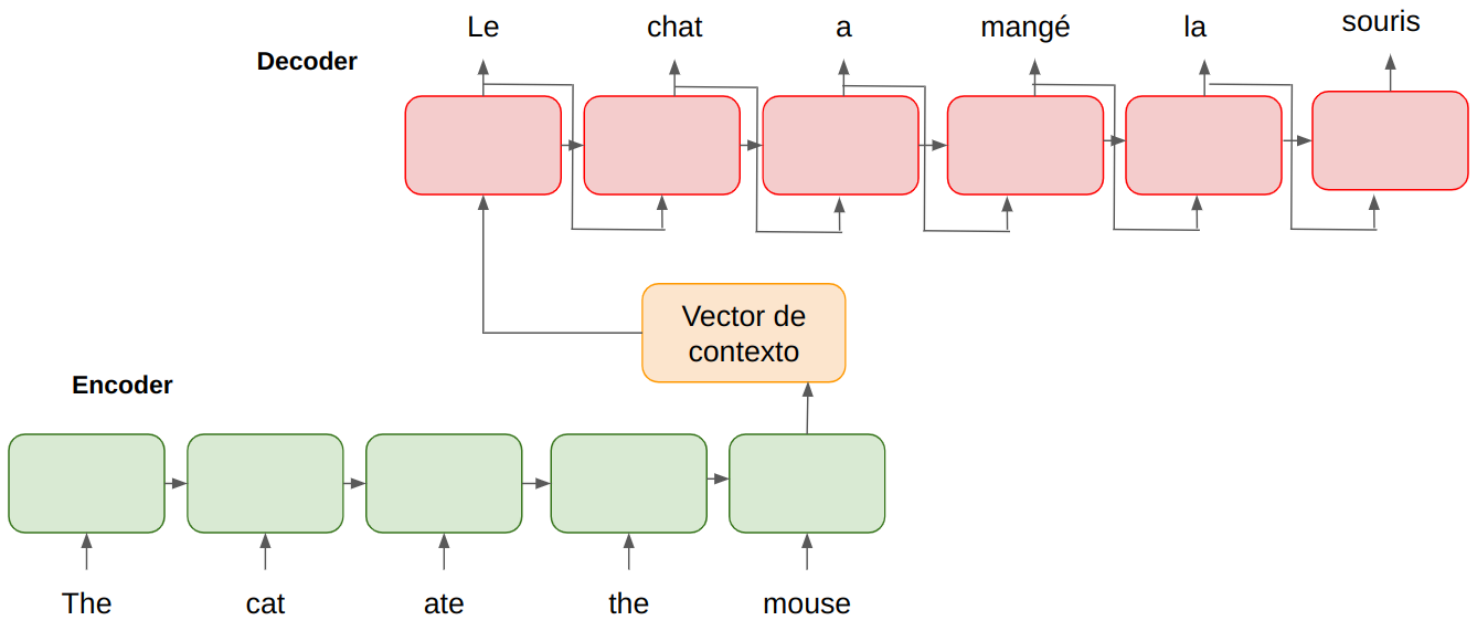
Traducción de sentencias (ejemplo 1)



Traducción de sentencias (vista general)

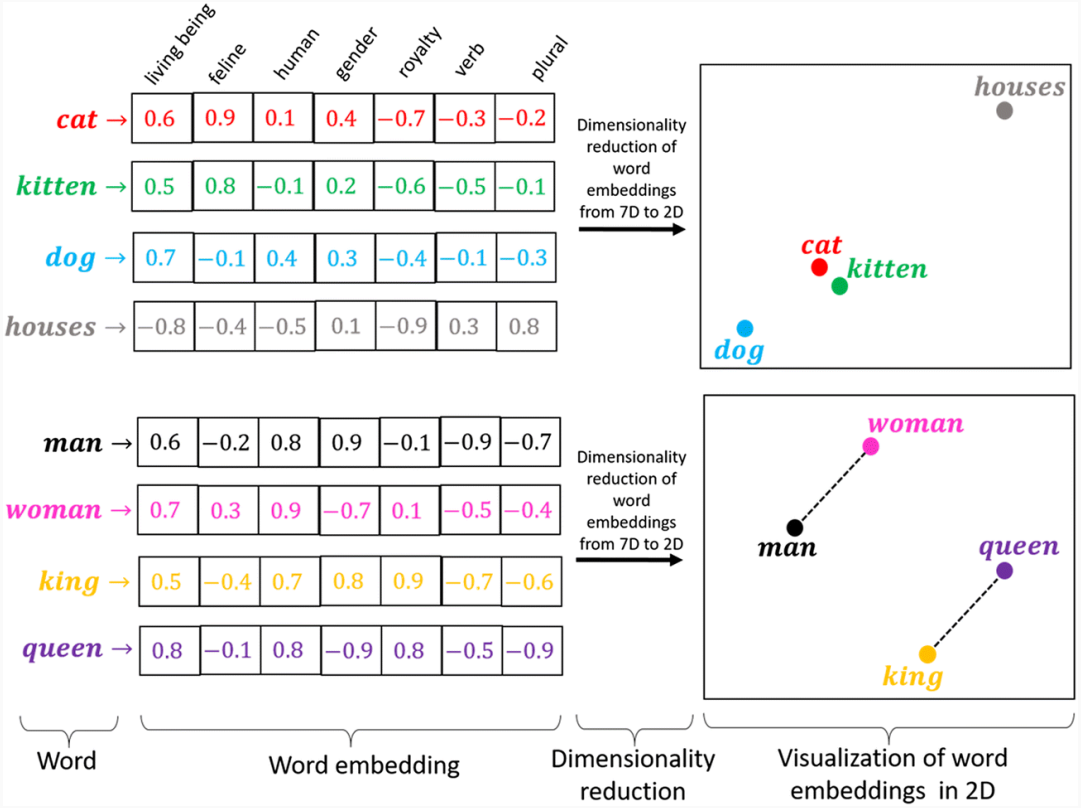


Sistema de traducción basado en la arquitectura encoder-decoder

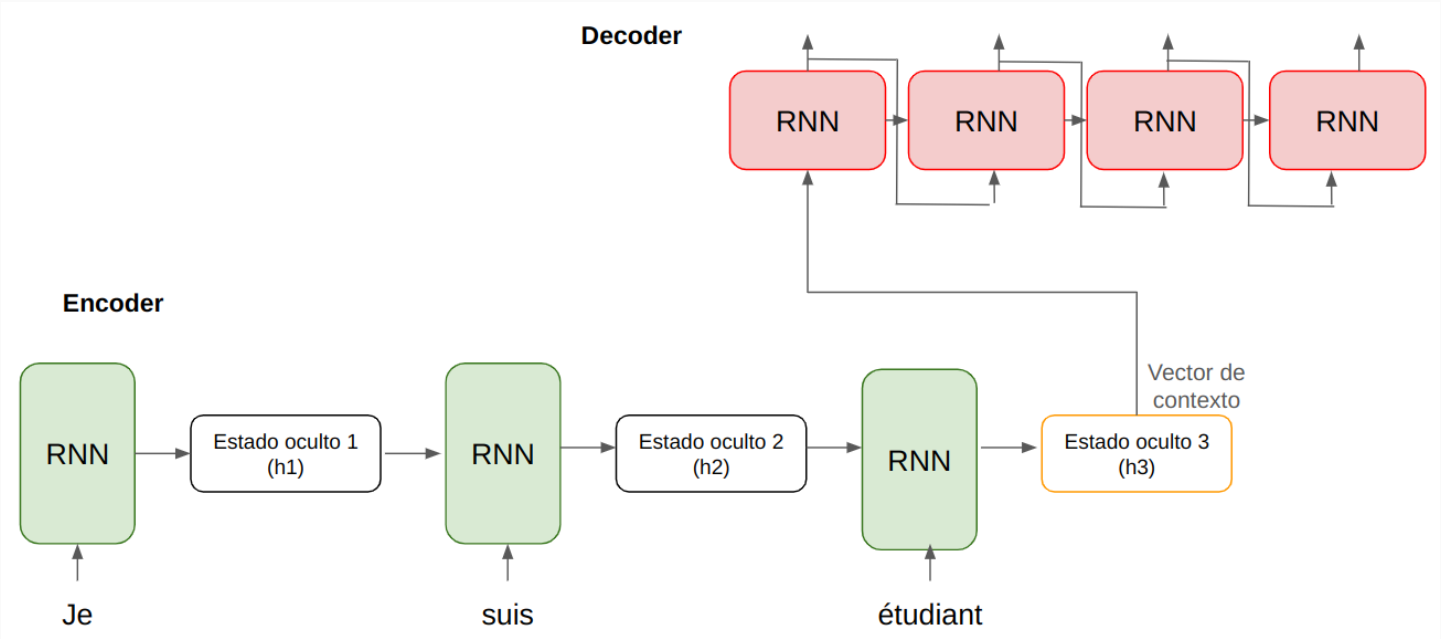


Embeddings

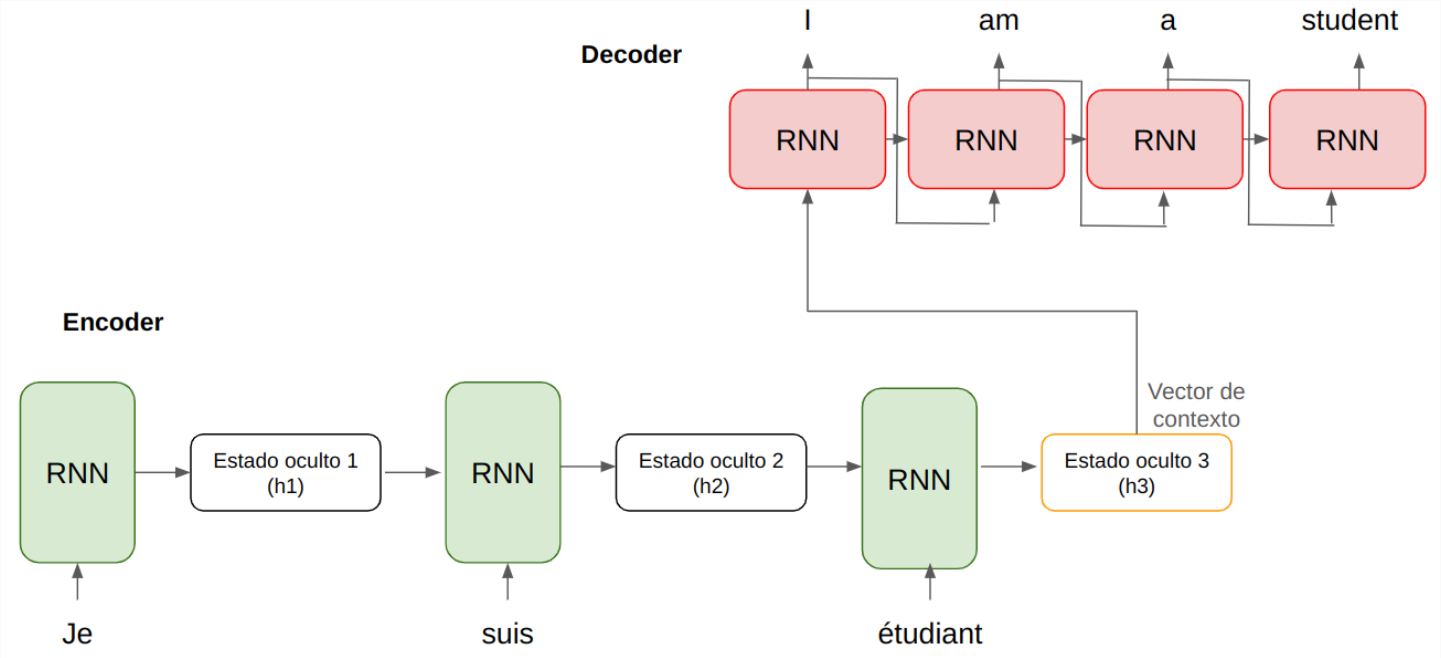
Un **embedding** es, básicamente, una representación vectorial de una palabra en un espacio multidimensional, donde palabras similares se encuentran cercanas entre sí



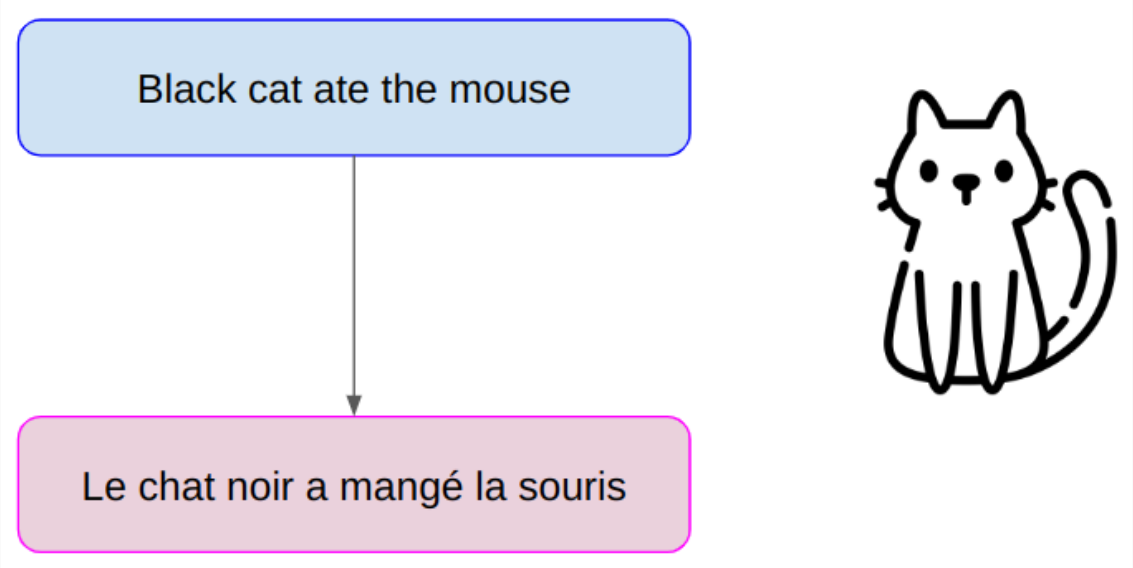
Sistema de traducción basado en la arquitectura encoder-decoder



Sistema de traducción basado en la arquitectura encoder-decoder



Traducción de sentencias (ejemplo 2)



Problemas con las redes encoder-decoder

- El codificador lee la sentencia completa y hace un resumen.
- El decodificador toma el resumen y lo traduce.
- Si el resumen es malo, entonces ...
- A esto se le llama ‘problema de dependencia a largo plazo’.

Introducción al mecanismo de atención

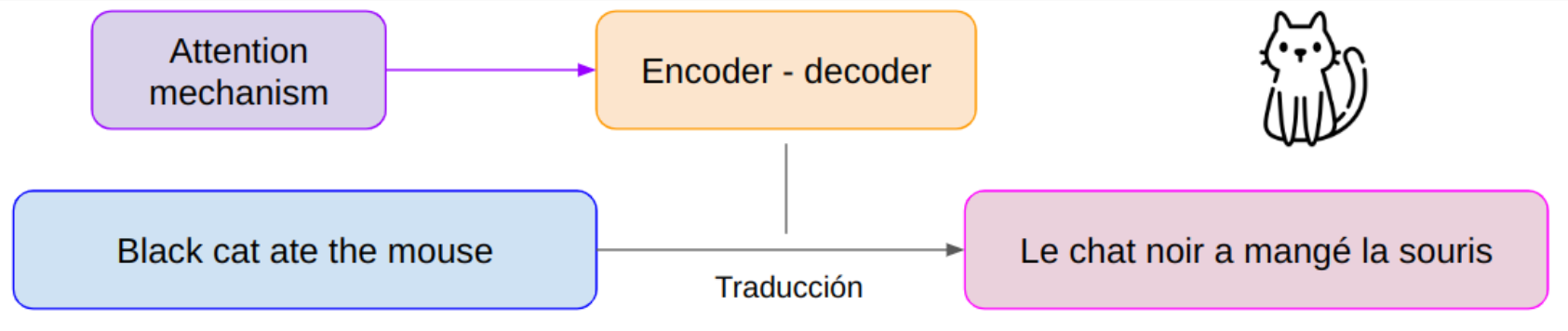
NEURAL MACHINE TRANSLATION
BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau
Jacobs University Bremen, Germany

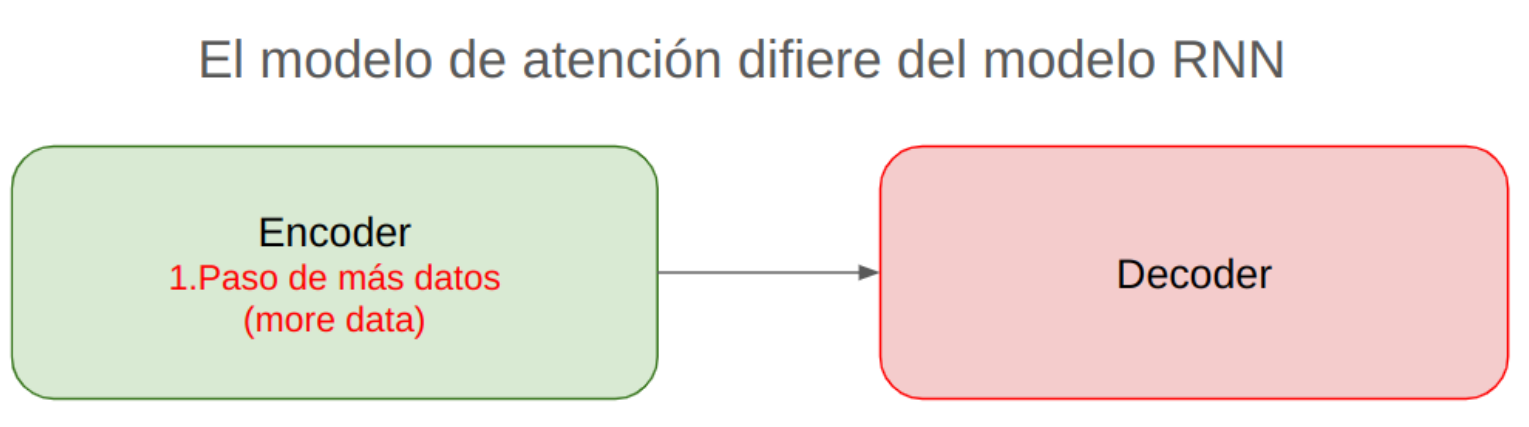
KyungHyun Cho **Yoshua Bengio***
Université de Montréal

Presentan una solución al problema de dependencias a largo plazo en el 2015.

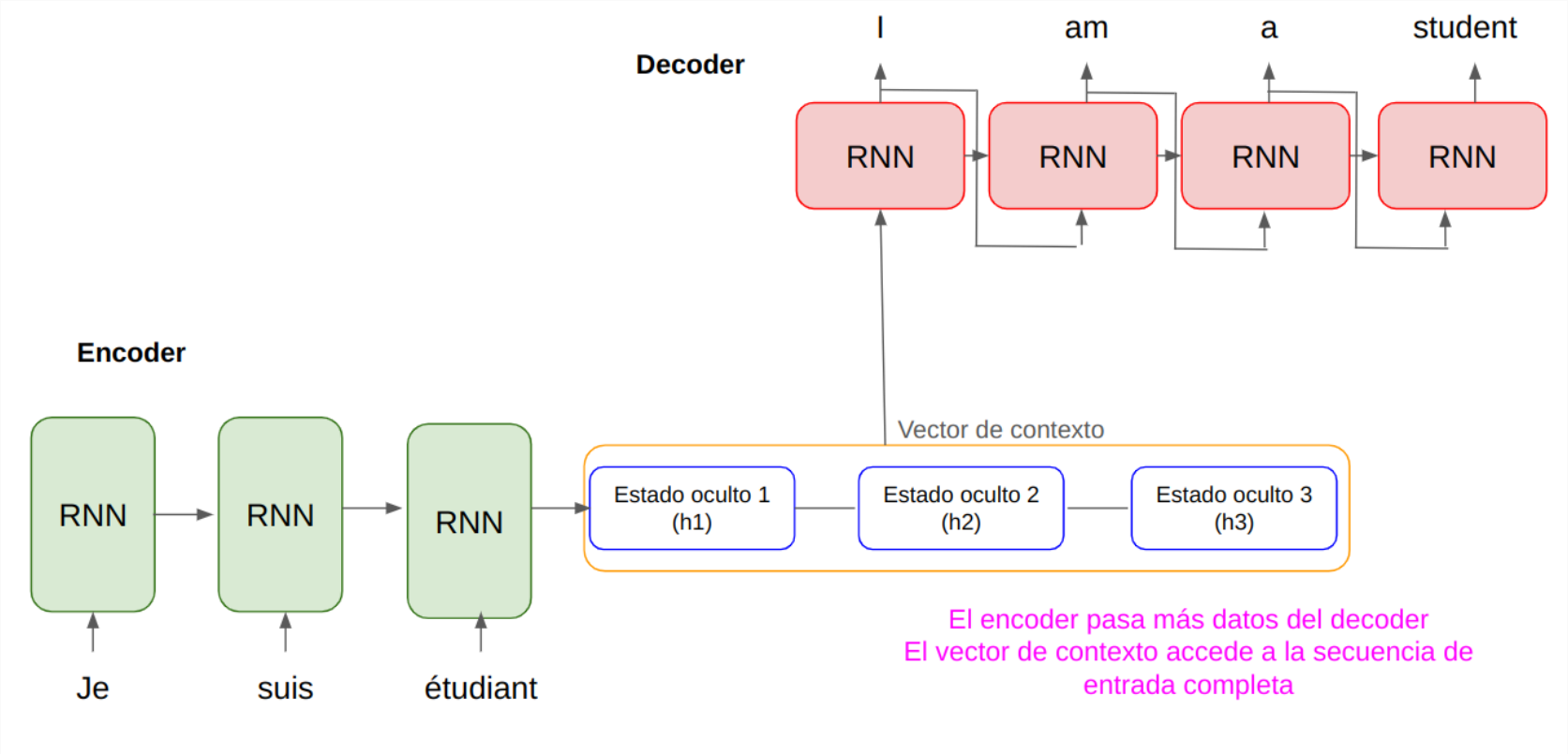
Introducción al mecanismo de atención



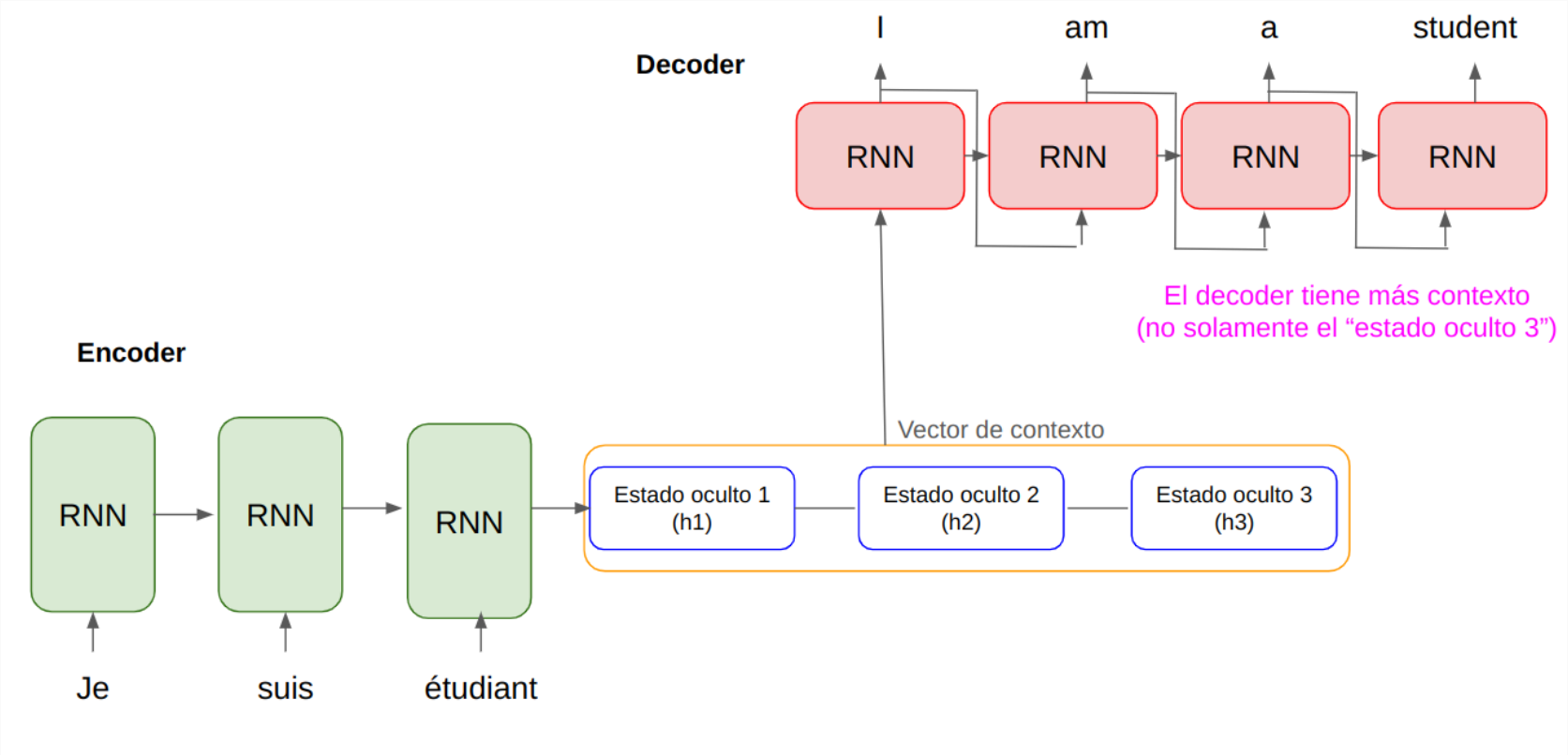
Diferencias entre el modelo RNN y el mecanismo de atención



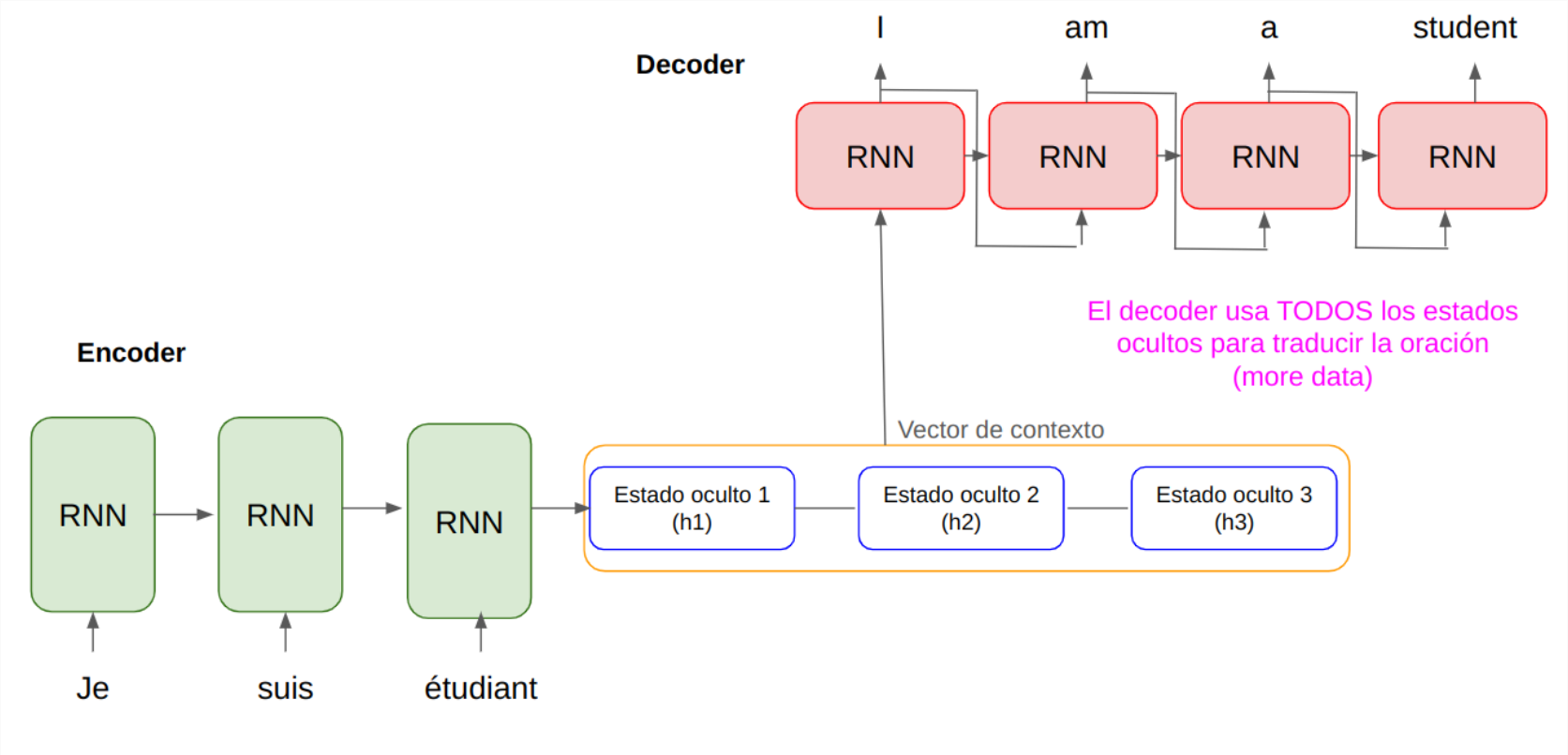
Encoder envía más datos



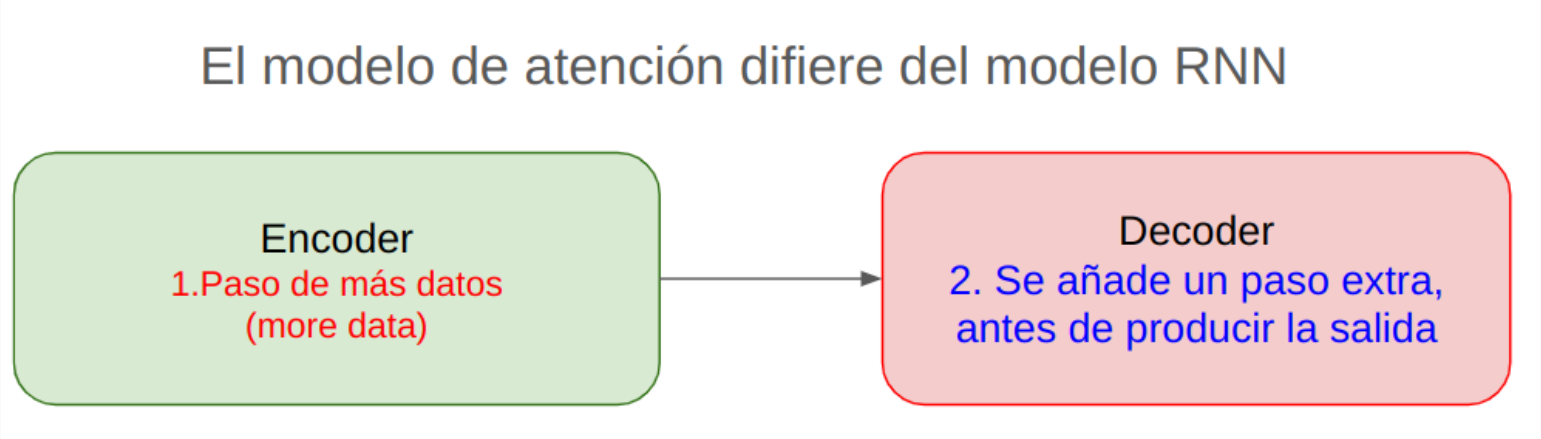
Decoder recibe más contexto



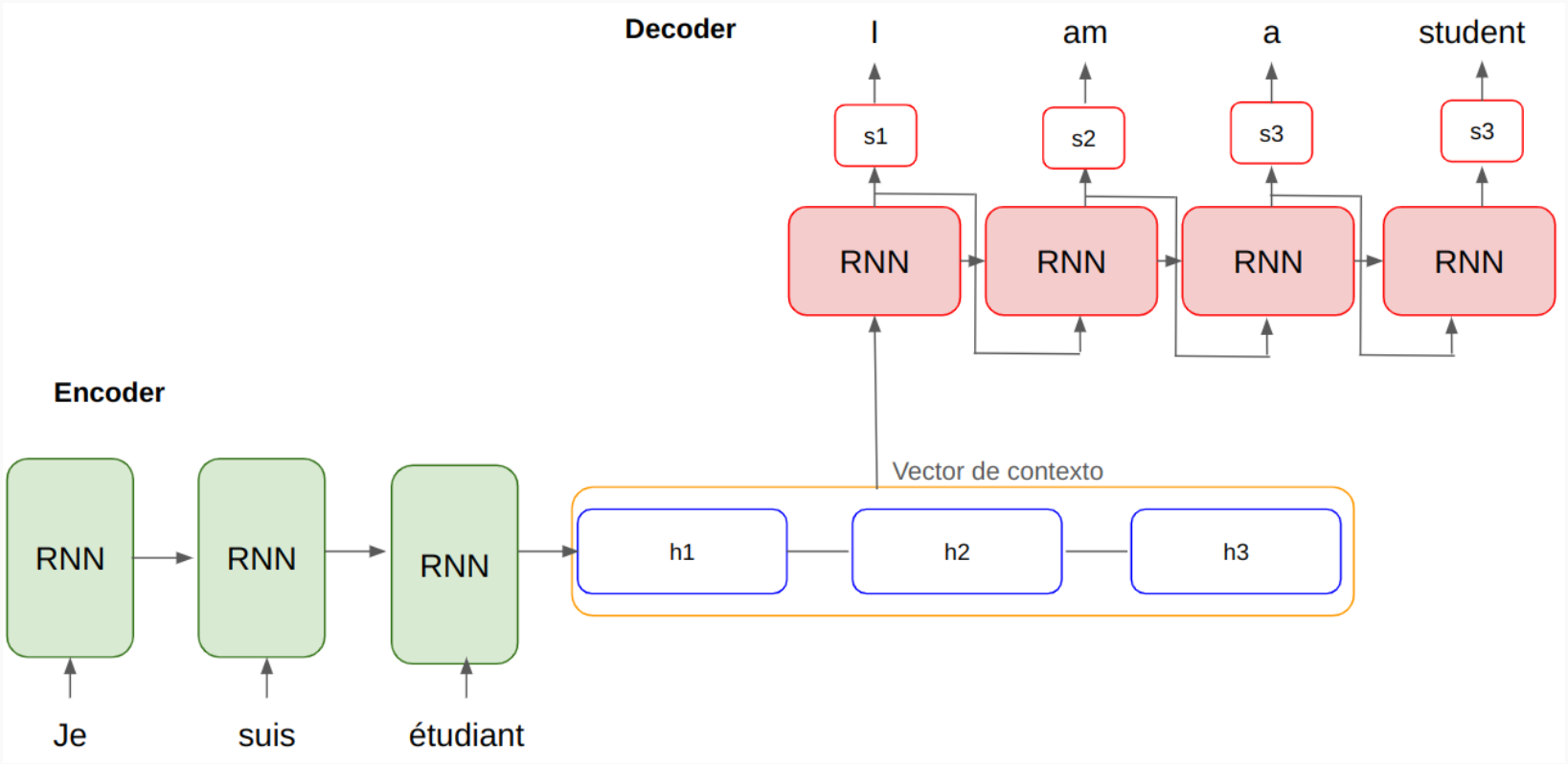
Decoder recibe más contexto



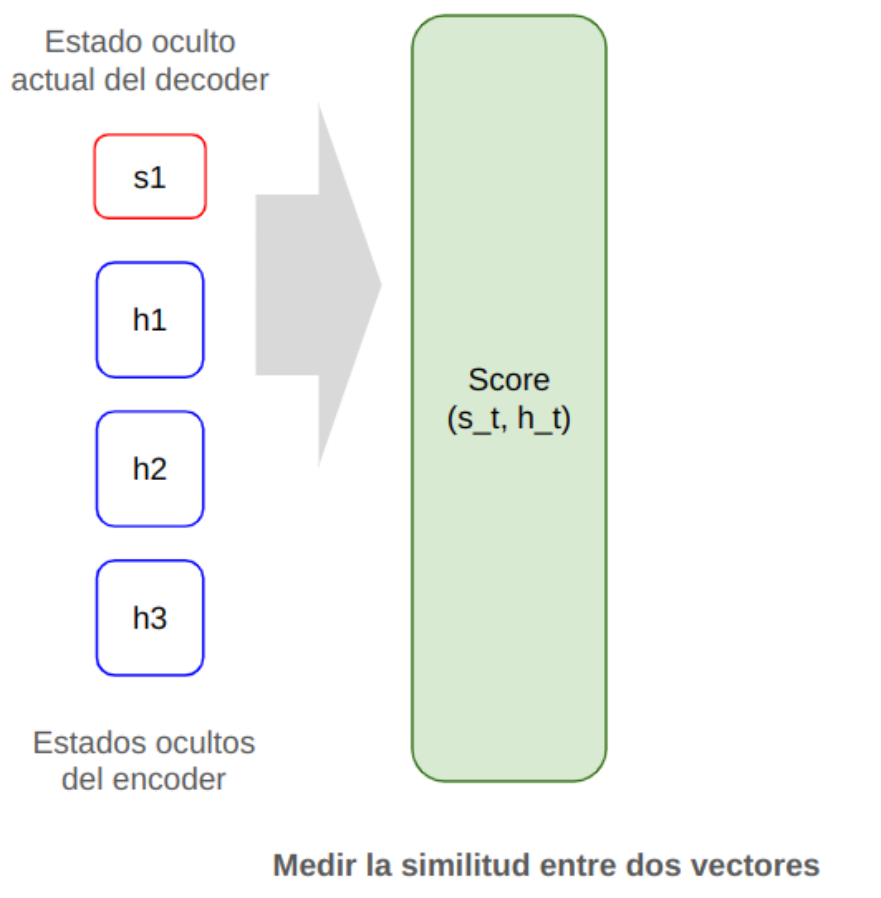
Diferencias entre el modelo RNN y el mecanismo de atención



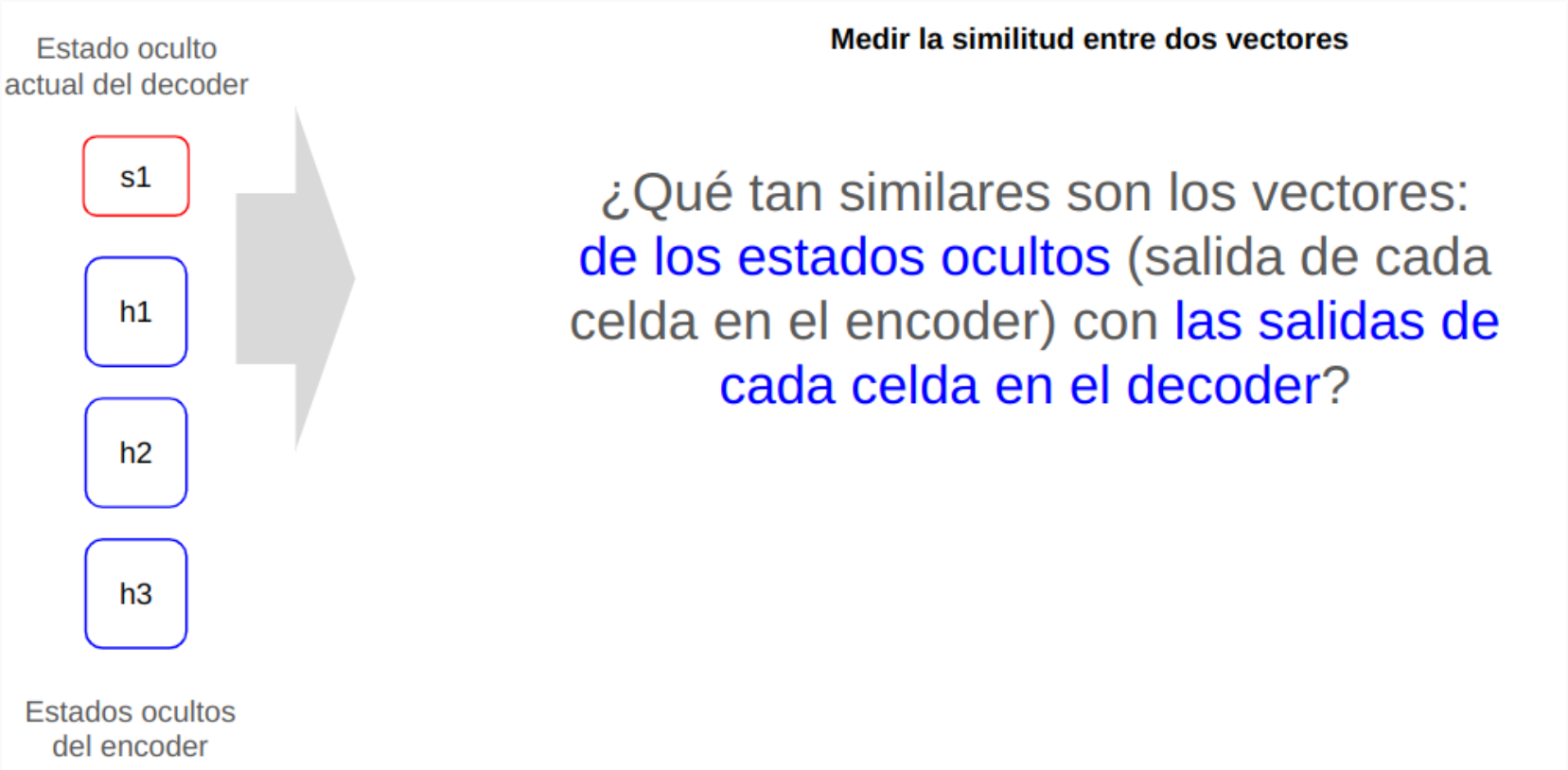
¿Cómo pasar mayor contexto del encoder al decoder



Mecanismo de atención: paso 1



Paso 1: medir similitud



Paso 1: medir similitud

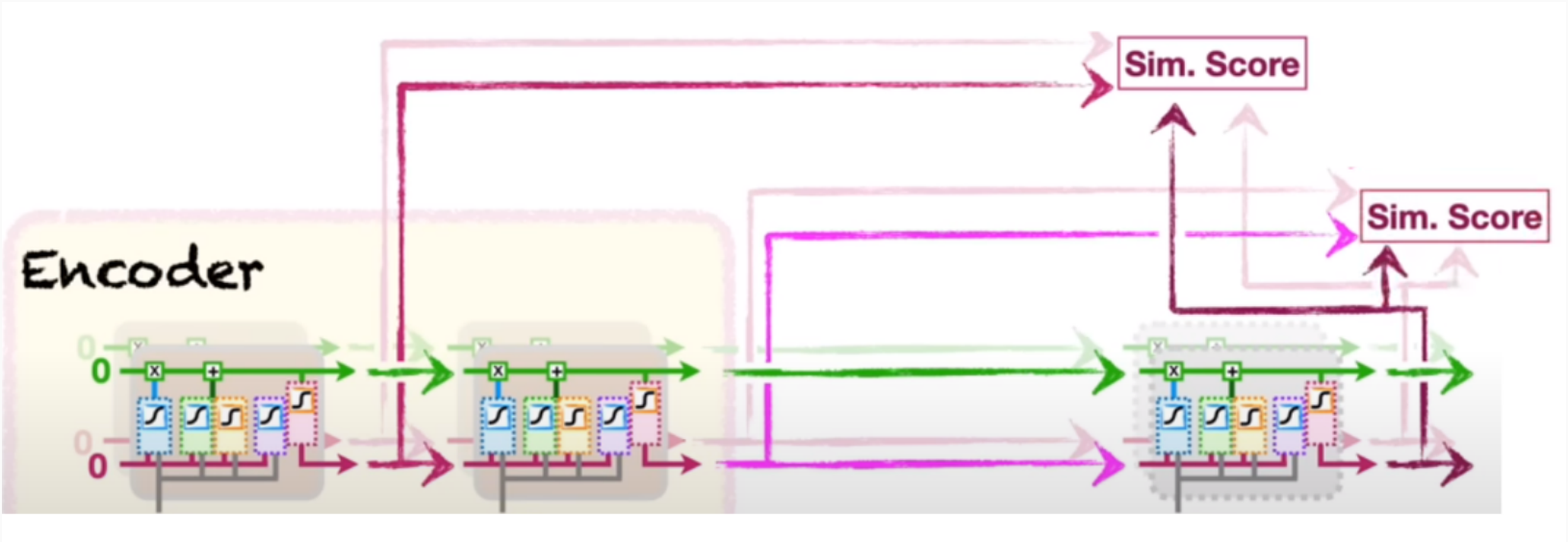


Imagen tomada de StatQuest, 2022.

Similitud coseno

$$Similitud\_coseno = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Similitud coseno

$$Similitud\_coseno = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

- El numerador calcula la similitud entre los 2 vectores.

Similitud coseno

$$Similitud\_coseno = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

- El numerador calcula la similitud entre los 2 vectores.
- El denominador escala los valores entre -1 y 1

Similitud coseno: matemáticas extras

$$Similitud\_coseno = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Similitud coseno: matemáticas extras

$$Similitud\_coseno = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

- Dado que el denominador, únicamente escala la magnitud del score de similitud entre -1 y 1, entonces..

Similitud coseno

$$Similitud\_coseno = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Se calcula únicamente el numerador

Producto punto (vista general)

$$Similitud\_coseno = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Producto punto
(dot product)

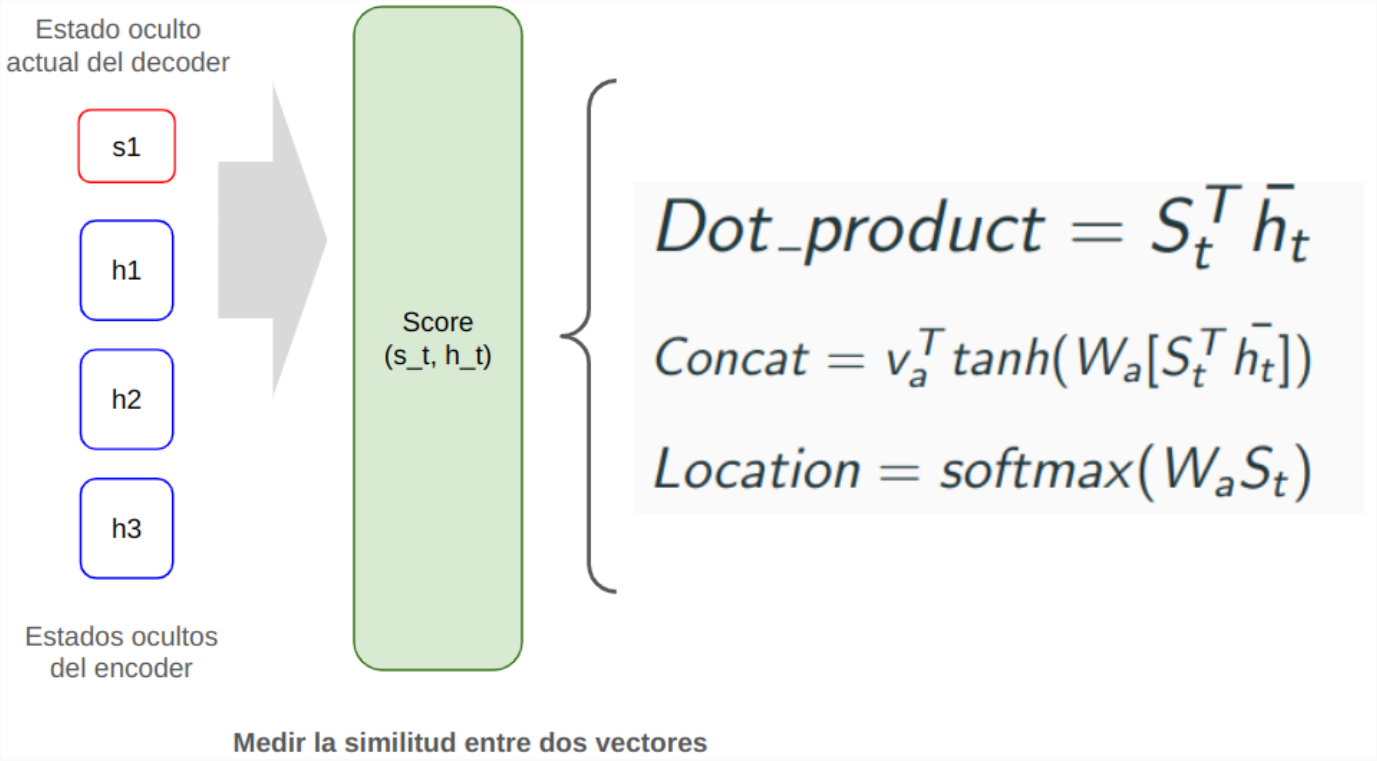
Producto punto

$$Dot\_product = \sum_{i=1}^{n} A_i B_i$$

Calcular el producto punto es más común que la similitud coseno:

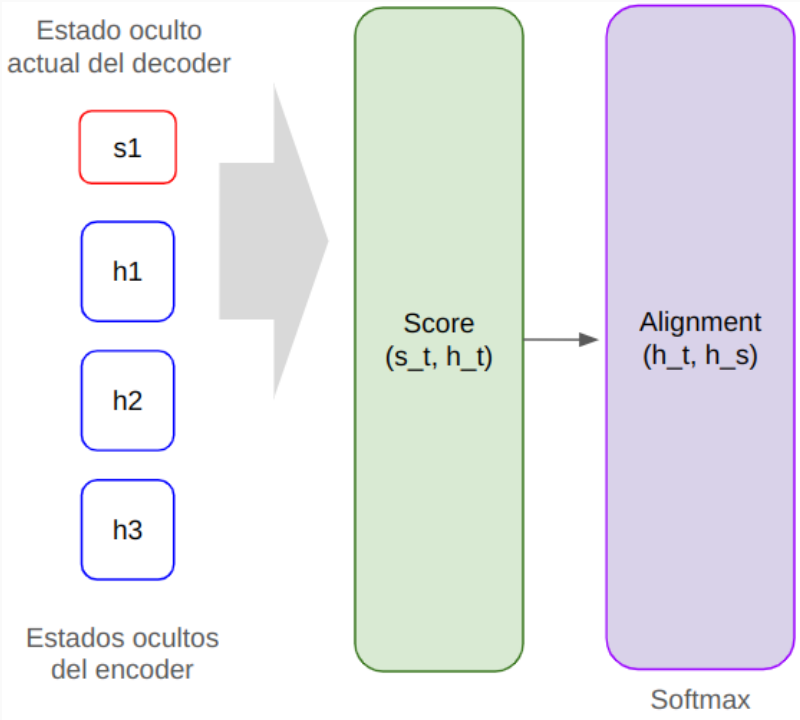
- Es fácil de calcular.
- Un número positivo grande (*large positive*) significa mayor similitud que un número positivo pequeño.
- Un número negativo grande (*large negative*) significa menos similitud que un número negativo pequeño.

Paso 1: medir la similitud



Medir la similitud entre dos vectores

Paso 2: alineación



$$a_t(s) = \frac{\exp(Score(h_t, \overline{h_s}))}{\sum_{s'} \exp(Score(h_t, \overline{h_{s'}}))}$$

La idea es obtener las probabilidades

Aplicar la función softmax

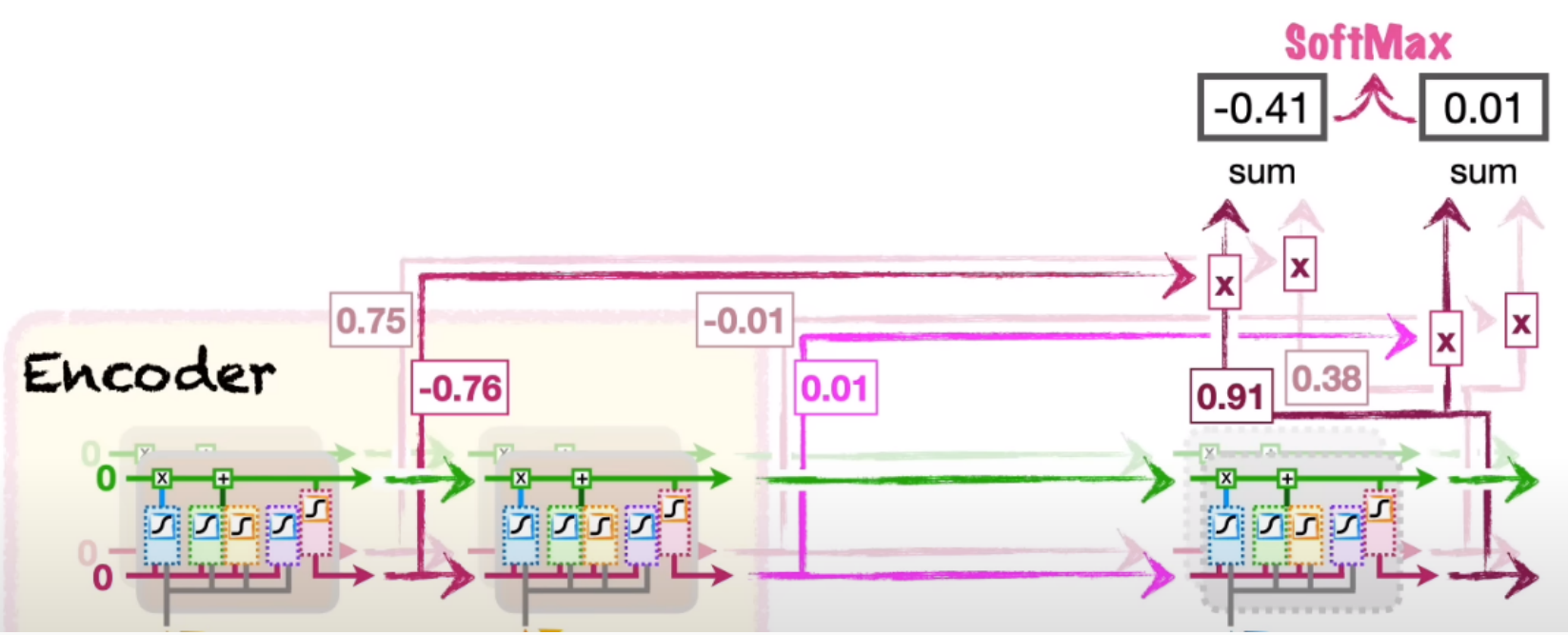


Imagen tomada de StatQuest, 2022.

Aplicar la función softmax

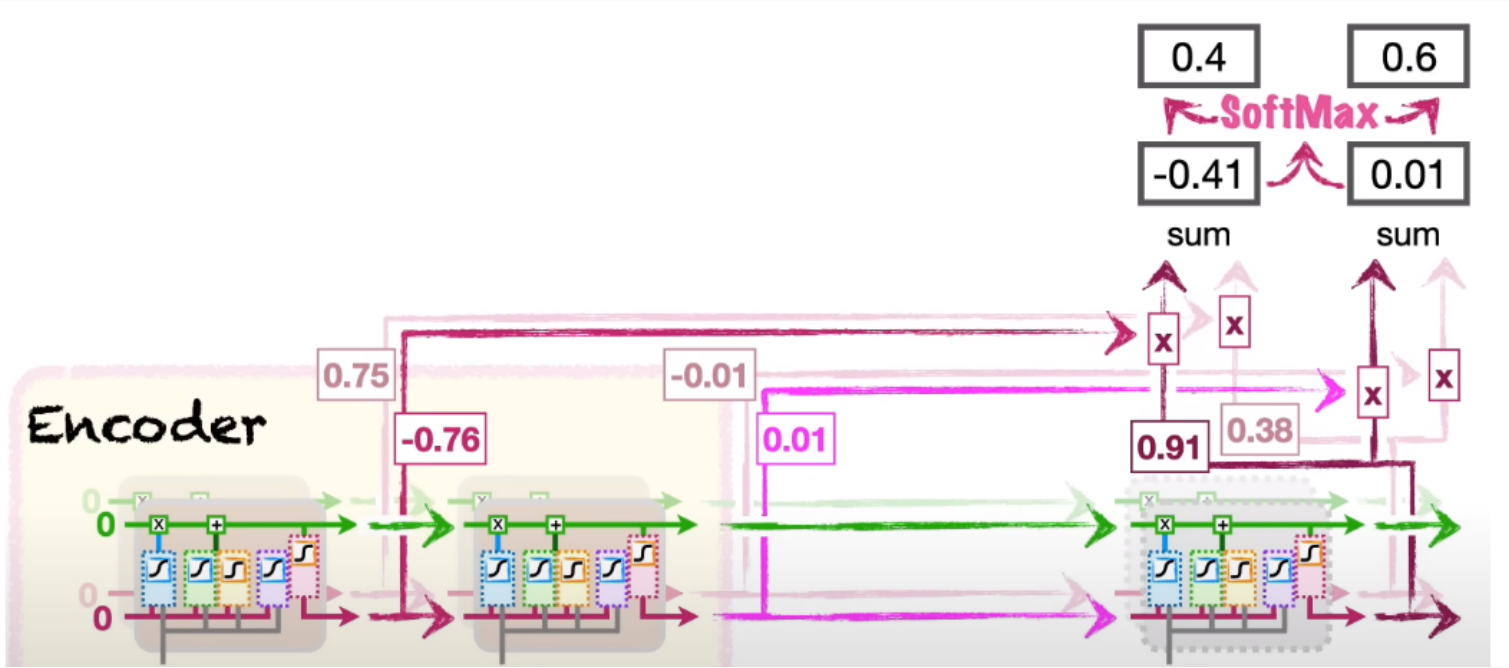


Imagen tomada de StatQuest, 2022.

Indica el porcentaje en qué se debe usar cada palabra en la decodificación

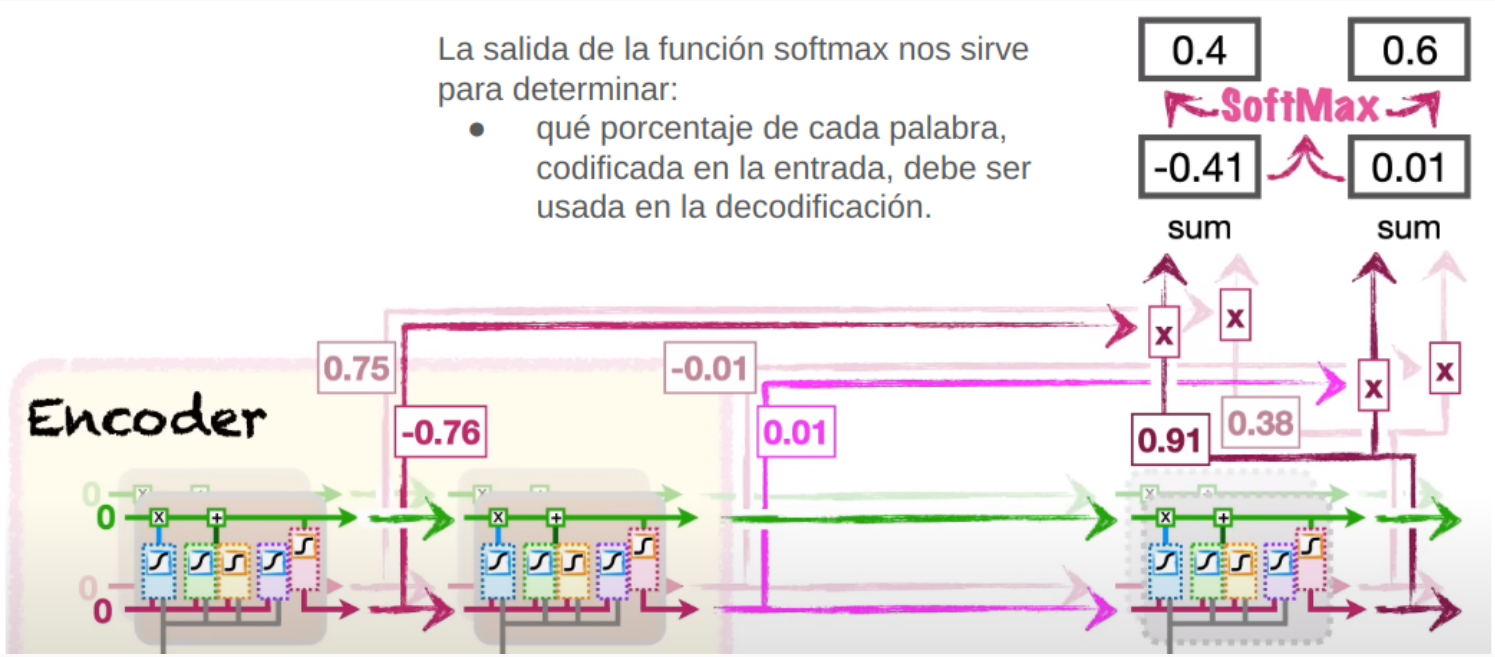
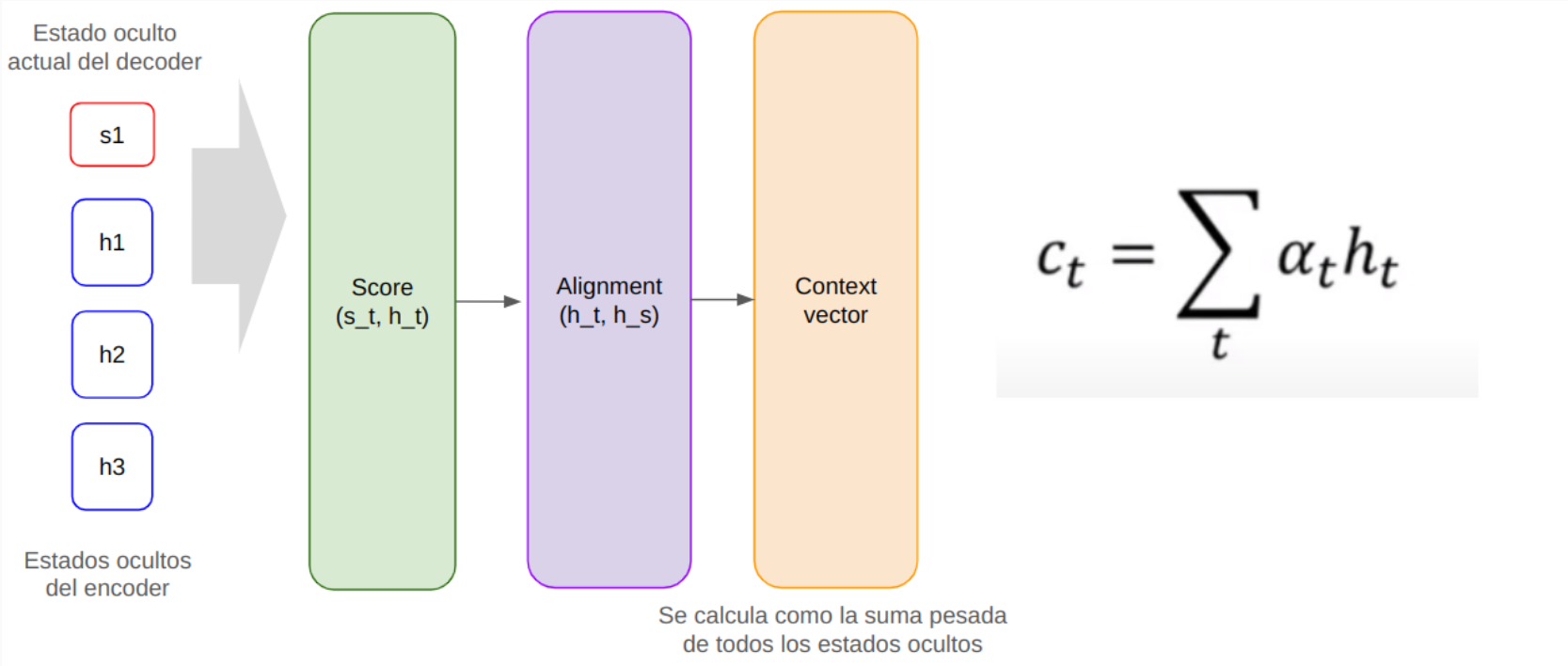


Imagen tomada de StatQuest, 2022.

Paso 3: vector de contexto



$$c_t = \sum_t \alpha_t h_t$$

En resumen

En el mecanismo de atención:

- El encoder pasa más datos para la generación de la siguiente palabra.
- El decoder tiene acceso a cada paso individual en el encoder.
- Se calcula un puntaje de similitud.
- Usando la función softmax se determina qué porcentaje de cada palabra codificada debe ser usada para predecir la siguiente palabra.

Vista general del mecanismo de atención

El modelo de atención es como un libro de contabilidad

Query

S_{j-1} →

Table

Keys	Value
h_1	0.4
h_2	0.6
h_3	0.1

Diagrama del mecanismo de atención

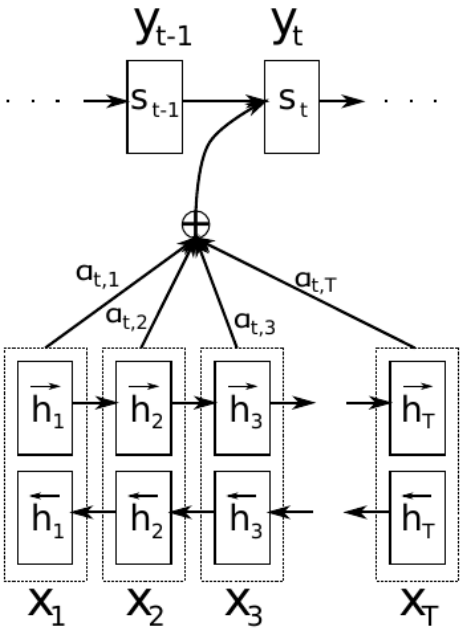


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

Imagen tomada de Bahdanau et al, 2015.

Comparativa de resultados

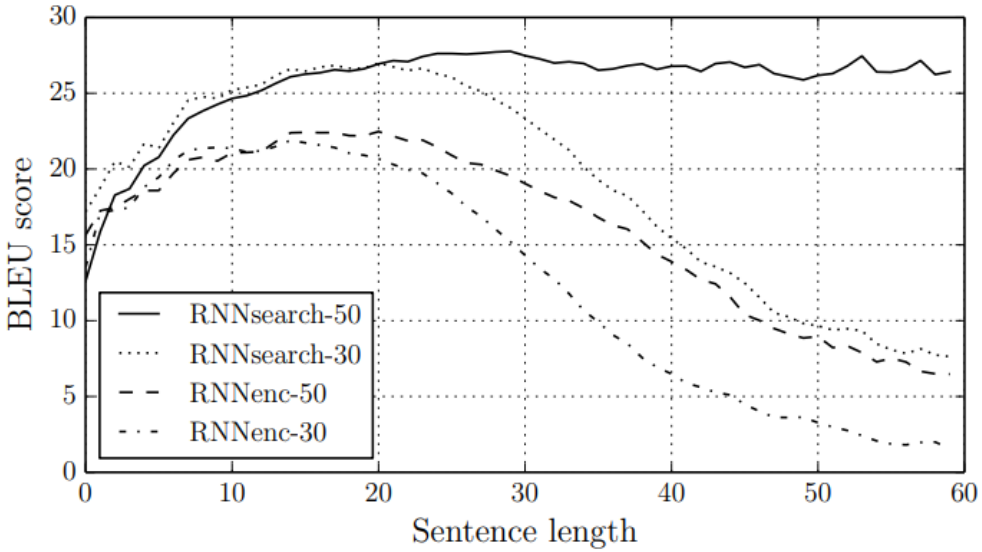


Figure 2: The BLEU scores of the generated translations on the test set with respect to the lengths of the sentences. The results are on the full test set which includes sentences having unknown words to the models.

Imagen tomada de Bahdanau et al, 2015.

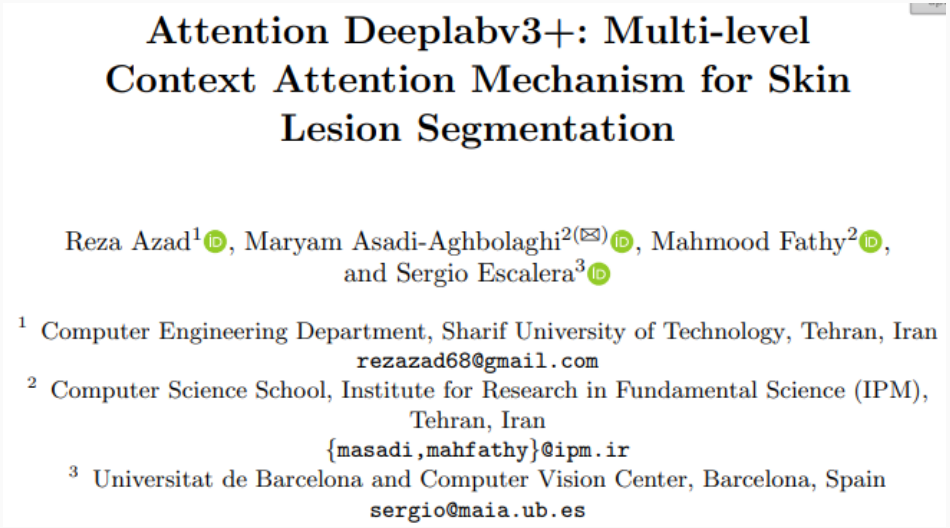
Tiempo de ejecución

Model	Updates ($\times 10^5$)	Epochs	Hours	GPU	Train NLL	Dev. NLL
RNNenc-30	8.46	6.4	109	TITAN BLACK	28.1	53.0
RNNenc-50	6.00	4.5	108	Quadro K-6000	44.0	43.6
RNNsearch-30	4.71	3.6	113	TITAN BLACK	26.7	47.2
RNNsearch-50	2.88	2.2	111	Quadro K-6000	40.7	38.1
RNNsearch-50*	6.67	5.0	252	Quadro K-6000	36.7	35.2

Table 2: Learning statistics and relevant information. Each update corresponds to updating the parameters once using a single minibatch. One epoch is one pass through the training set. NLL is the average conditional log-probabilities of the sentences in either the training set or the development set. Note that the lengths of the sentences differ.

Imagen tomada de Bahdanau et al, 2015.

Aplicaciones



Attention Deeplabv3+: Multi-level Context Attention Mechanism for Skin Lesion Segmentation

Reza Azad[1], Maryam Asadi-Aghbolaghi[2(✉)], Mahmood Fathy[2], and Sergio Escalera[3]

[1] Computer Engineering Department, Sharif University of Technology, Tehran, Iran
rezazad68@gmail.com
[2] Computer Science School, Institute for Research in Fundamental Science (IPM), Tehran, Iran
{masadi,mahfathy}@ipm.ir
[3] Universitat de Barcelona and Computer Vision Center, Barcelona, Spain
sergio@maia.ub.es

Attention Deeplabv3+

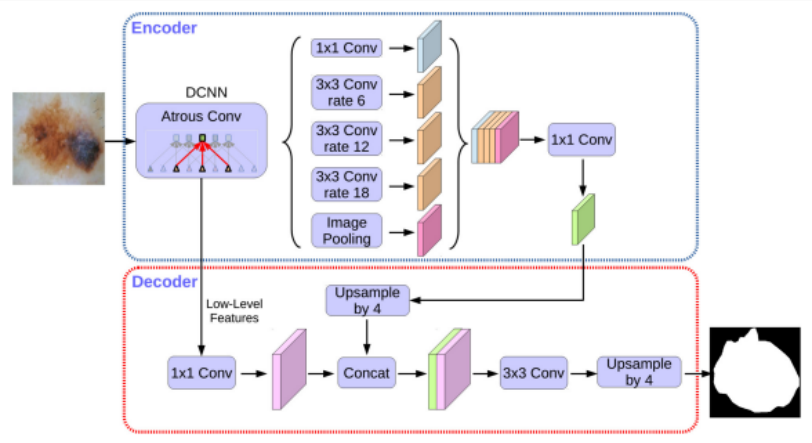


Imagen tomada de Azad, 2020.

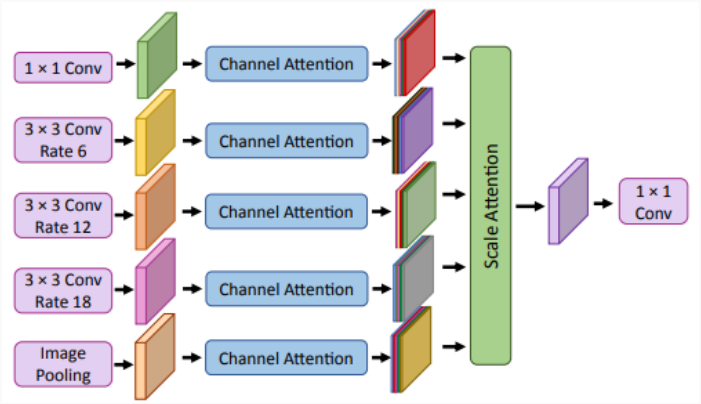


Imagen tomada de Azad, 2020.

Rendimiento: Attention Deeplabv3+

Table 1. Performance comparison of the proposed network and other methods on ISIC 2017.

Methods	F1-Score	Sensitivity	Specificity	Accuracy	Jaccard similarity
U-net [27]	0.8682	0.9479	0.9263	0.9314	0.9314
Melanoma det. [13]	–	–	–	0.9340	–
Lesion analysis [18]	–	0.8250	0.9750	0.9340	–
R2U-net [3]	0.8920	**0.9414**	0.9425	0.9424	0.9421
BCDU-Net [5]	0.8810	0.8647	0.9751	0.9528	0.9528
MCGU-Net [4]	0.8927	0.8502	0.9855	0.9570	0.9570
Deeplabv3+ [9]	0.9162	0.8733	**0.9921**	0.9691	0.9691
Att-Deeplabv3+	**0.9190**	0.8851	0.9901	**0.9698**	**0.9698**

Imagen tomada de Azad, 2020.

Rendimiento: Attention Deeplabv3+

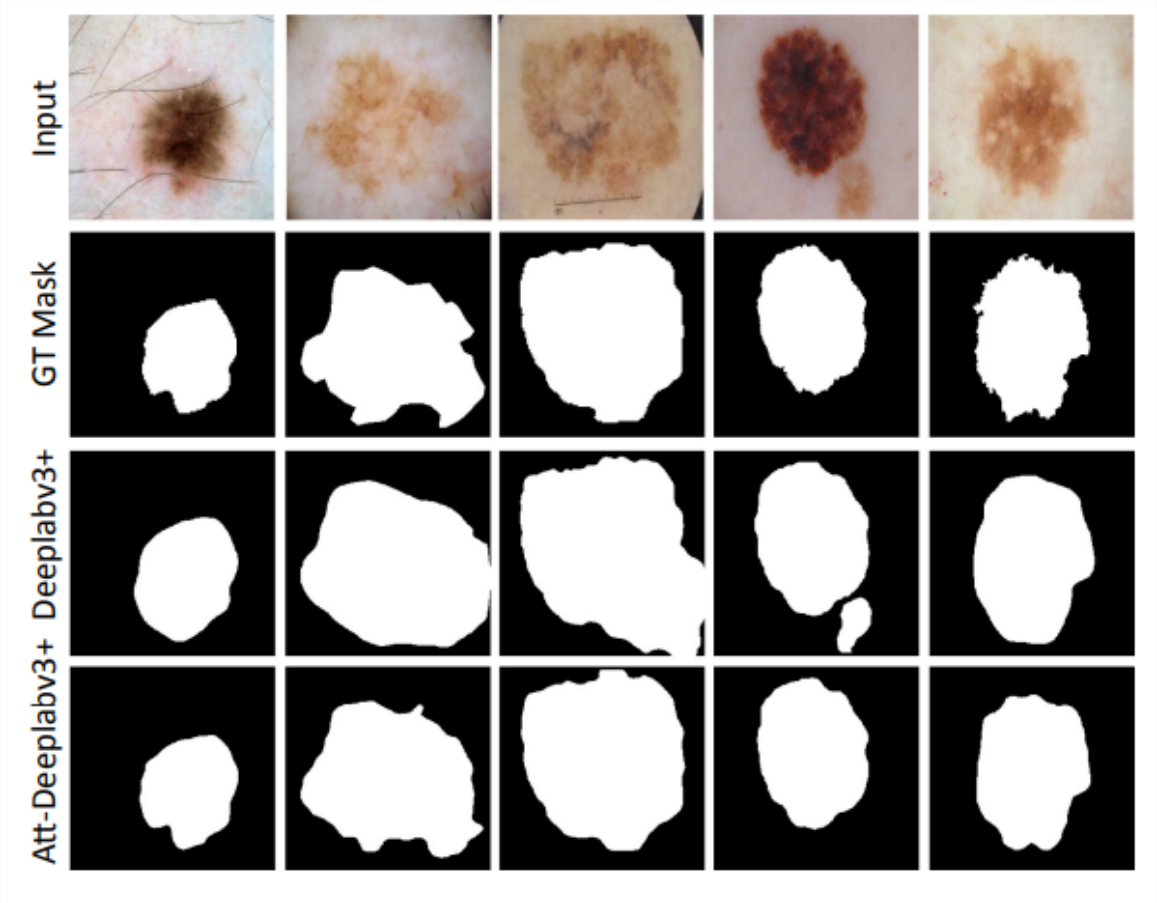


Imagen tomada de Azad, 2020.

Aplicaciones

Attention mechanism-based CNN for facial expression recognition

Jing Li [a], Kan Jin [a], Dalin Zhou [b], Naoyuki Kubota [c], Zhaojie Ju [b,*]

[a] School of Information Engineering, Nanchang University, Nanchang 330031, China
[b] School of Computing, University of Portsmouth, Portsmouth PO1 3HE, UK
[c] Graduate School of Systems Design, Tokyo Metropolitan University, China

Attention mechanism-based CNN for facial expression recognition

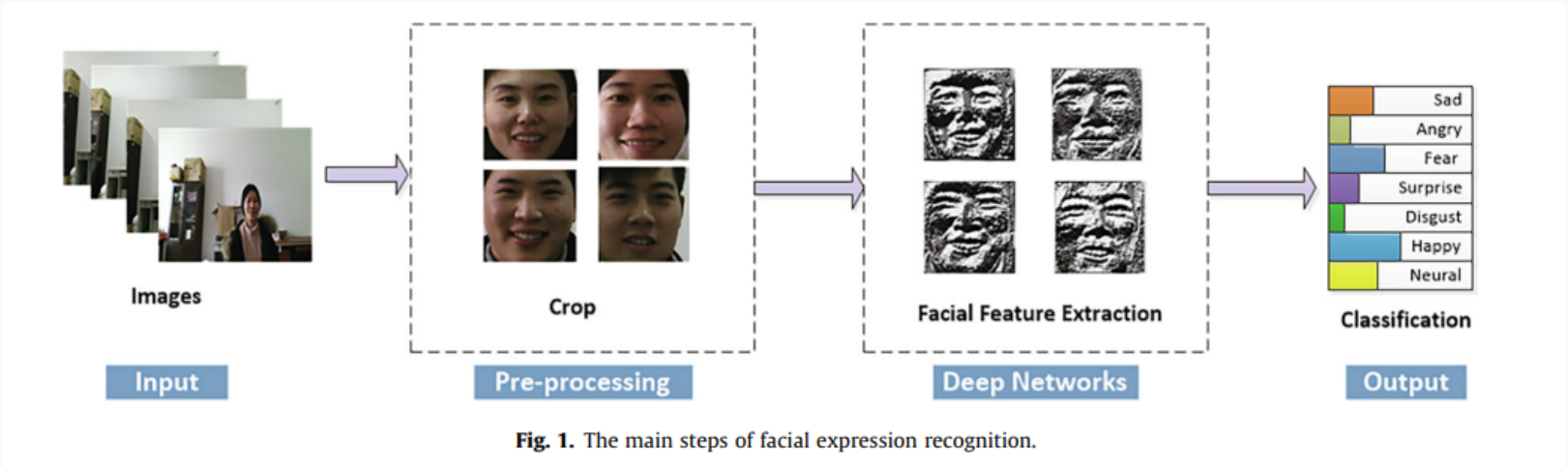


Fig. 1. The main steps of facial expression recognition.

Imagen tomada de Li, 2020.

Attention mechanism-based CNN for facial expression recognition

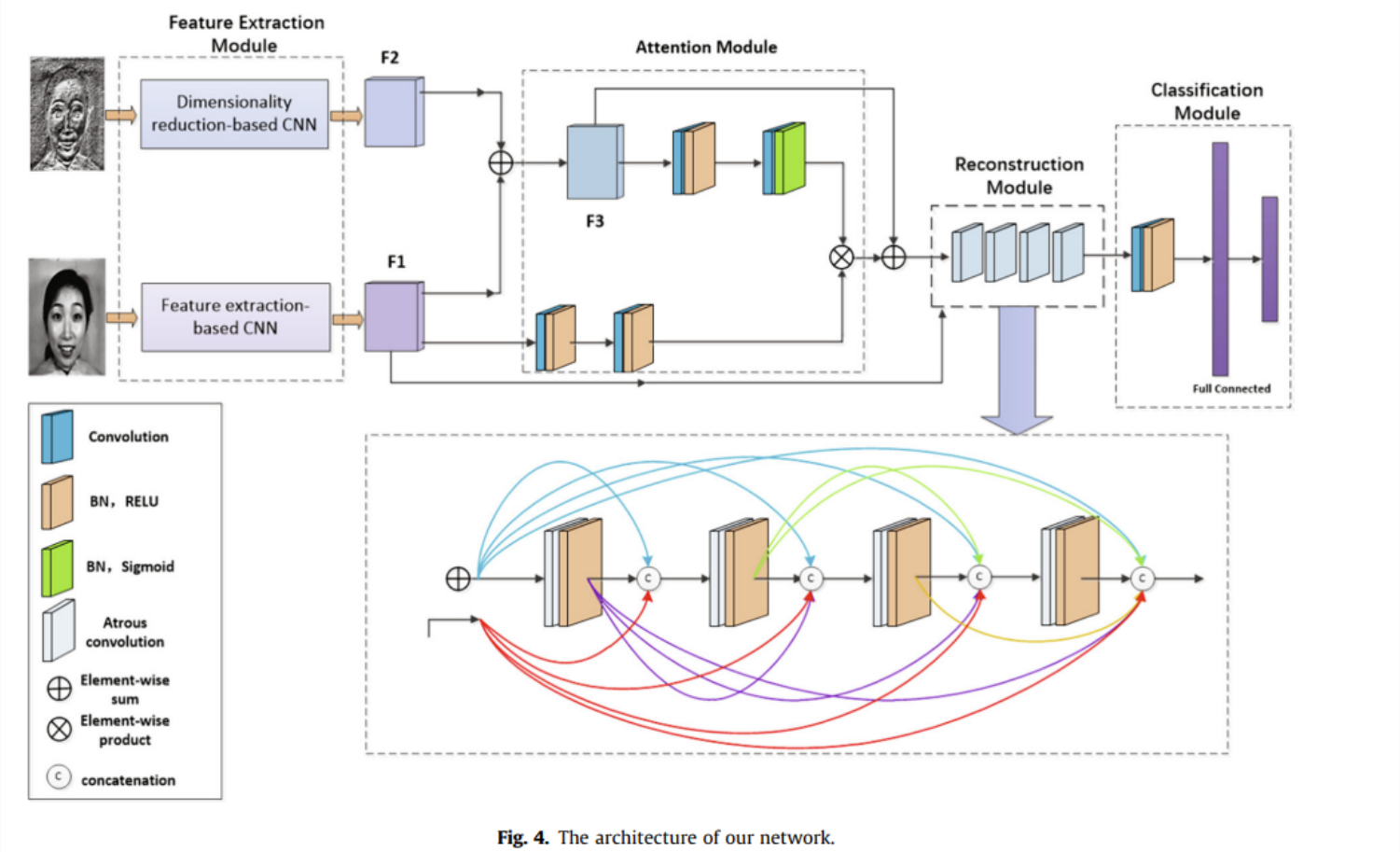


Fig. 4. The architecture of our network.

Imagen tomada de Li, 2020.

Attention mechanism-based CNN for facial expression recognition

Fig. 6. Results of attention heatmap.

Imagen tomada de Li, 2020.

Comparison of different methods on NCUFE dataset.

Method	Recognition Rates (%)
Li et al. [34]	91.58
Arriaga et al. [29]	81.3
Light-CNN. [30]	82.5
Ours	**94.33**
Ours (Without LBP)	93.91
Ours (Without attention module)	92.07
Ours (raw + depth)	90.35

Imagen tomada de Li, 2020.

Attention Is All You Need

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

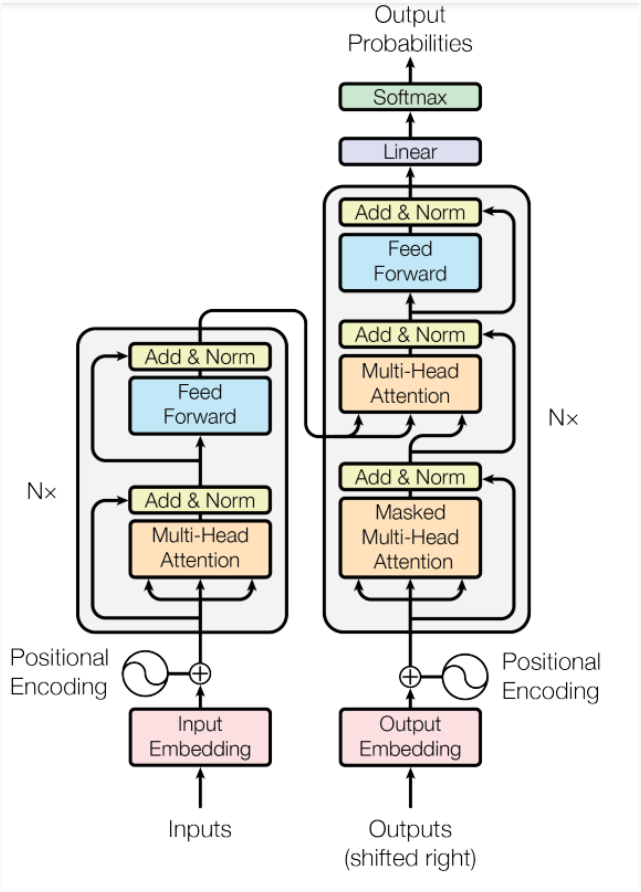
Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Attention Is All You Need



Arquitectura del modelo *Transformer*

Imagen tomada de Vaswani et al, 2017.

Attention Is All You Need

- La arquitectura *Transformer* evita las recurrencias.
- Se basa únicamente en mecanismos de atención para generar dependencias globales entre entrada y salida.

Principal ventaja

- Paralelización.
- Permite el entrenamiento en secuencias muy largas.
- Se caracteriza por tener un módulo denominado *multi-headed attention*.

Encoder: inputs

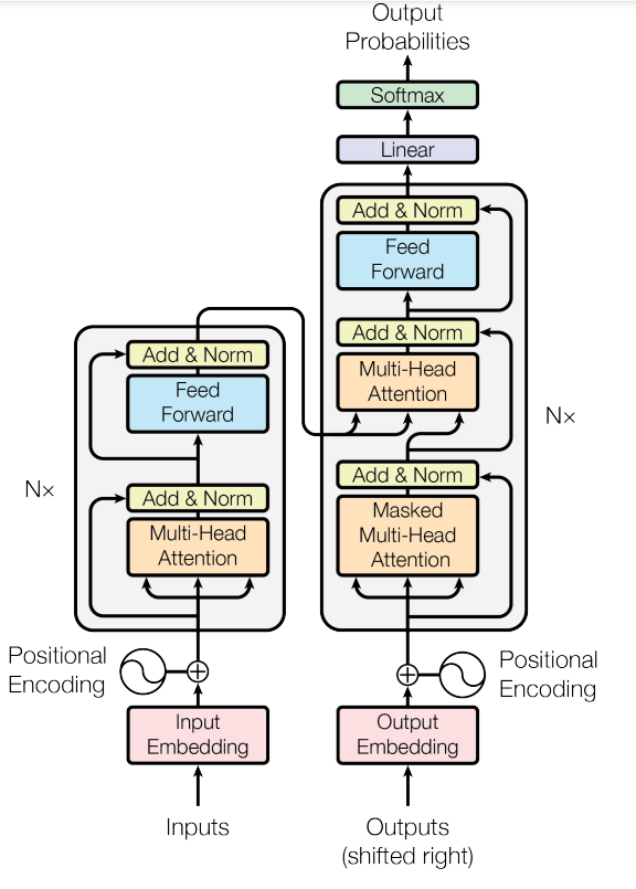


Imagen tomada de Vaswani et al, 2017.

- Inputs (pre-procesamiento)
  - Texto
  - Índice del vocabulario

Encoder: embedding layer

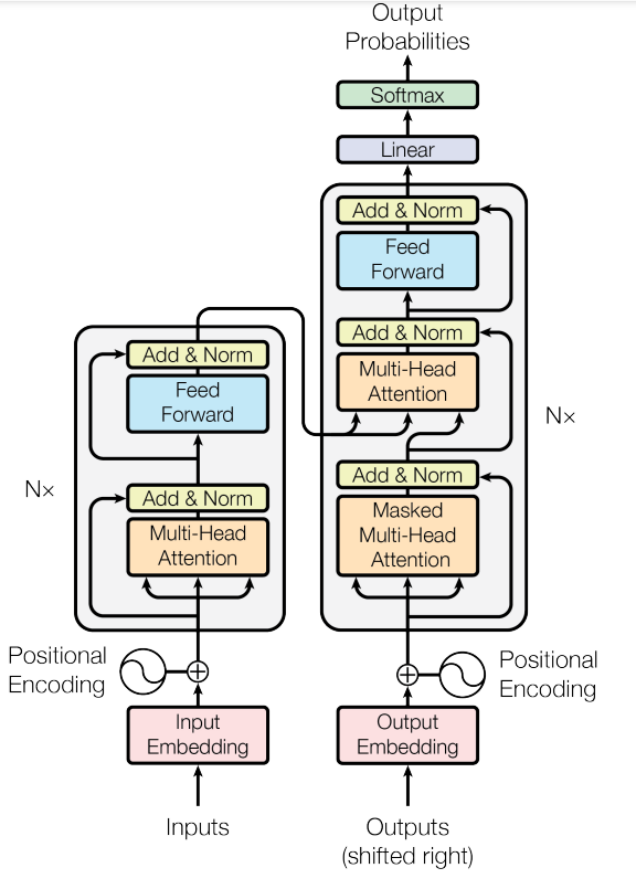
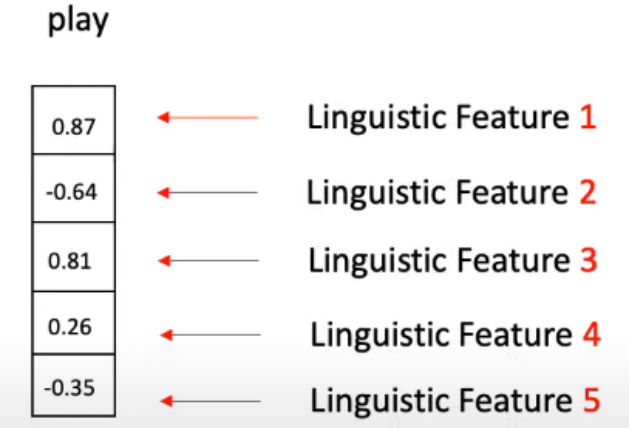


Imagen tomada de Vaswani et al, 2017.

- Input embedding
  - Toma como entrada los índices.
  - Los vectores se inicializan con números aleatorios.
  - Durante el entrenamiento, estos vectores se actualizan.
  - Embedding size $= 512$

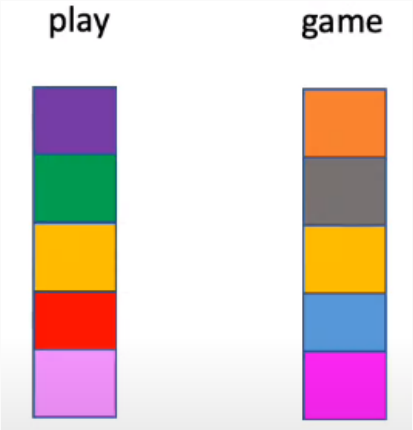
Embeddings

Los embeddings es una representación vectorial de una palabra. Donde cada dimensión de la palabra trata de capturar alguna característica lingüística.



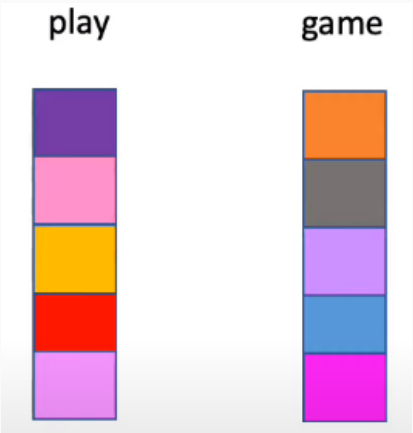
Embeddings: ejemplo

Los embeddings de las palabras ‘play’ y ‘game’ son inicializados de forma aleatoria.



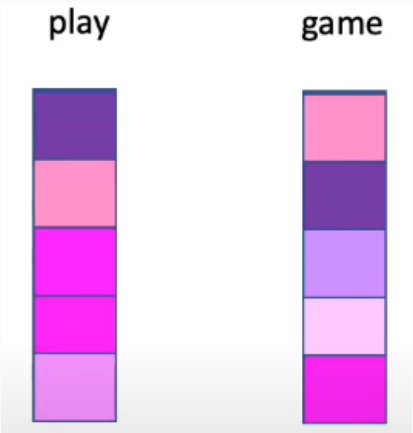
Embeddings: ejemplo

Pero durante el entrenamiento, comenzarán a parecer similares.



Embeddings: ejemplo

Pero durante el entrenamiento, comenzarán a parecer similares.



Visualizando las palabras

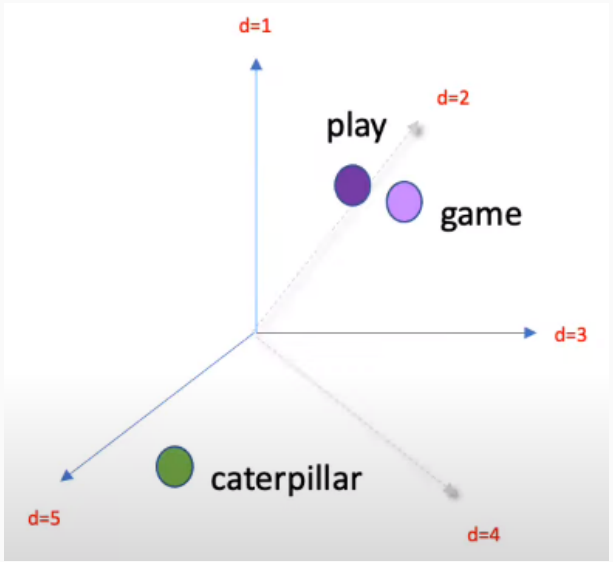


Imagen tomada de Haider, 2021.

Encoder: embedding layer

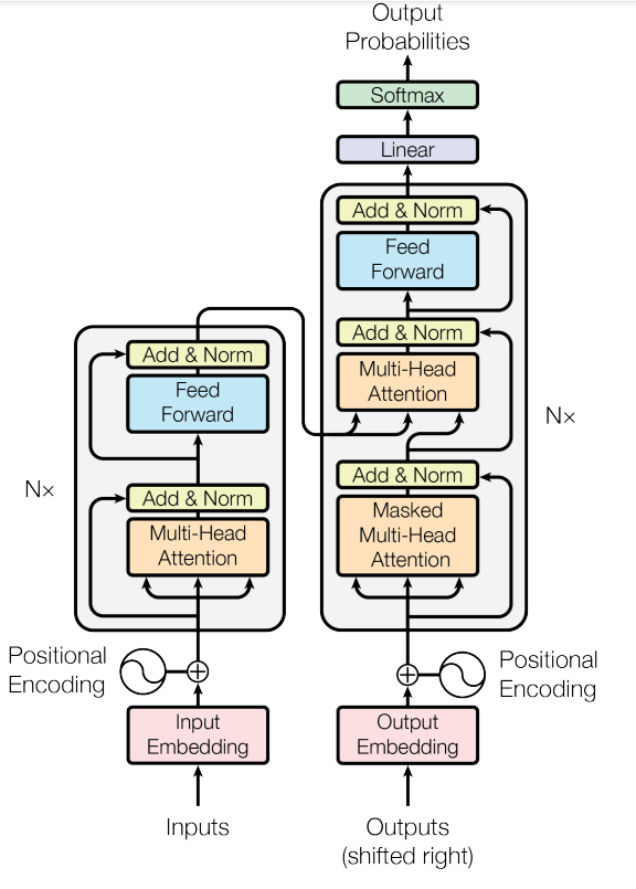


Imagen tomada de Vaswani et al, 2017.

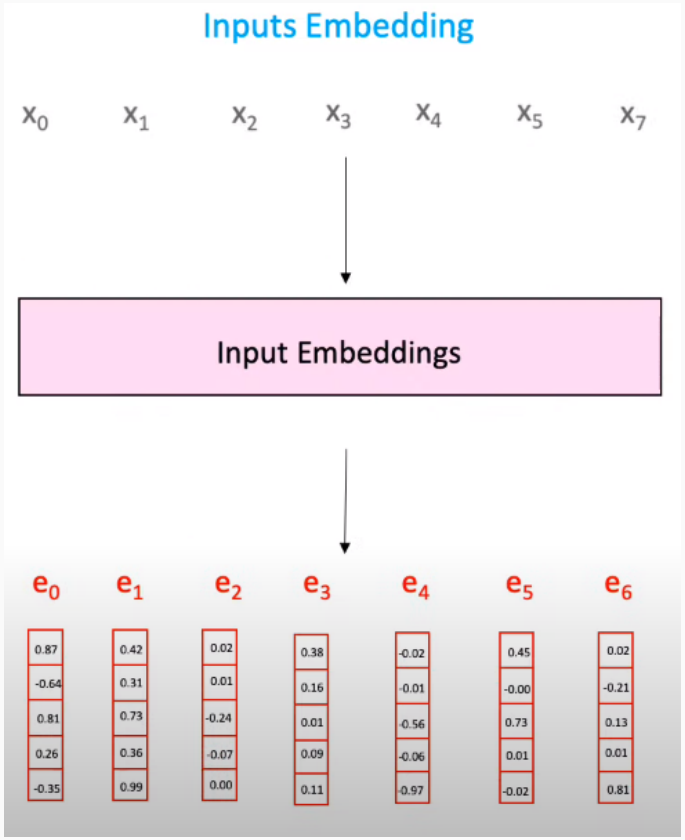


Imagen tomada de Haider, 2021.

Encoder: position embeddings

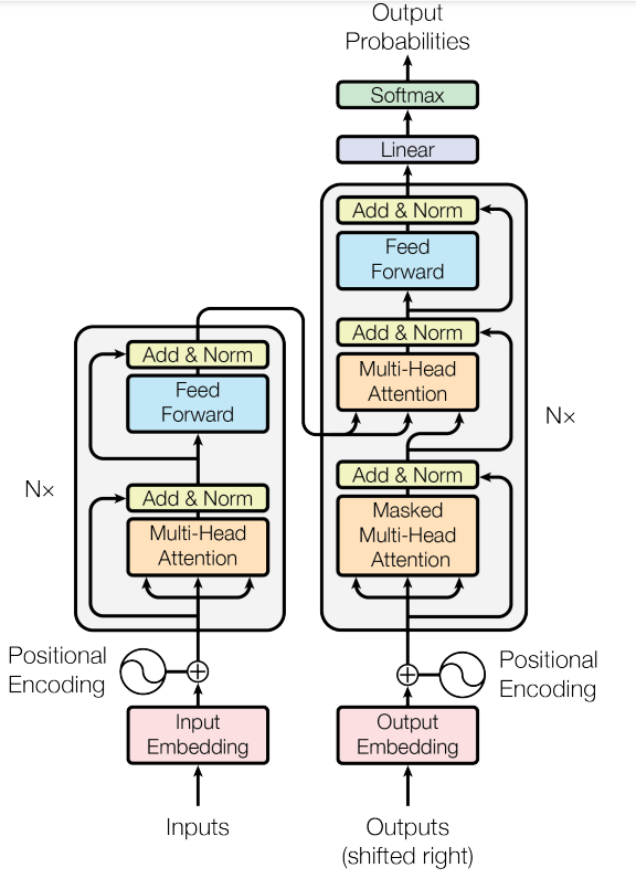


Imagen tomada de Vaswani et al, 2017.

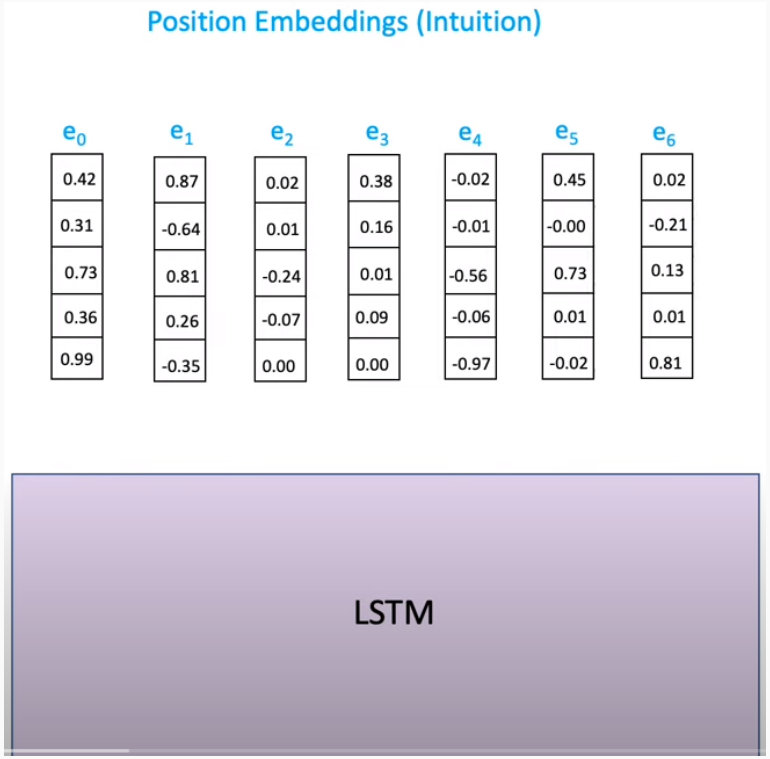


Imagen tomada de Haider, 2021.

Encoder: position embeddings

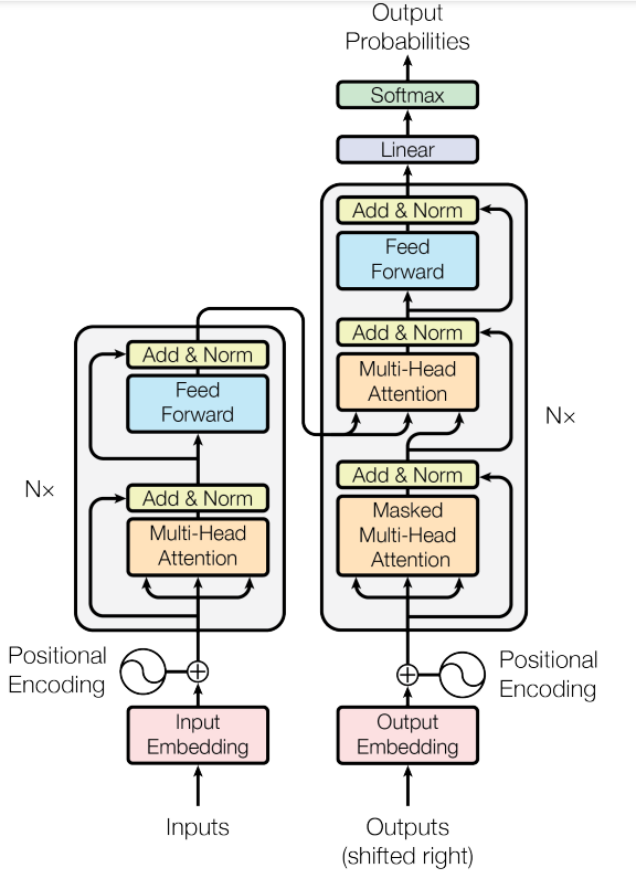


Imagen tomada de Vaswani et al, 2017.

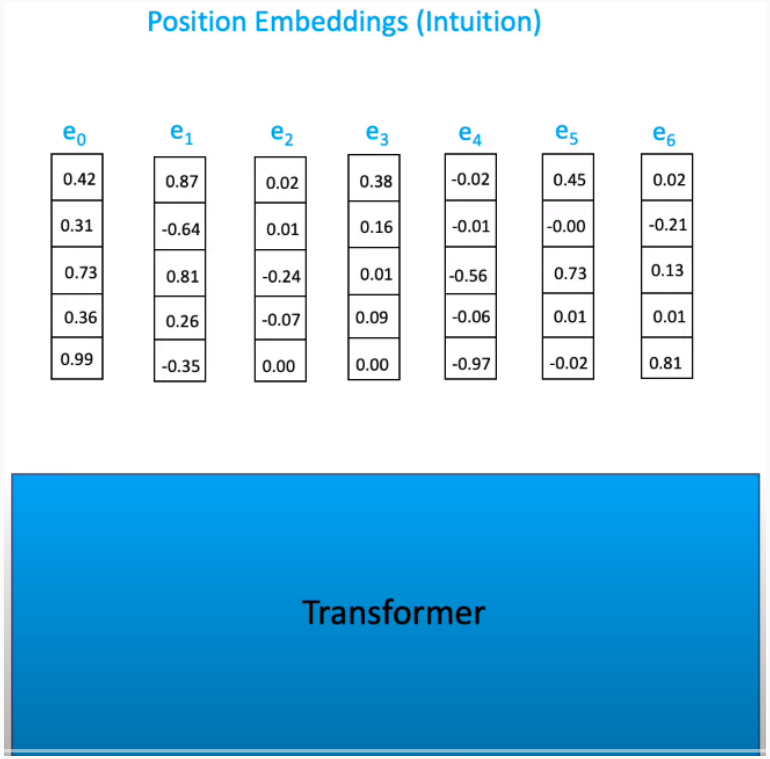


Imagen tomada de Haider, 2021.

Encoder: position embeddings

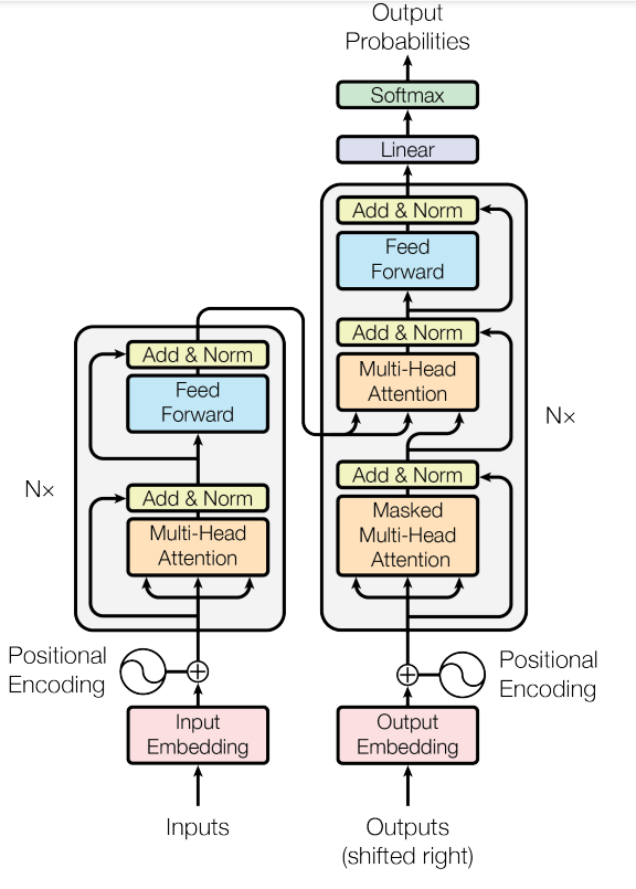


Imagen tomada de Vaswani et al, 2017.

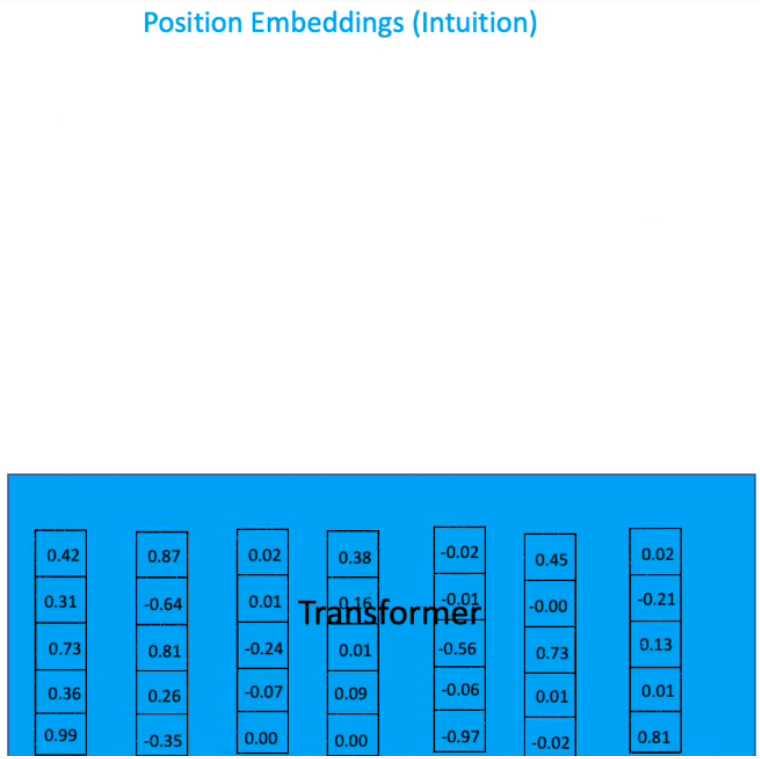


Imagen tomada de Haider, 2021.

Evitan la pérdida de información que pueda ocasionar llevar el orden de las palabras.

Encoder: position embeddings

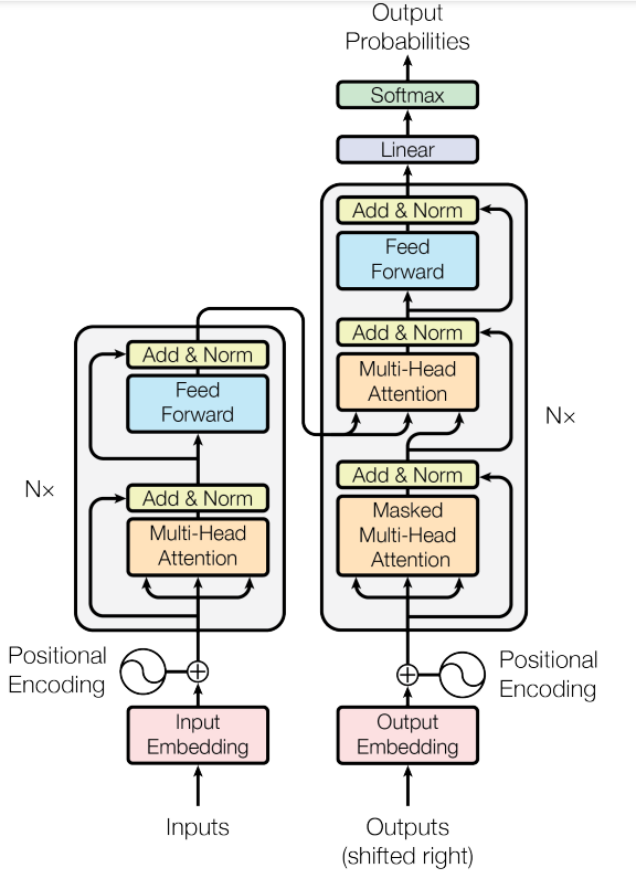


Imagen tomada de Vaswani et al, 2017.

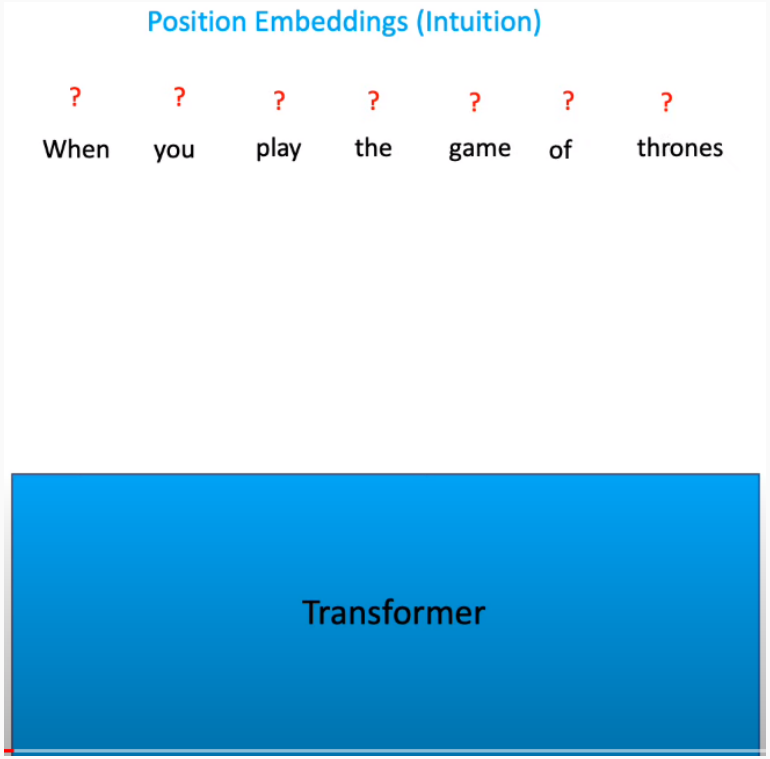


Imagen tomada de Haider, 2021.

Encoder: position embeddings

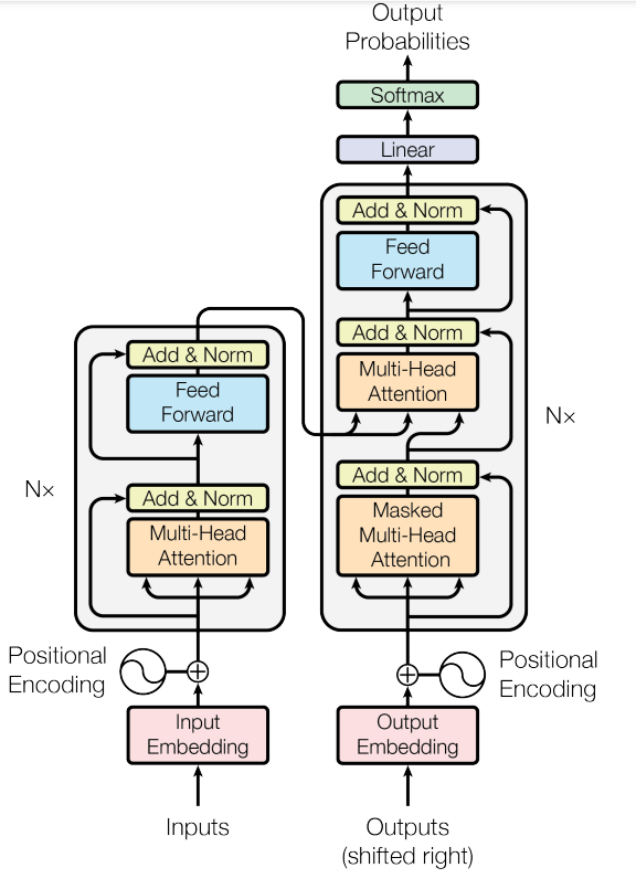


Imagen tomada de Vaswani et al, 2017.

Position Embeddings (Intuition)

Here is why order matters

Even though she did not win the award, she was satisfied.

Even though she did win the award, she was not satisfied.

Imagen tomada de Haider, 2021.

Encoder: position embeddings

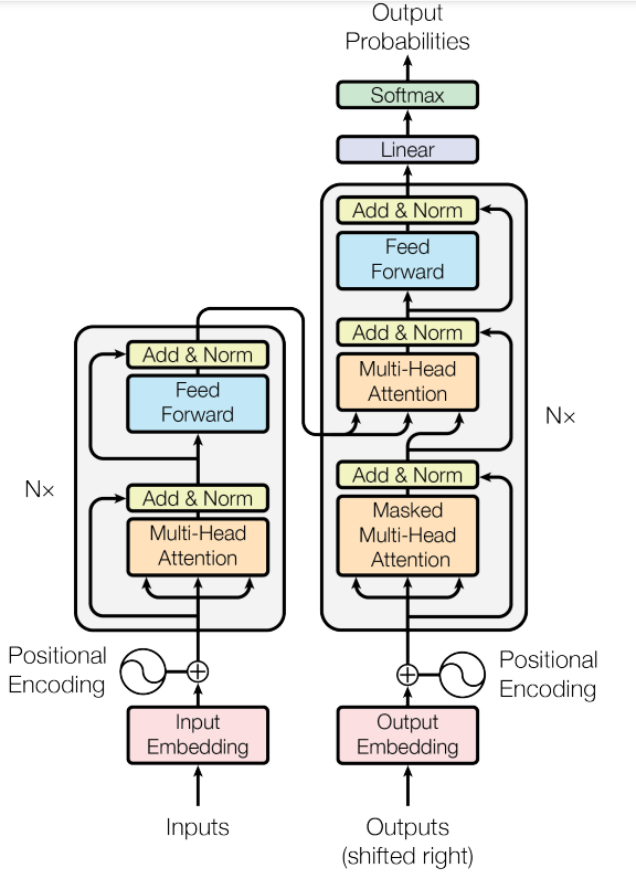


Imagen tomada de Vaswani et al, 2017.

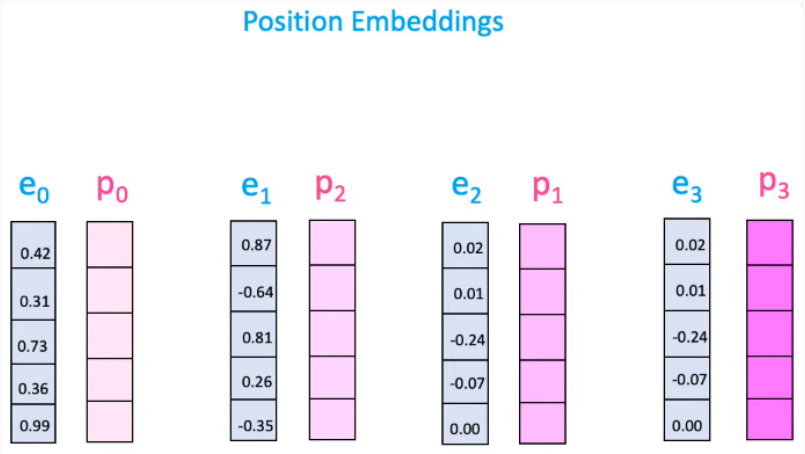


Imagen tomada de Haider, 2021.

Se introduce un nuevo conjunto de vectores que contienen / guardan la posición de las palabras.

Encoder: position embeddings

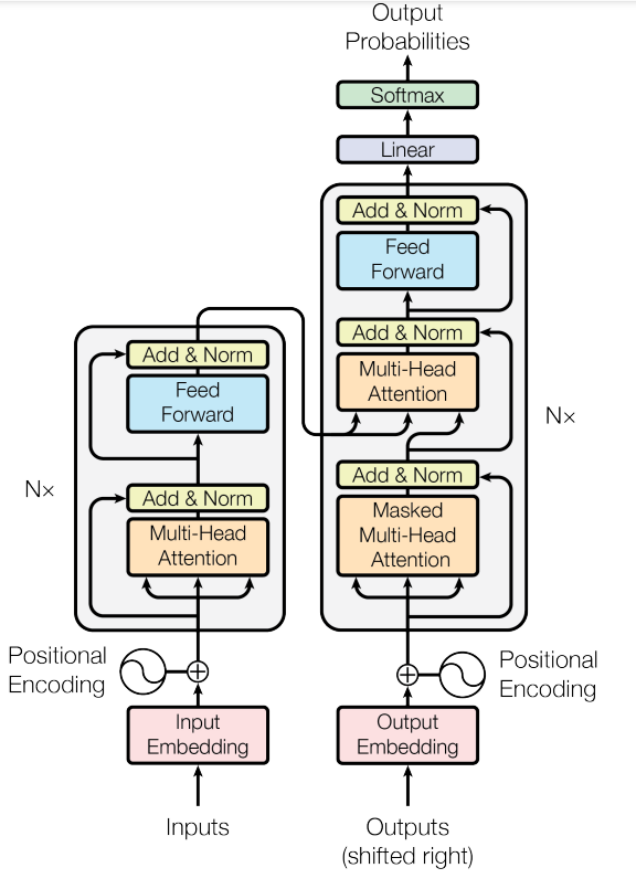


Imagen tomada de Vaswani et al, 2017.

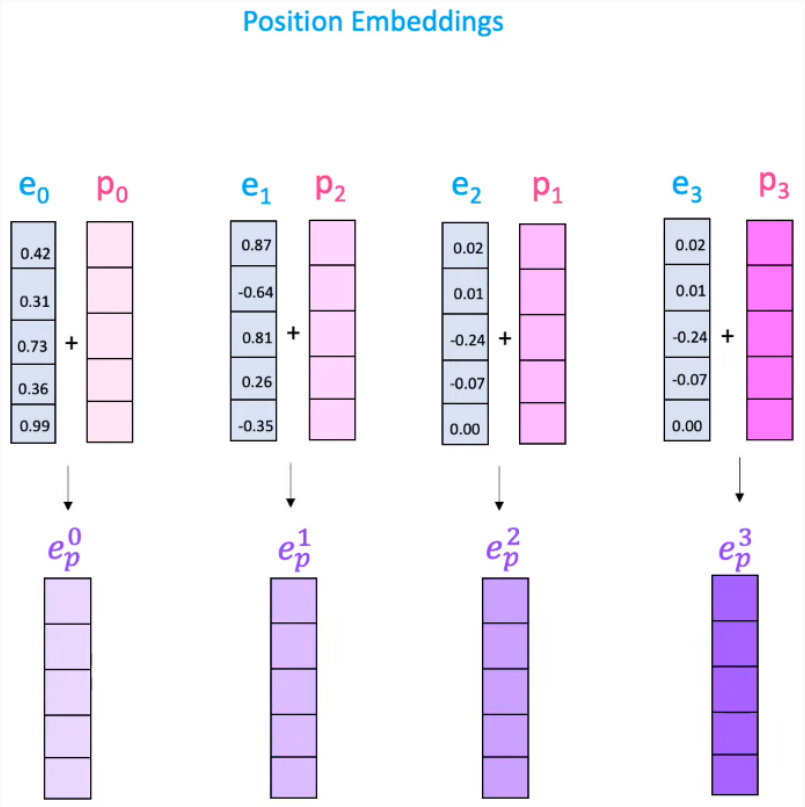


Imagen tomada de Haider, 2021.

Encoder: position embeddings

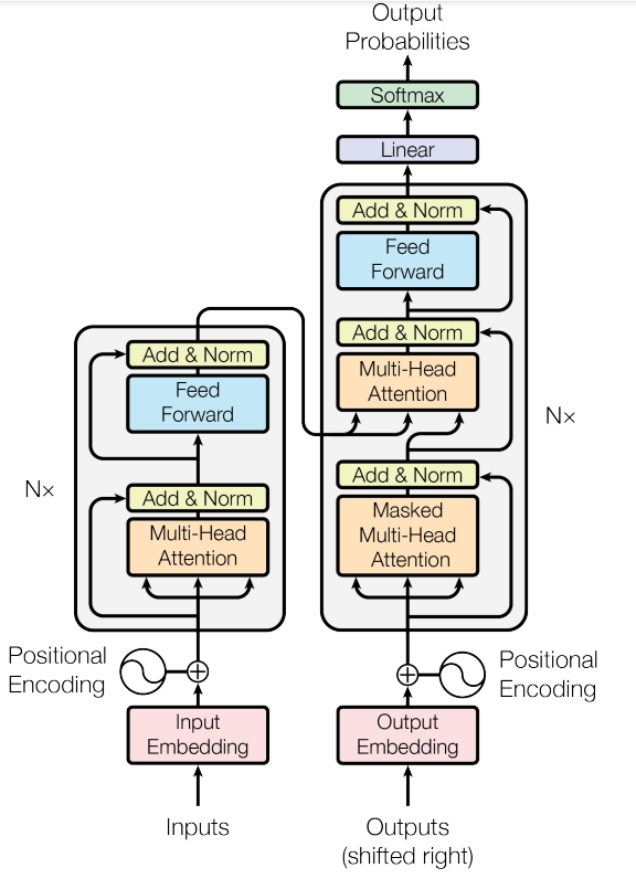


Imagen tomada de Vaswani et al, 2017.

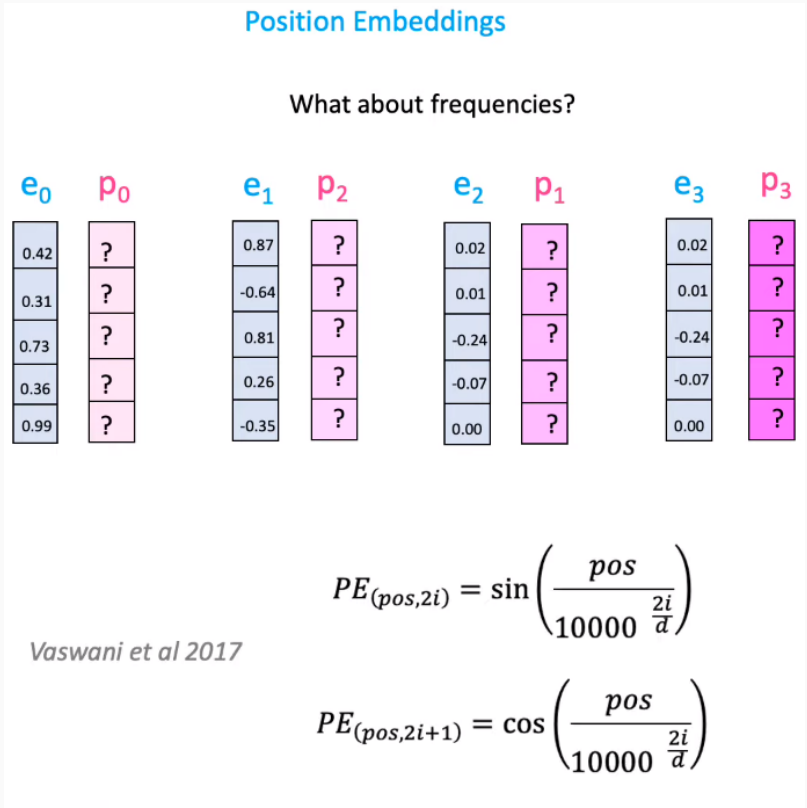


Imagen tomada de Haider, 2021.

Encoder: position embeddings

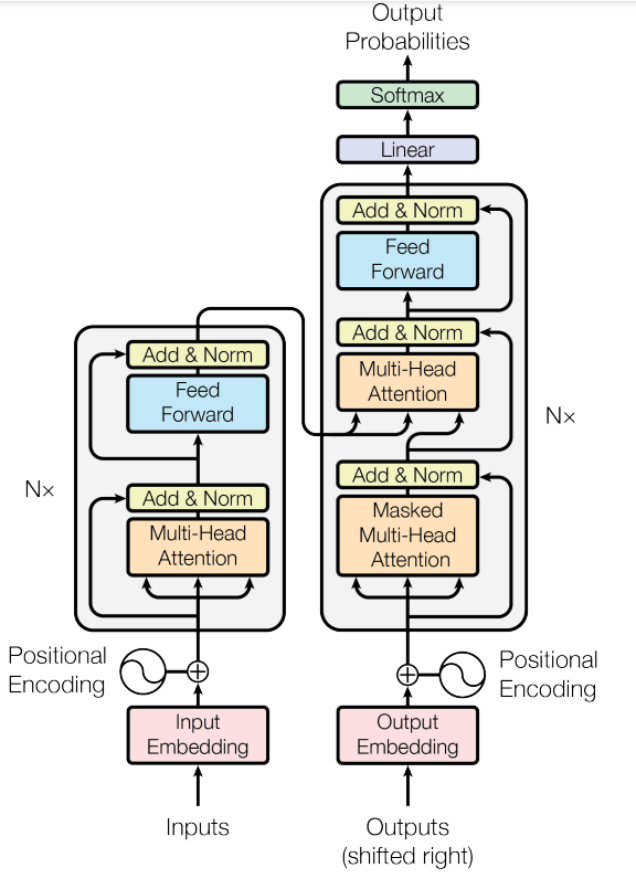


Imagen tomada de Vaswani et al, 2017.

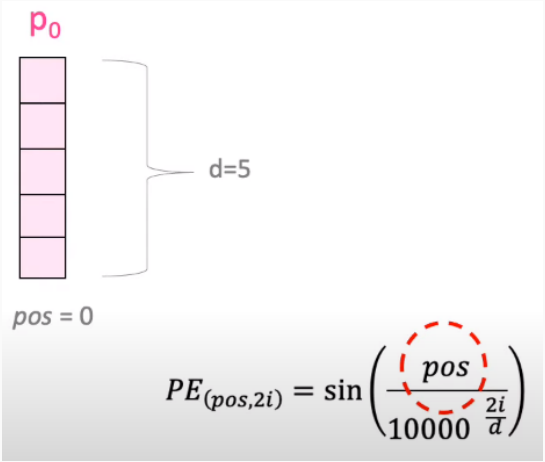


Imagen tomada de Haider, 2021.

Encoder: position embeddings

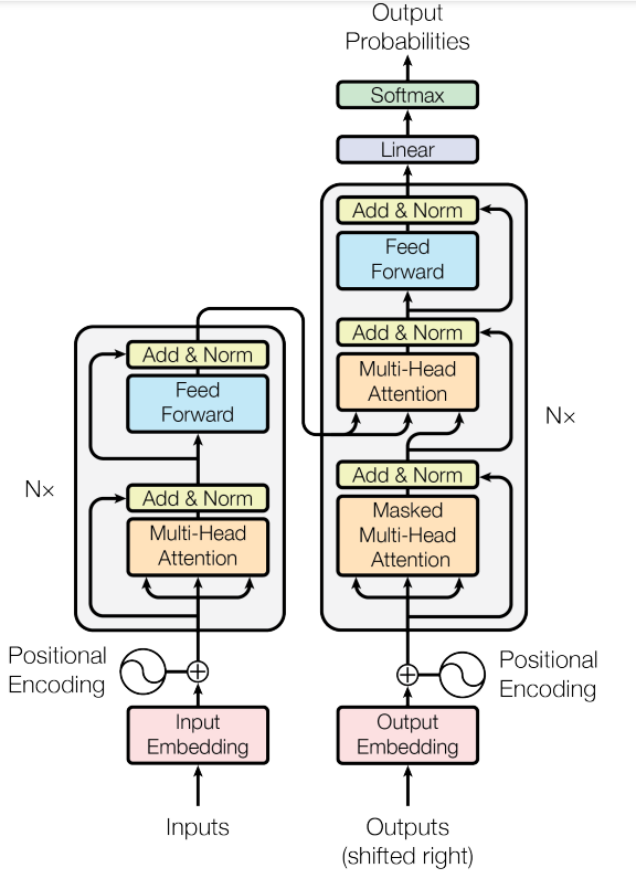


Imagen tomada de Vaswani et al, 2017.

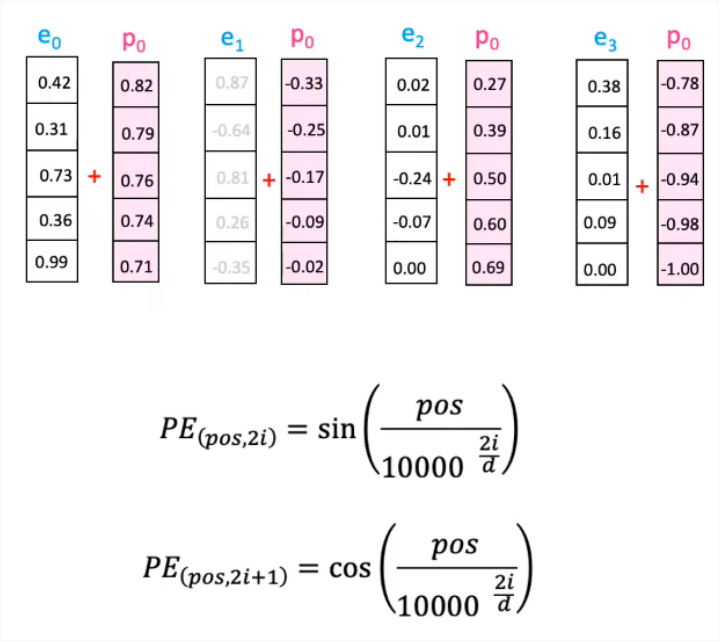


Imagen tomada de Haider, 2021.

Encoder: position embeddings

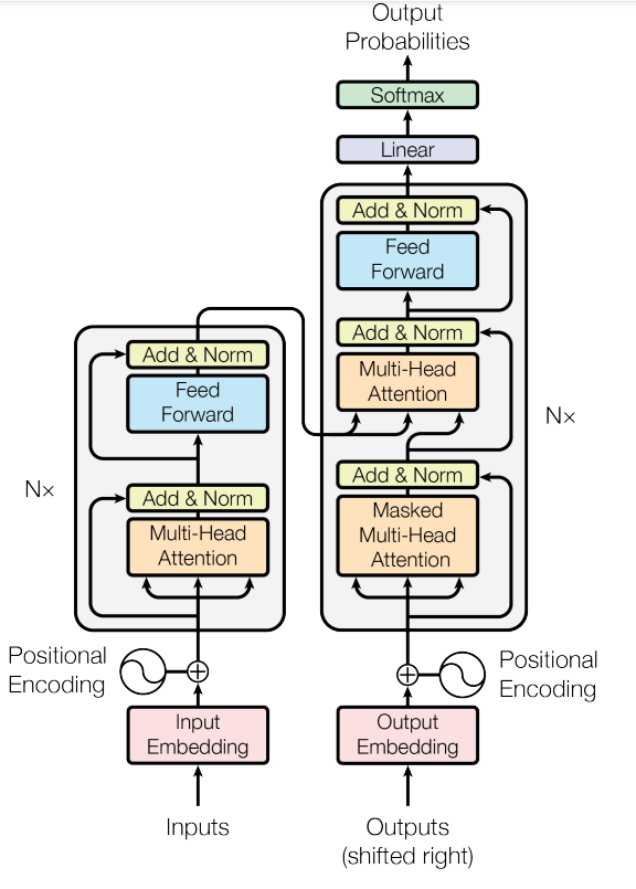


Imagen tomada de Vaswani et al, 2017.

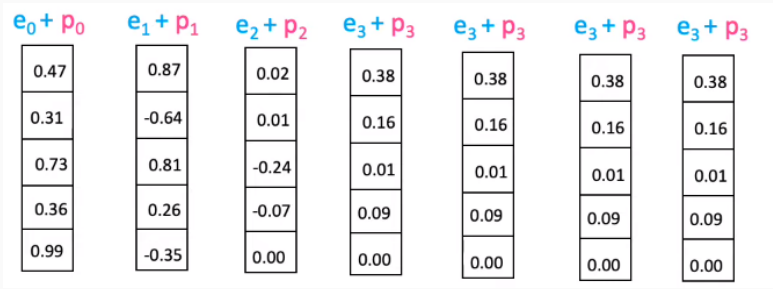


Imagen tomada de Haider, 2021.

Encoder: position embeddings

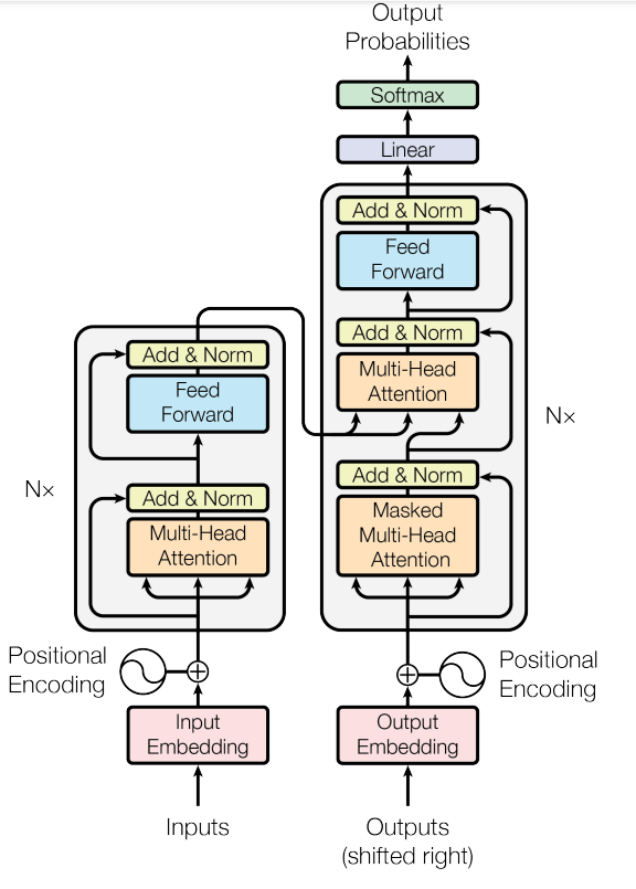


Imagen tomada de Vaswani et al, 2017.

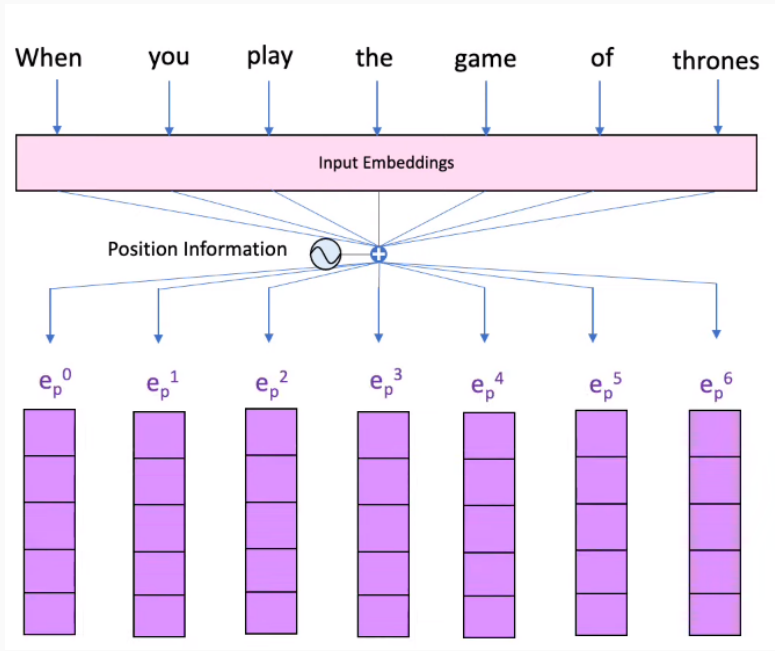


Imagen tomada de Haider, 2021.

Encoder: multi-head attention

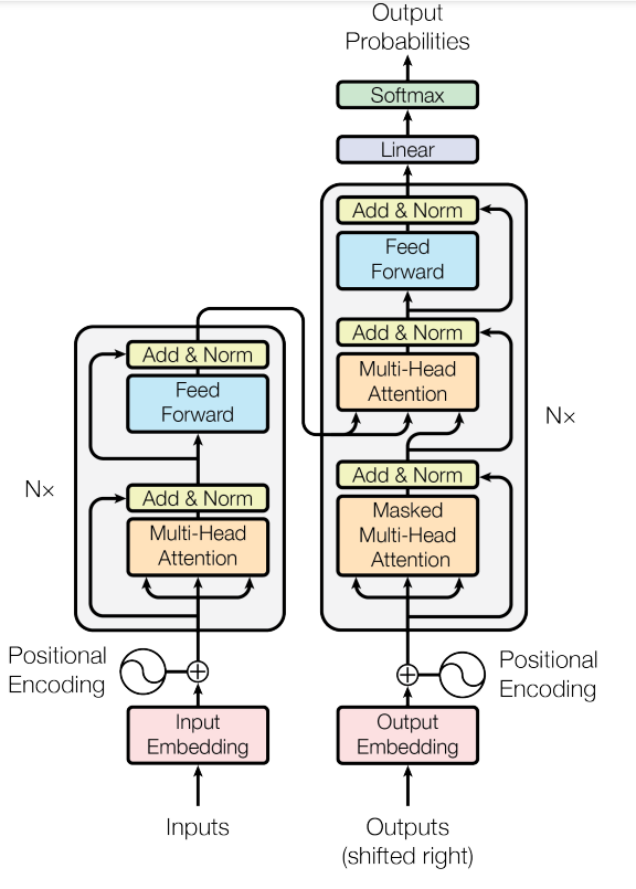


Imagen tomada de Vaswani et al, 2017.

Self-Attention

He went to the bank and learned of his empty account, after which he went to a river bank and cried.

Imagen tomada de Haider, 2021.

Encoder: multi-head attention

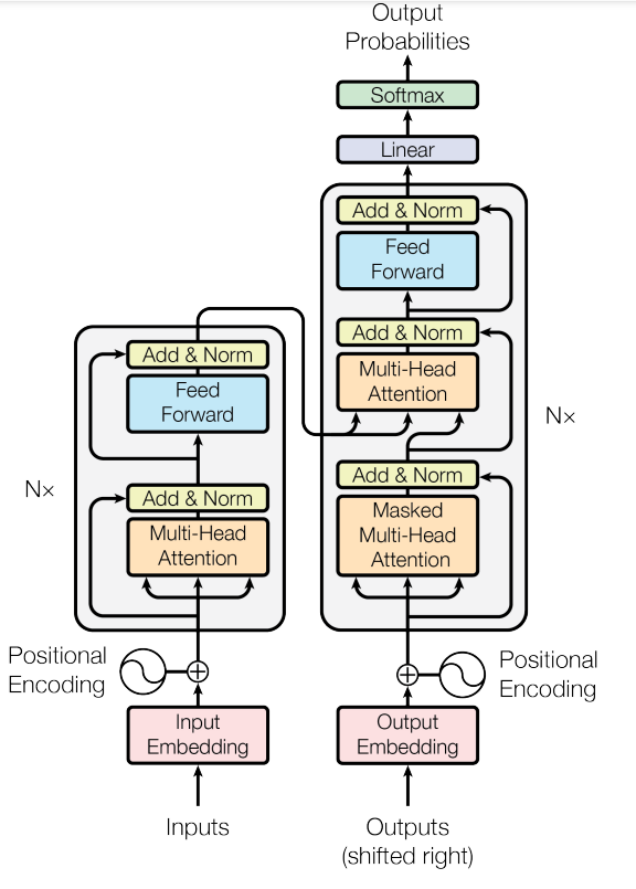


Imagen tomada de Vaswani et al, 2017.

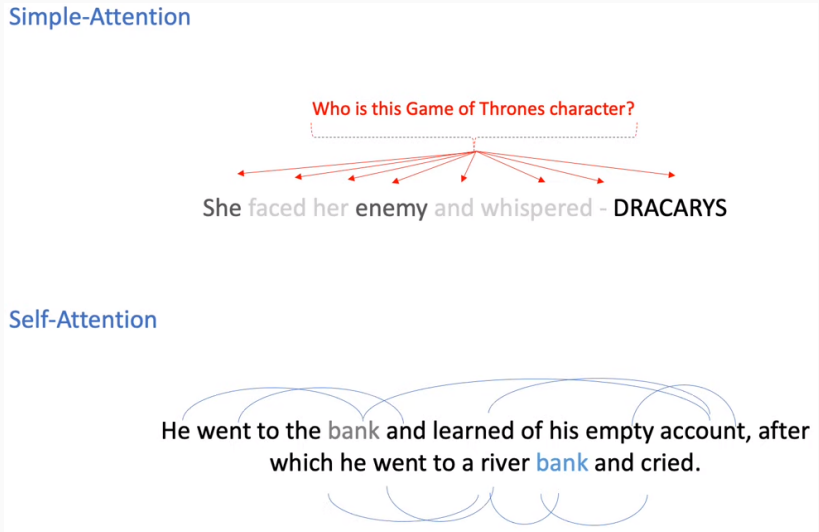


Imagen tomada de Haider, 2021.

Encoder: multi-head attention

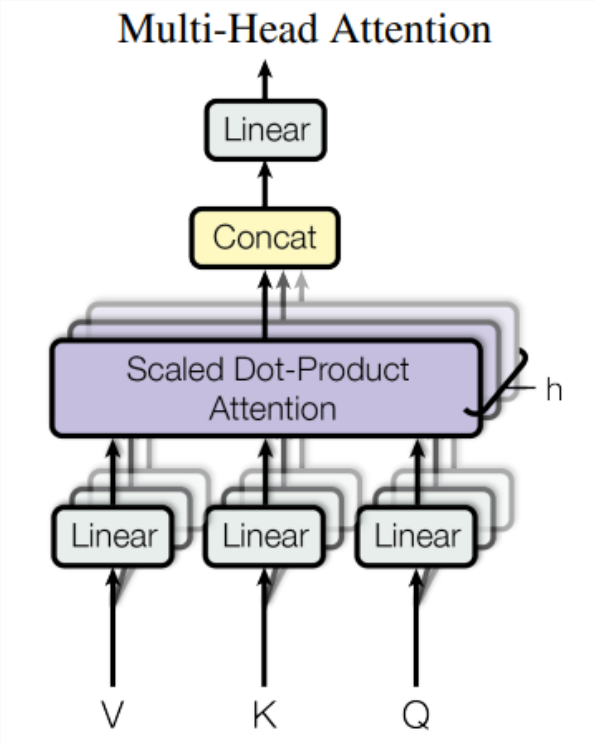


Imagen tomada de Vaswani et al, 2017.

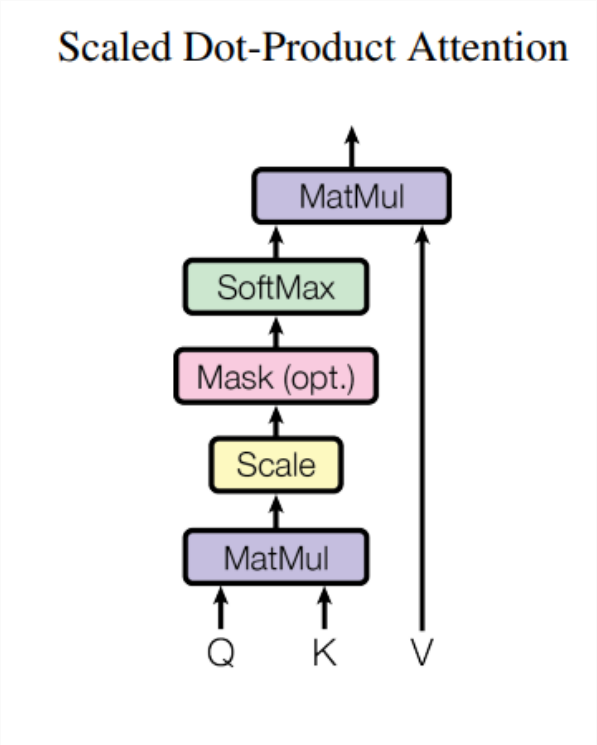


Imagen tomada de Vaswani et al, 2017.

Encoder: multi-head attention

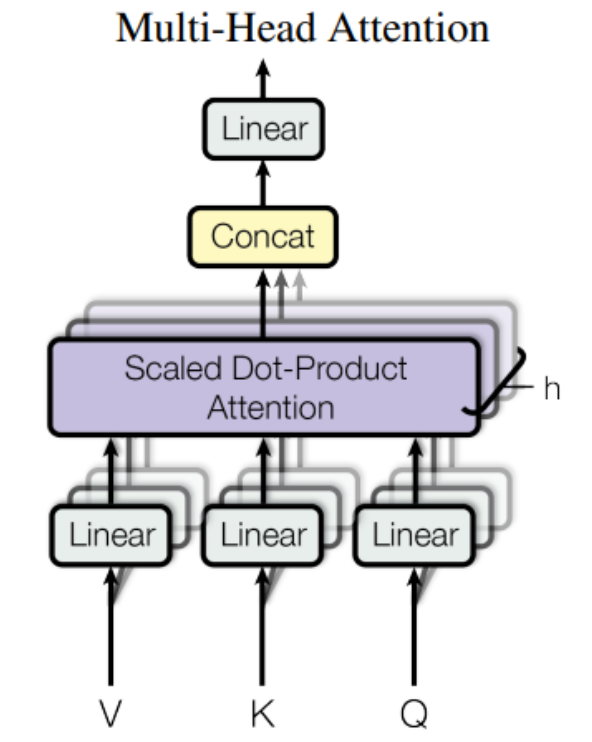


Imagen tomada de Vaswani et al, 2017.

Linear layers

- No tienen función de activación
- Mapear entradas hacia salidas
- Cambiar las dimensiones de los vectores / matrices.

Encoder: multi-head attention

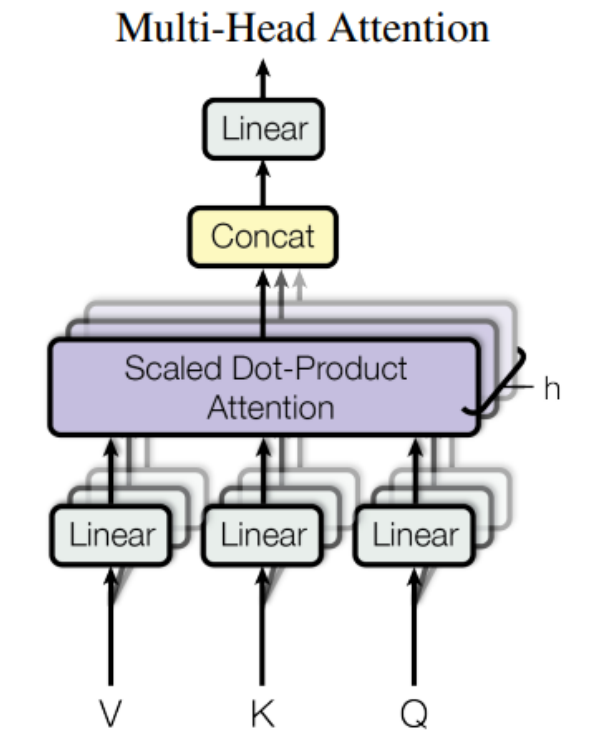


Imagen tomada de Vaswani et al, 2017.

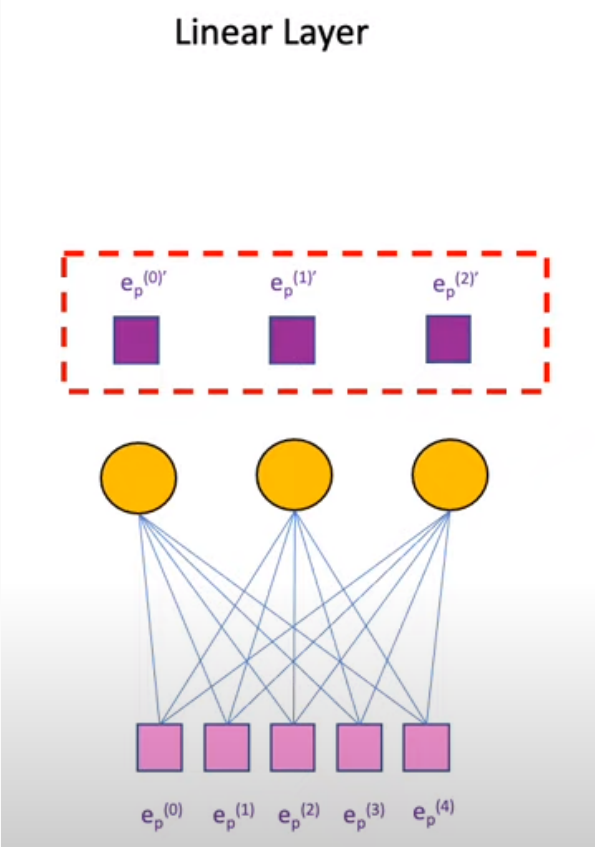


Imagen tomada de Haider, 2021.

Encoder: multi-head attention

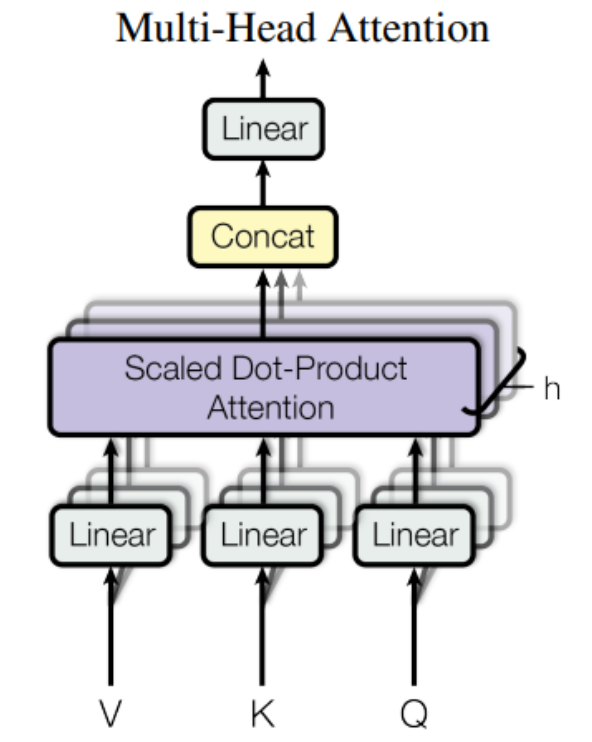


Imagen tomada de Vaswani et al, 2017.

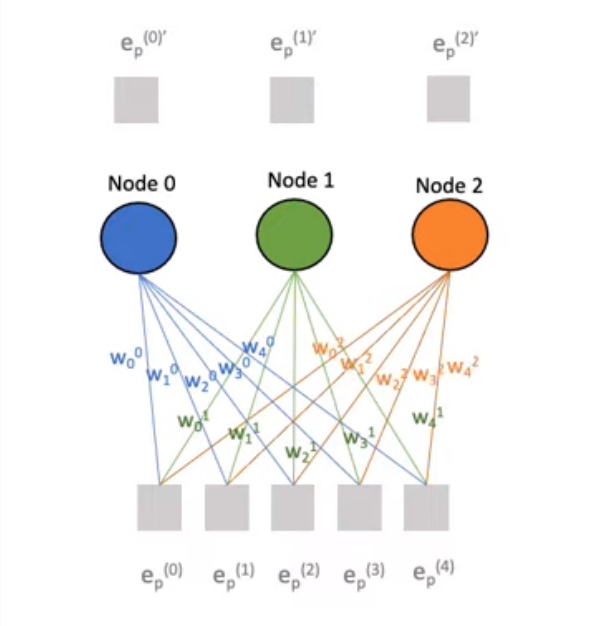


Imagen tomada de Haider, 2021.

Encoder: multi-head attention

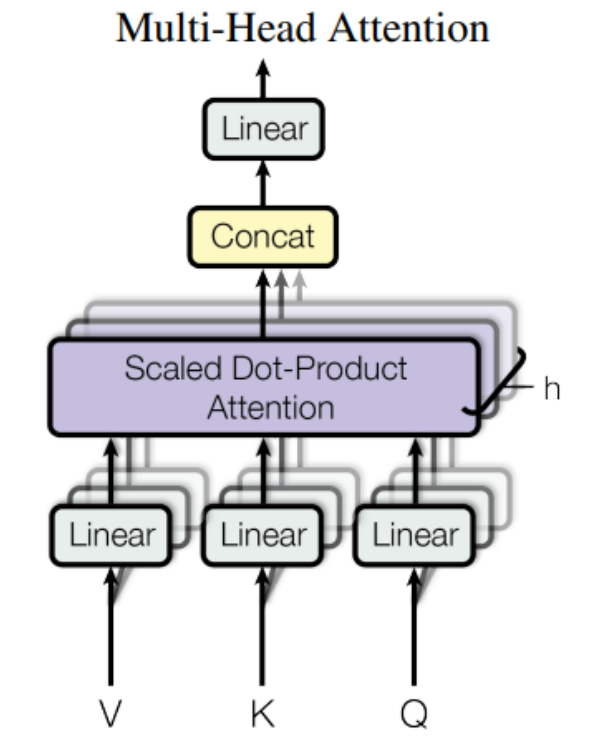


Imagen tomada de Vaswani et al, 2017.

Linear layers

- Query
- Key
- Value

Encoder: multi-head attention

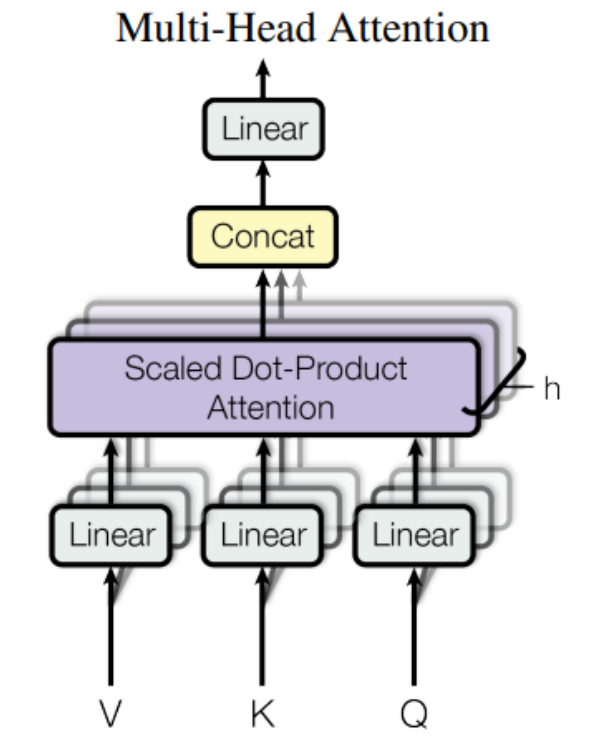


Imagen tomada de Vaswani et al, 2017.

Calculando la similitud:

$$S(Q, K) = \frac{Q * K}{scaling}$$

Encoder: multi-head attention

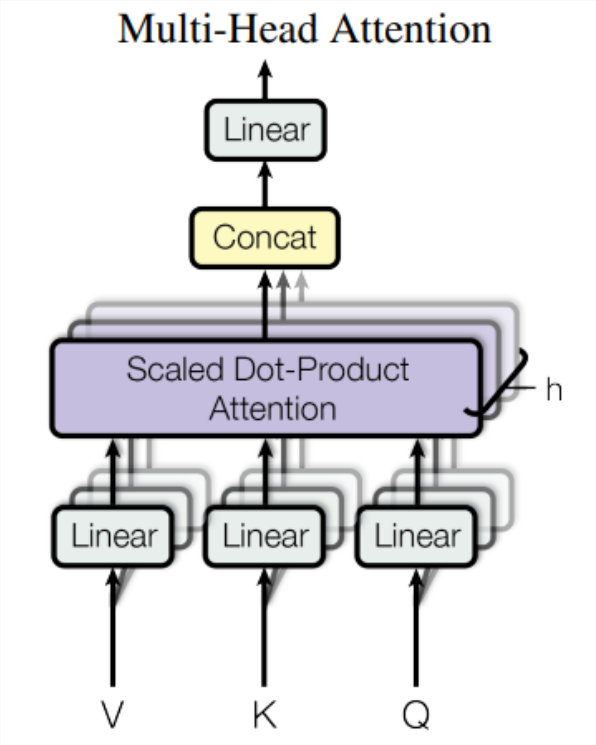


Imagen tomada de Vaswani et al, 2017.

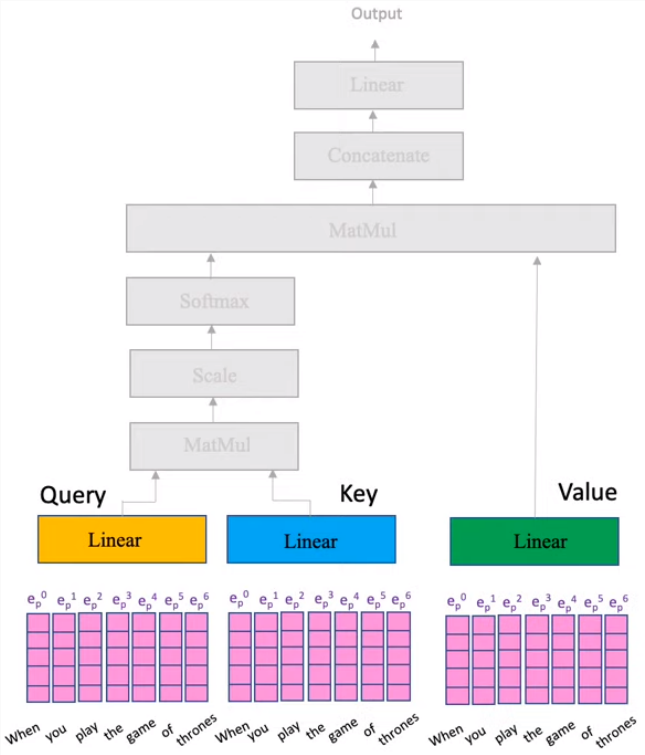


Imagen tomada de Haider, 2021.

Encoder: multi-head attention

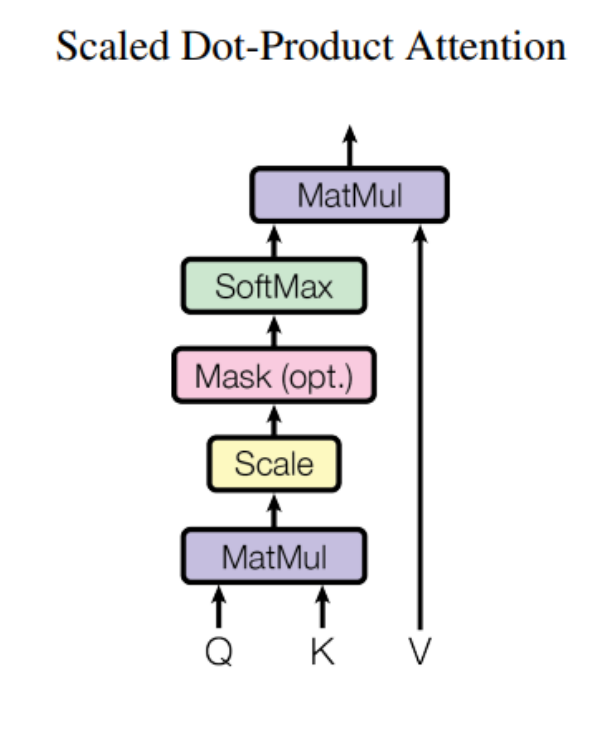


Imagen tomada de Vaswani et al, 2017.

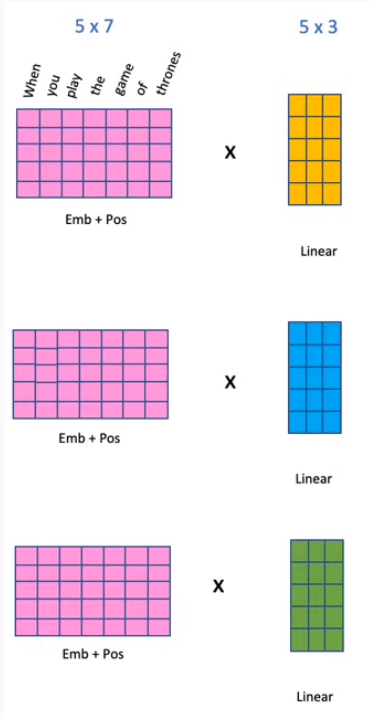


Imagen tomada de Haider, 2021.

Encoder: multi-head attention

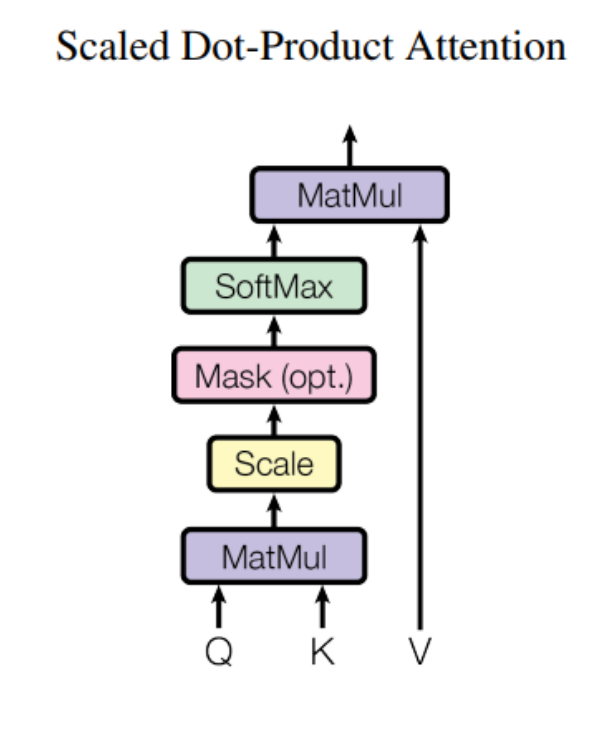


Imagen tomada de Vaswani et al, 2017.

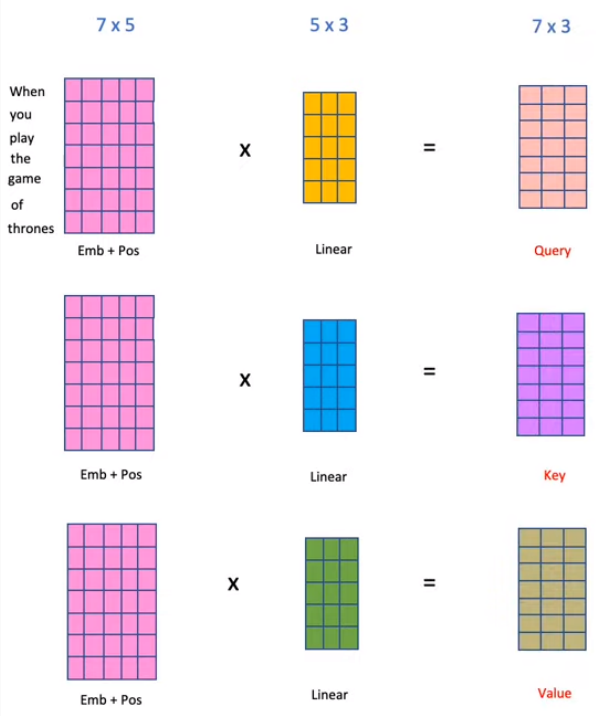


Imagen tomada de Haider, 2021.

Encoder: multi-head attention

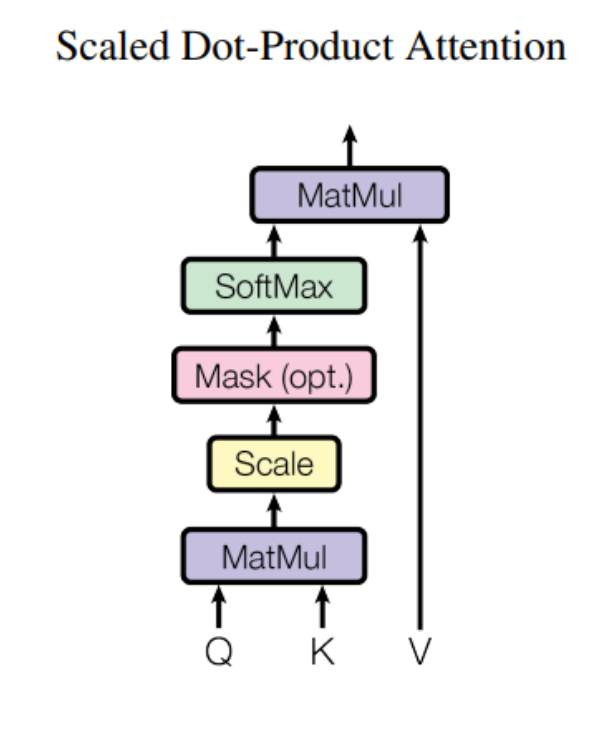


Imagen tomada de Vaswani et al, 2017.

Cálculo de la similitud:

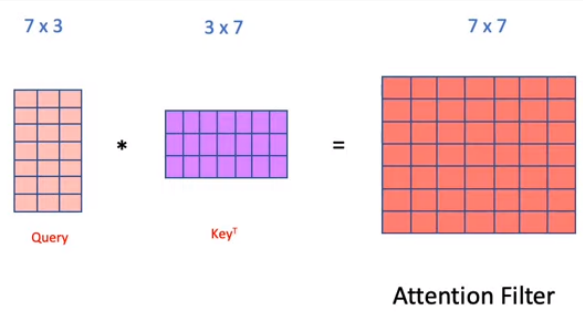


Imagen tomada de Haider, 2021.

Encoder: multi-head attention

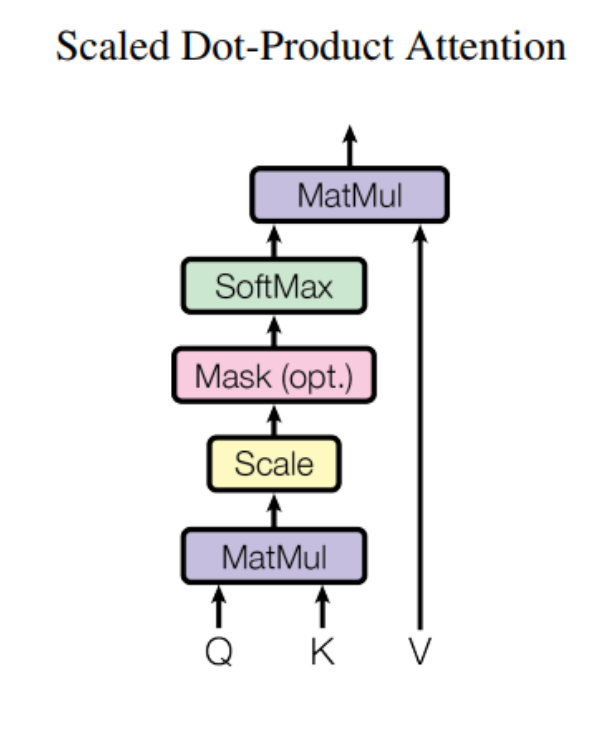


Imagen tomada de Vaswani et al, 2017.

Attention filter

	When	you	play	the	game	of	thrones
When	89	48	41	36	35	40	19
you	67	91	11	92	17	99	11
play	91	10	11	11	12	41	98
the	11	96	28	12	98	11	00
game	76	11	91	24	12	12	12
of	11	29	77	78	22	93	13
thrones	11	87	12	12	13	98	19

Imagen tomada de Haider, 2021.

Encoder: multi-head attention

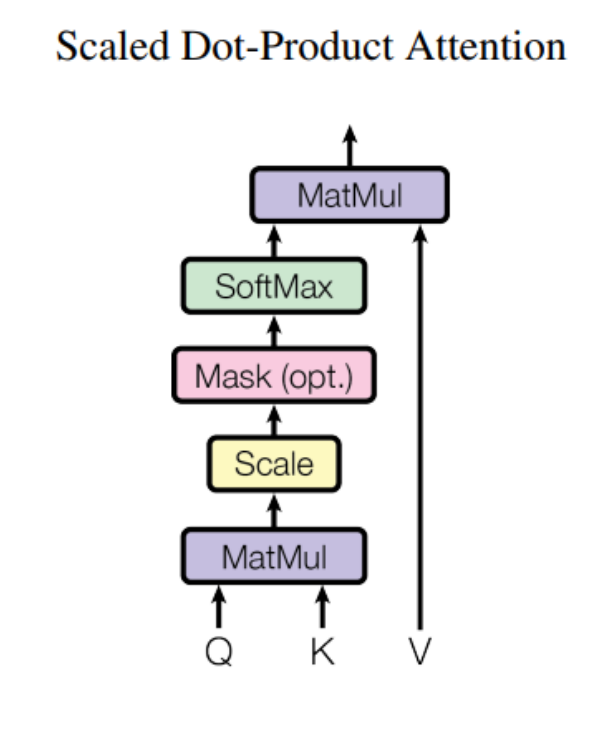


Imagen tomada de Vaswani et al, 2017.

Attention filter (after training)

	When	you	play	the	game	of	thrones
When	89	20	41	10	55	78	59
you	90	98	81	22	87	15	32
play	29	81	95	10	90	30	92
the	10	22	67	12	88	40	89
game	22	70	90	56	98	44	80
of	10	15	30	40	44	44	59
thrones	59	72	92	90	13	59	99

Attention scores

Imagen tomada de Haider, 2021.

Encoder: multi-head attention

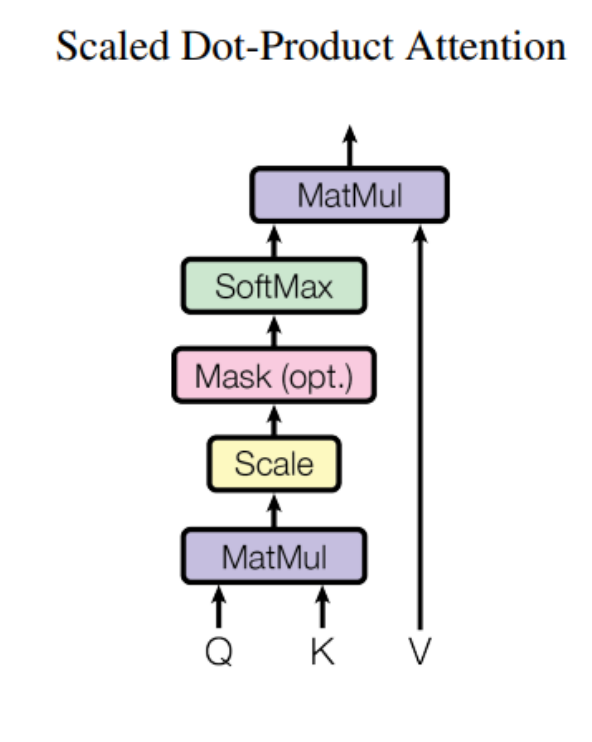


Imagen tomada de Vaswani et al, 2017.

Scale

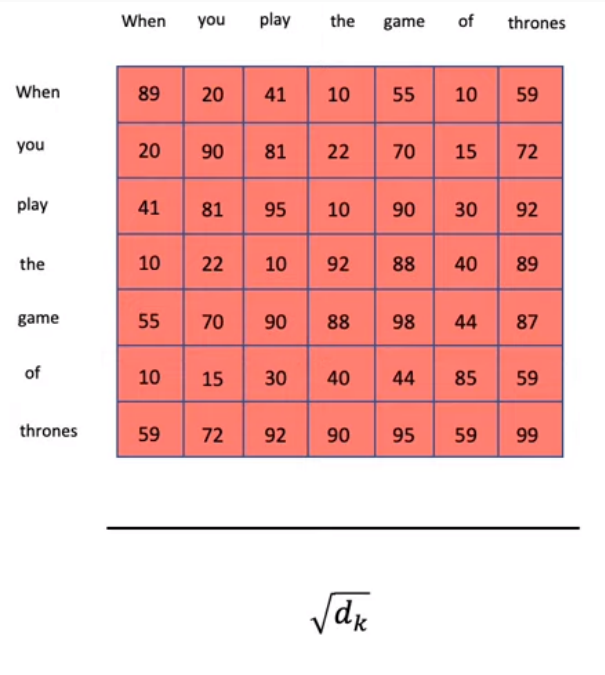


Imagen tomada de Haider, 2021.

Encoder: multi-head attention

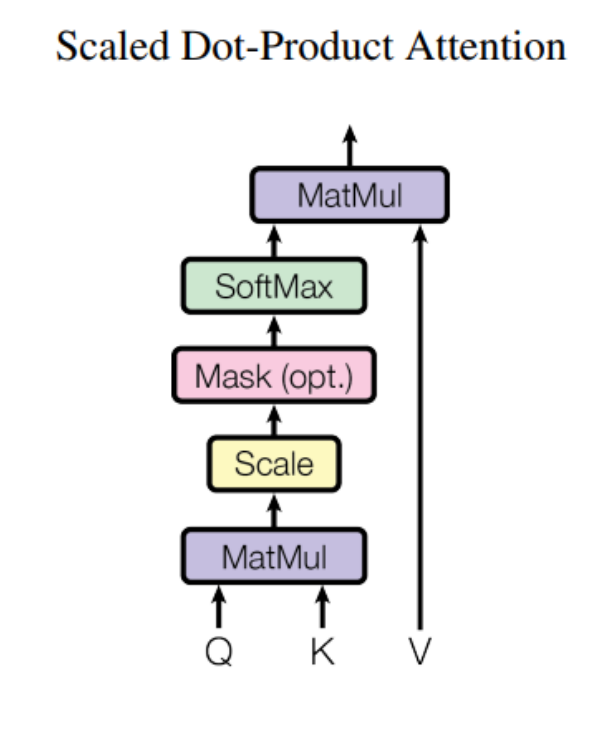


Imagen tomada de Vaswani et al, 2017.

Softmax

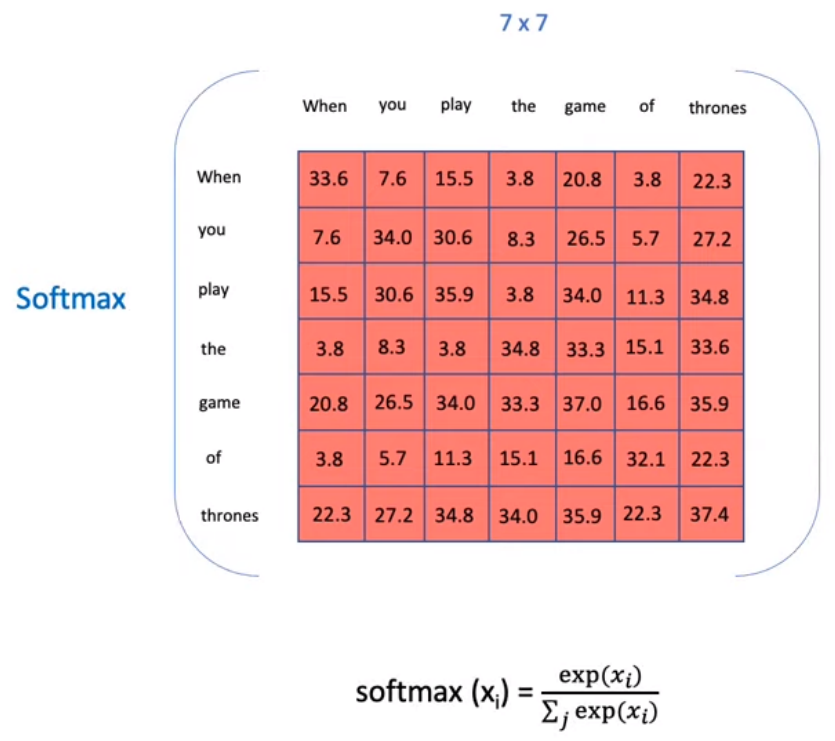


Imagen tomada de Haider, 2021.

Encoder: multi-head attention

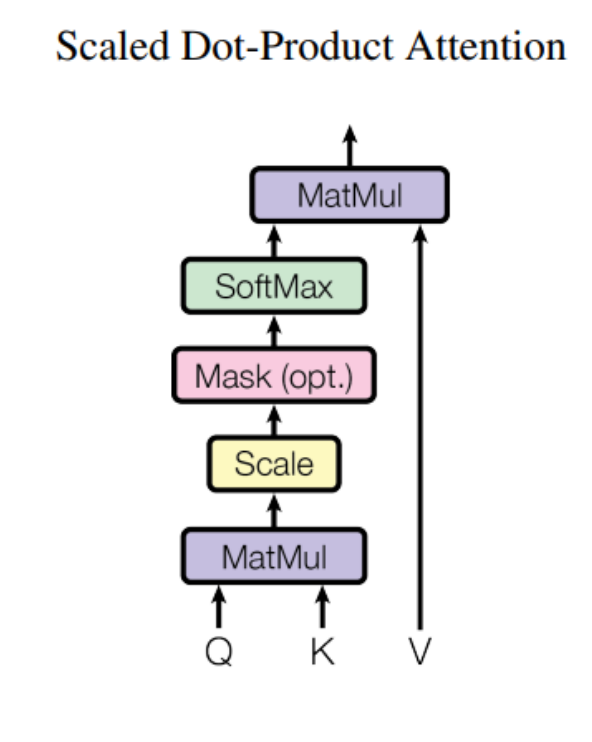


Imagen tomada de Vaswani et al, 2017.

MatMul

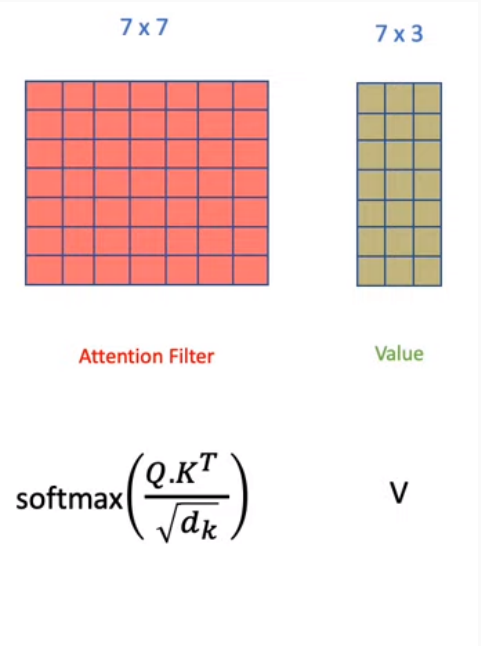


Imagen tomada de Haider, 2021.

Encoder: multi-head attention

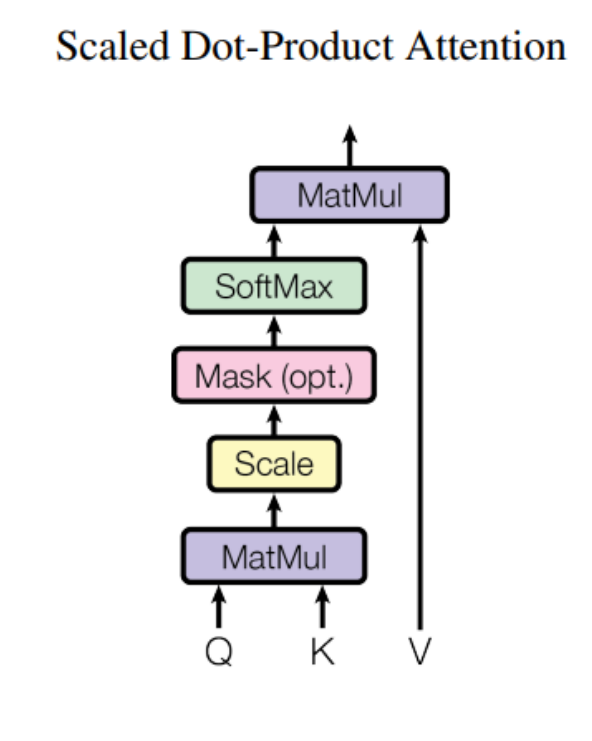


Imagen tomada de Vaswani et al, 2017.

MatMul

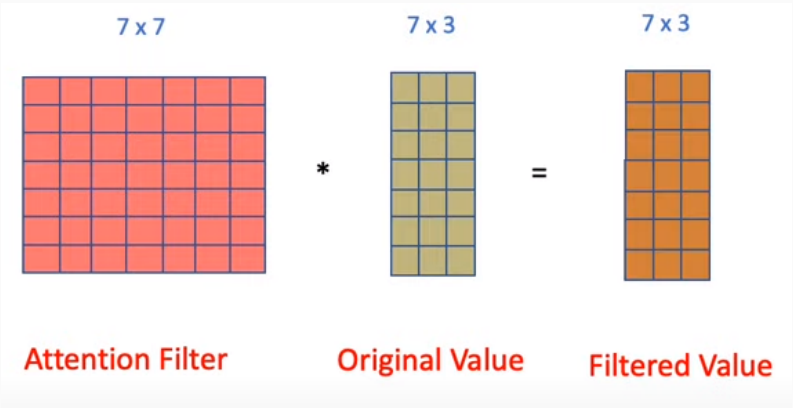


Imagen tomada de Haider, 2021.

Encoder: multi-head attention

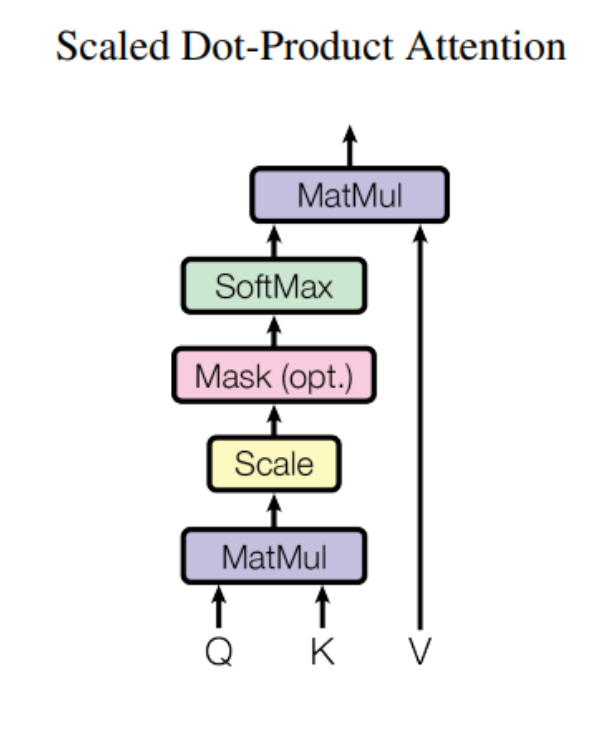


Imagen tomada de Vaswani et al, 2017.

Multi-Head attention output

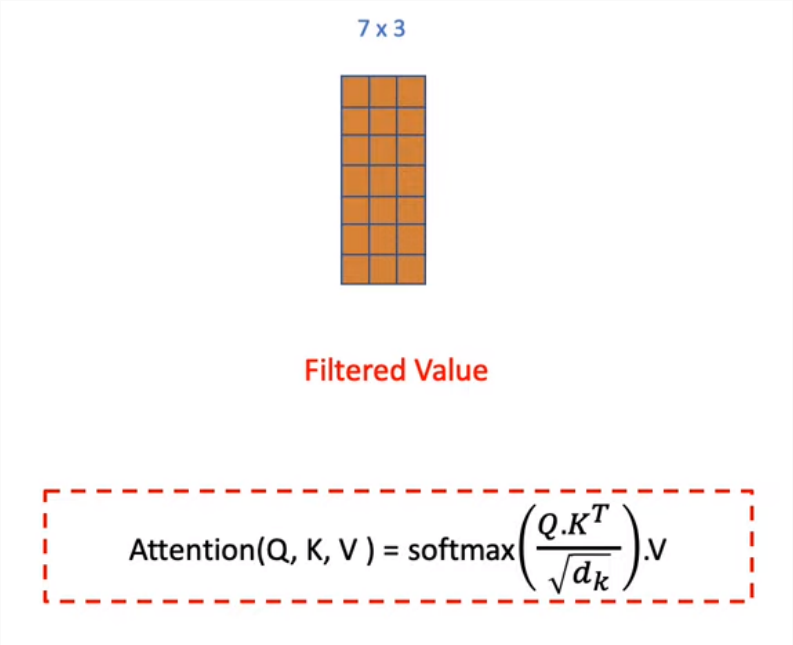


Imagen tomada de Haider, 2021.

Multi-head attention

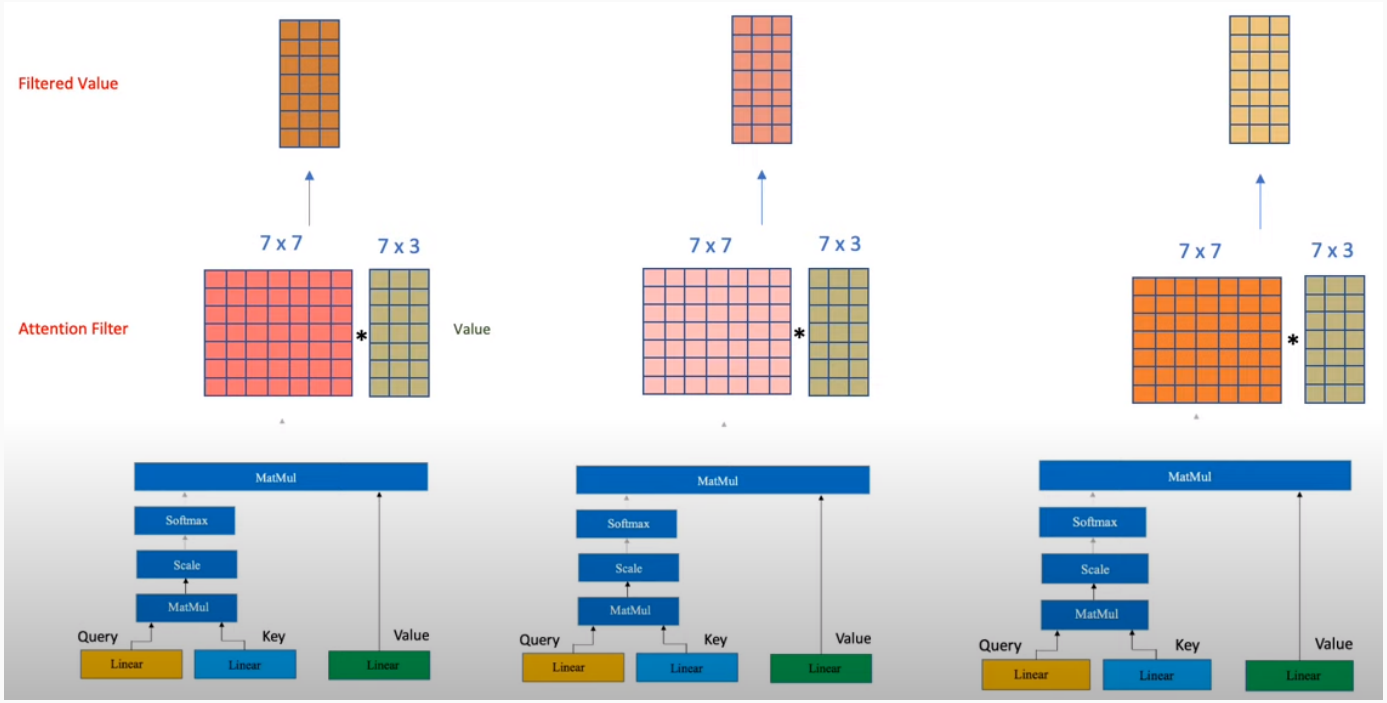


Imagen tomada de Haider, 2021.

Residual connection

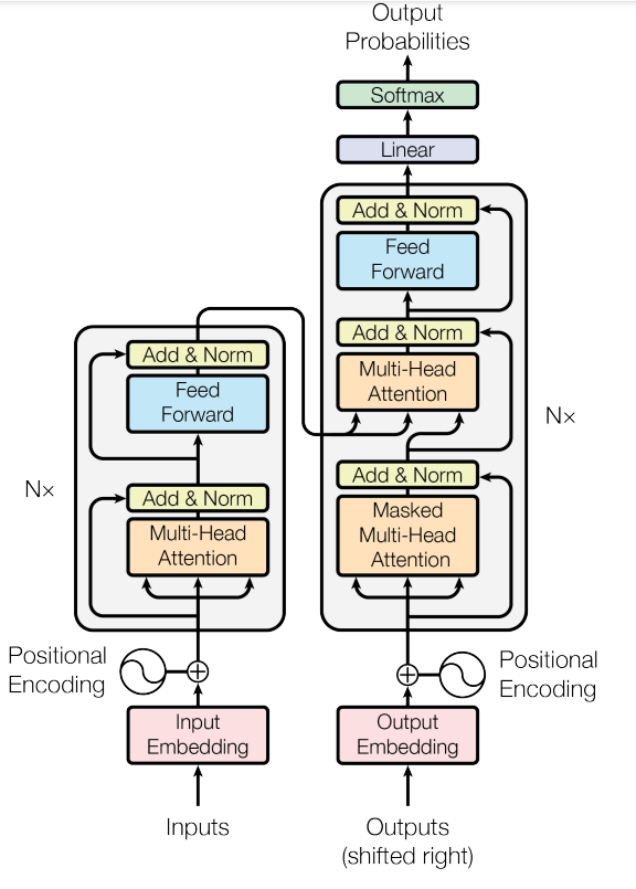


Imagen tomada de Vaswani et al, 2017.

- Preservar / mantener el conocimiento
- Evitar el problema del desvanecimiento del gradiente.

Residual connection

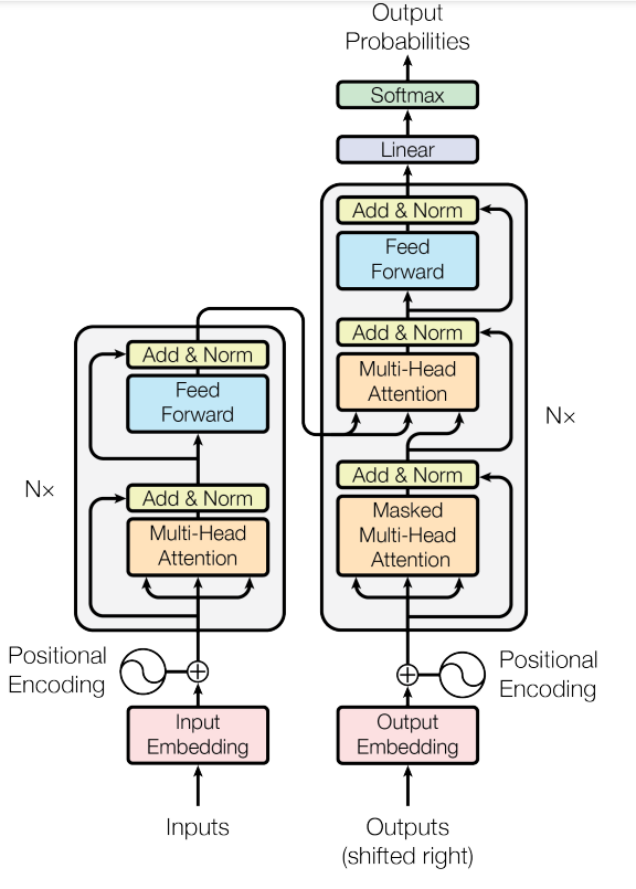


Imagen tomada de Vaswani et al, 2017.

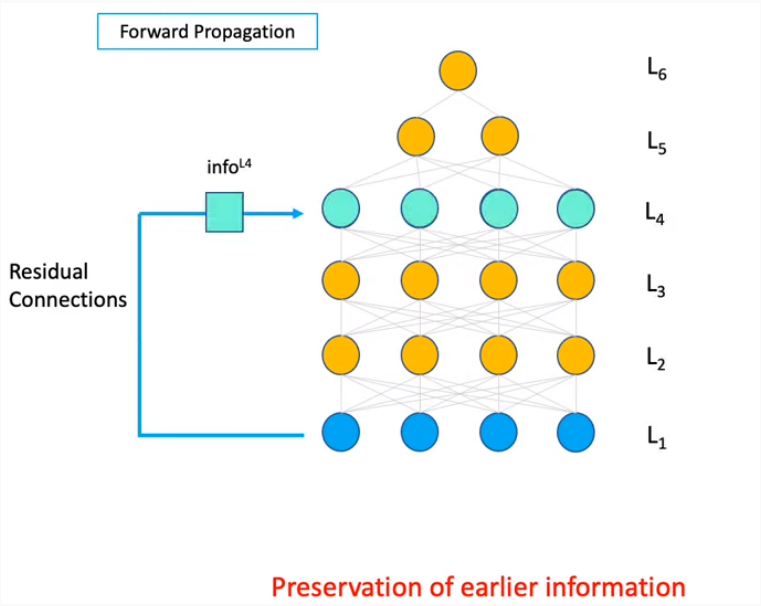


Imagen tomada de Haider, 2021.

Residual connection

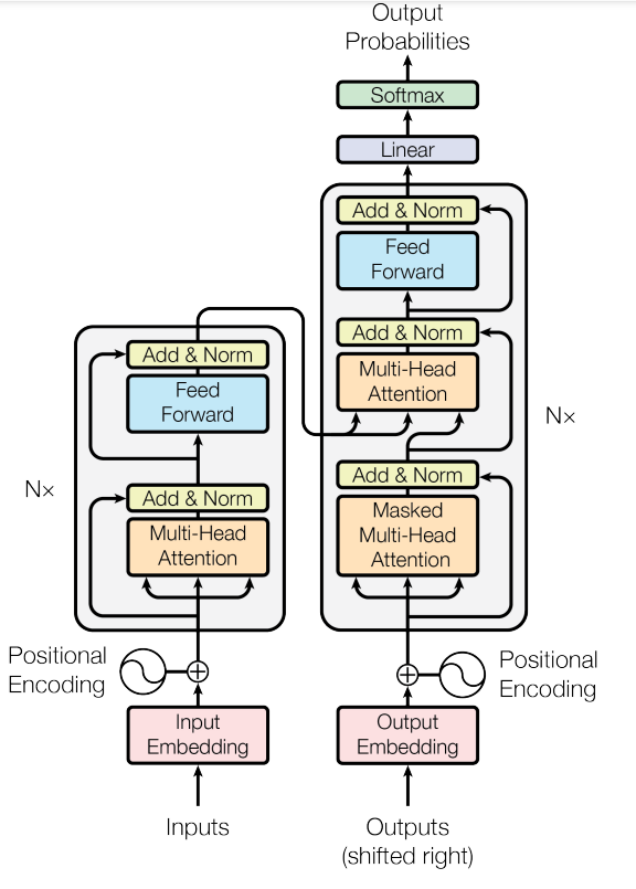


Imagen tomada de Vaswani et al, 2017.

Conexión residual

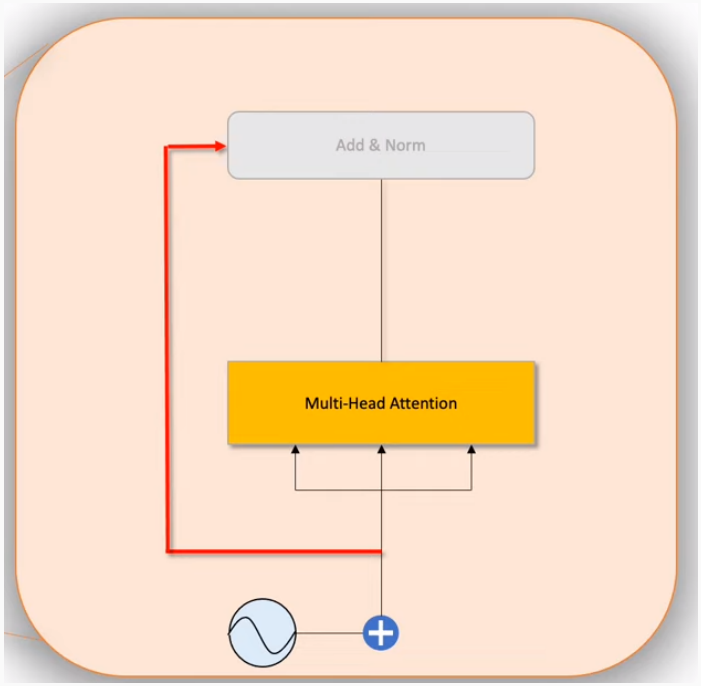


Imagen tomada de Haider, 2021.

Add layer

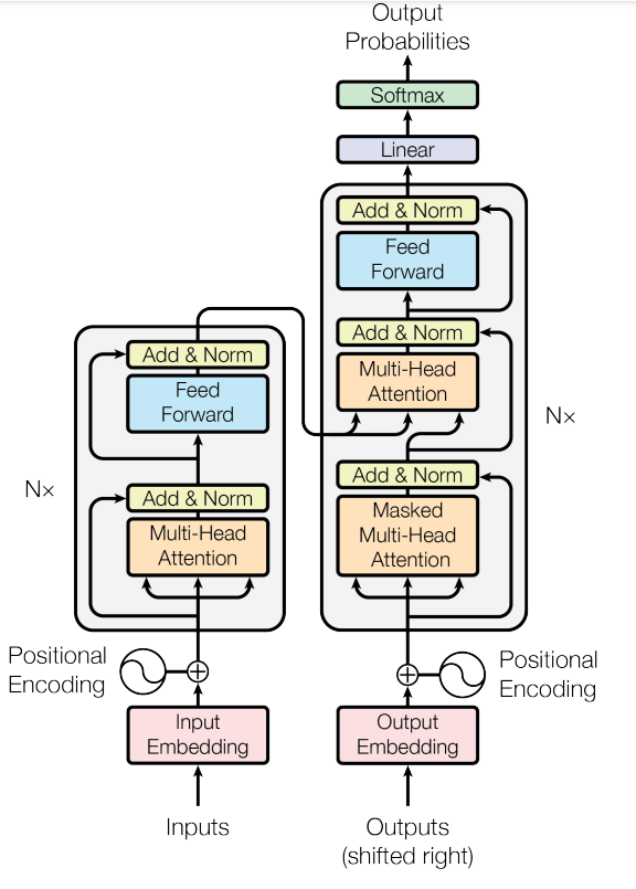


Imagen tomada de Vaswani et al, 2017.

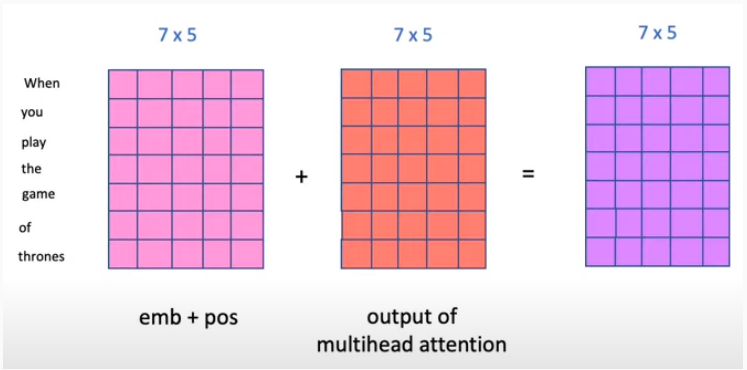


Imagen tomada de Haider, 2021.

Normalization layer

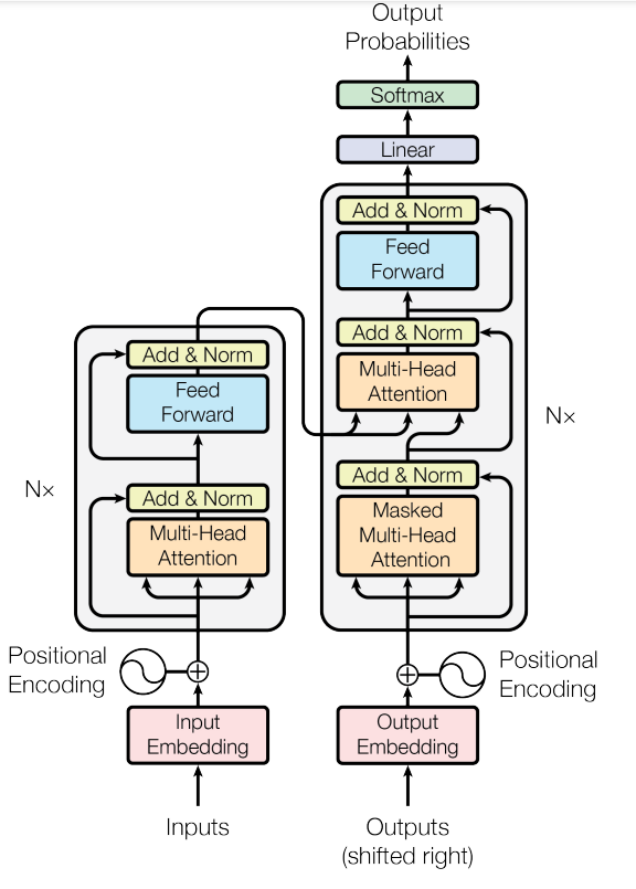


Imagen tomada de Vaswani et al, 2017.

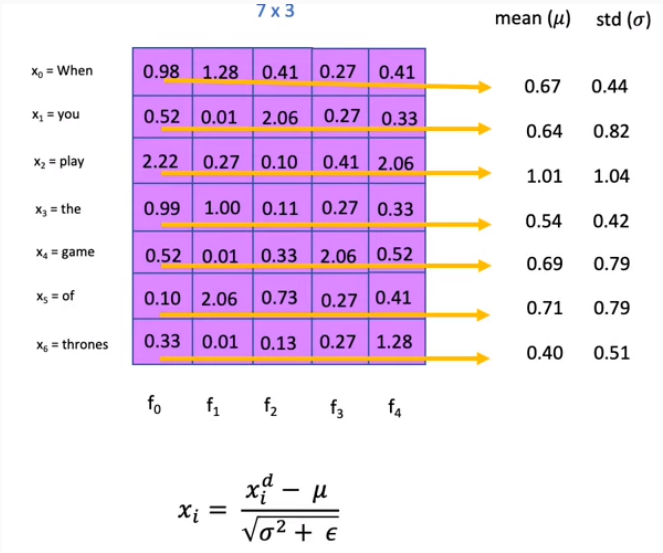


Imagen tomada de Haider, 2021.

Feed forward layer

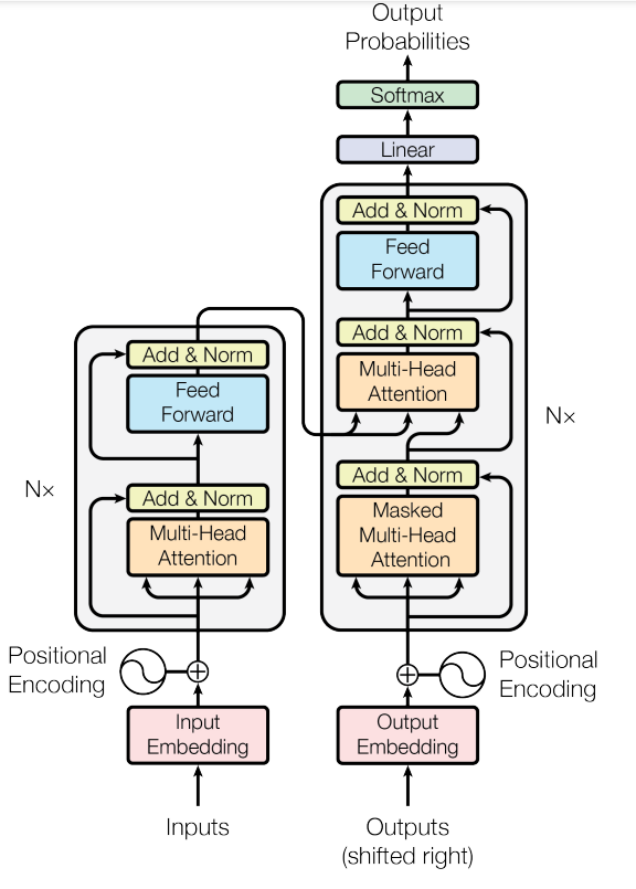


Imagen tomada de Vaswani et al, 2017.

- Linear
- ReLu
- Linear

Encoder layer

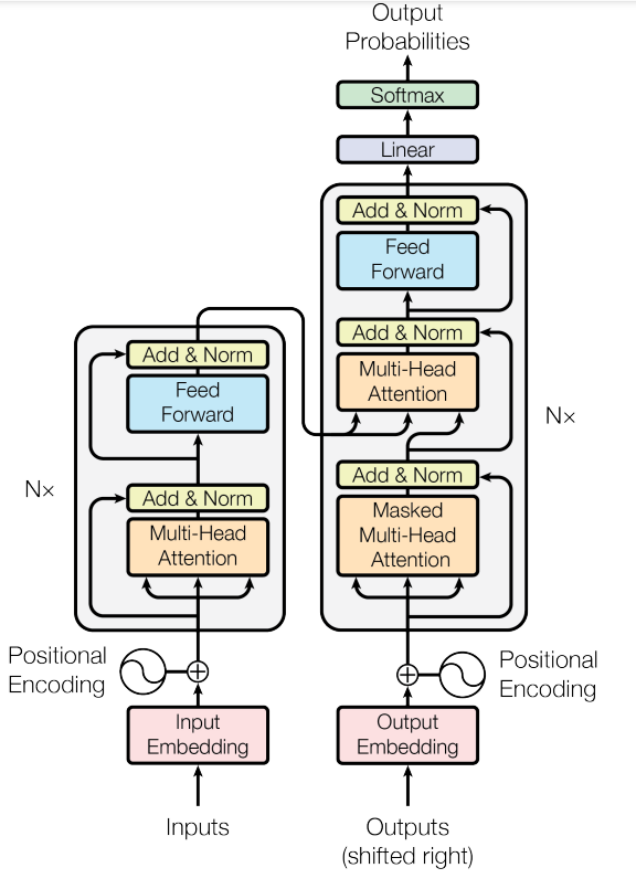


Imagen tomada de Vaswani et al, 2017.

Felicidades: ahora ya conoces cómo funciona la red *Encoder* en la arquitectura *Transformer.*

En resumen:

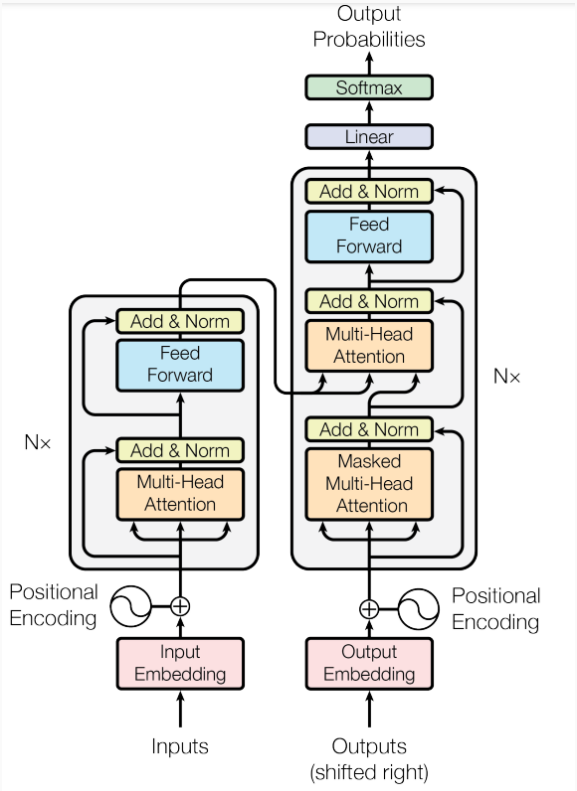


Imagen tomada de Vaswani et al, 2017.

- El *Encoder*, toma como entrada un texto y lo transforma a una representación vectorizada.
- El *Decoder* toma esta representación y lo convierte en un nuevo 'texto'.

Decoder layer

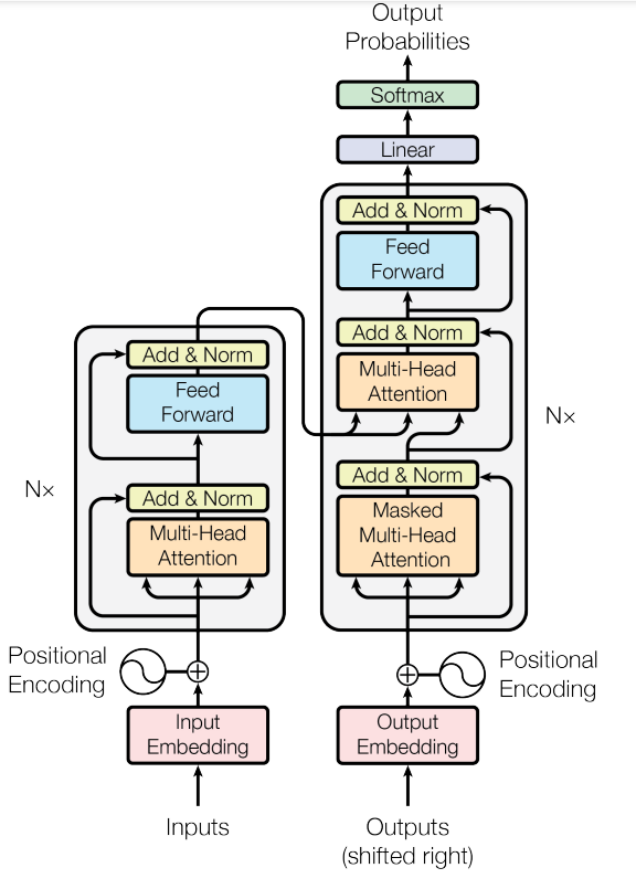


Imagen tomada de Vaswani et al, 2017.

Entradas:

1. La representación vectorizada (encoder)
2. La salida generada (decoder).

Multi-head attention (decoder)

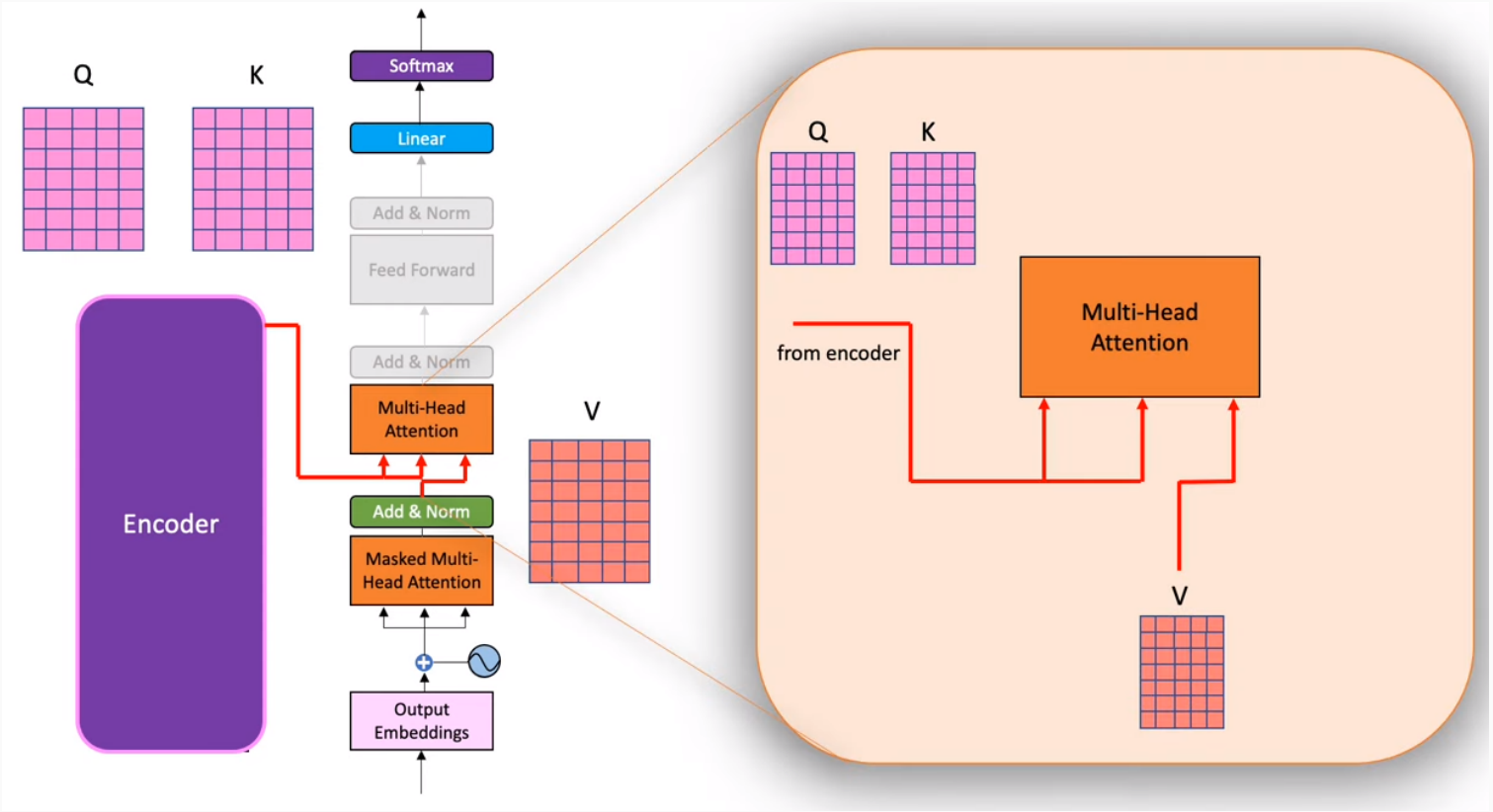


Imagen tomada de Haider, 2021.

Tamaño de la última capa lineal

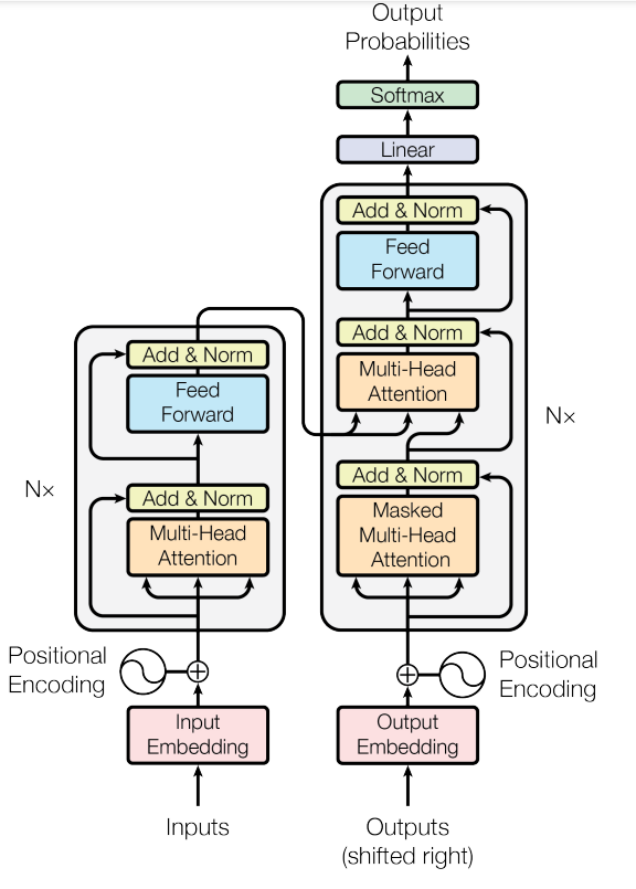


Imagen tomada de Vaswani et al, 2017.

- El número de neuronas de la capa final, será el número de palabras en el vocabulario.

Tamaño de la última capa lineal

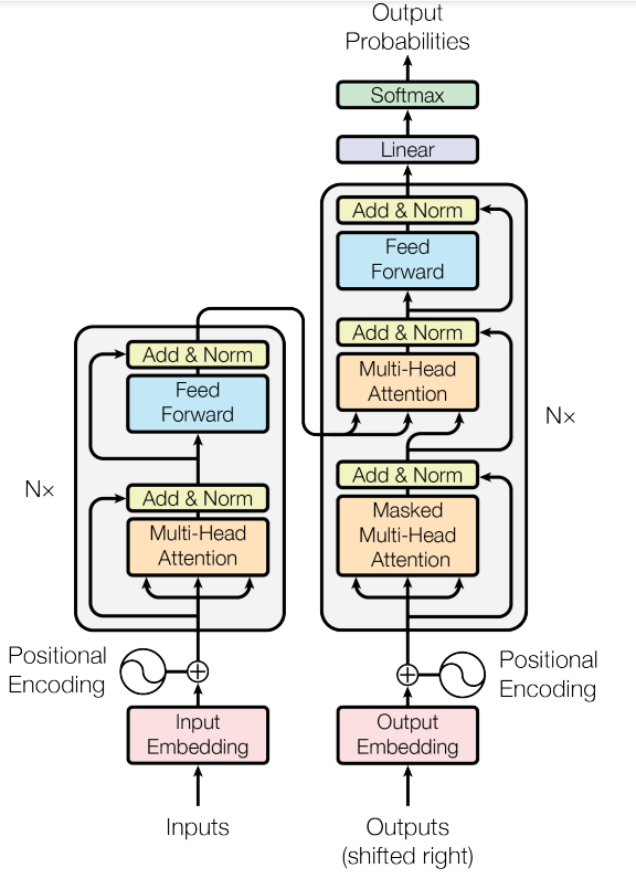


Imagen tomada de Vaswani et al, 2017.

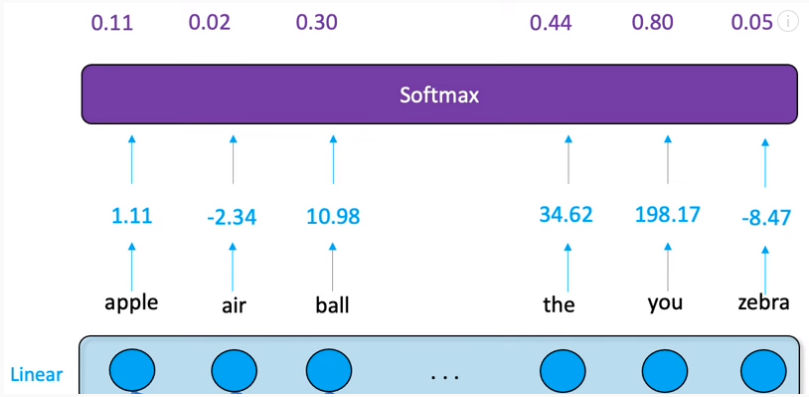


Imagen tomada de Haider, 2021.

Masked multi-head attention module

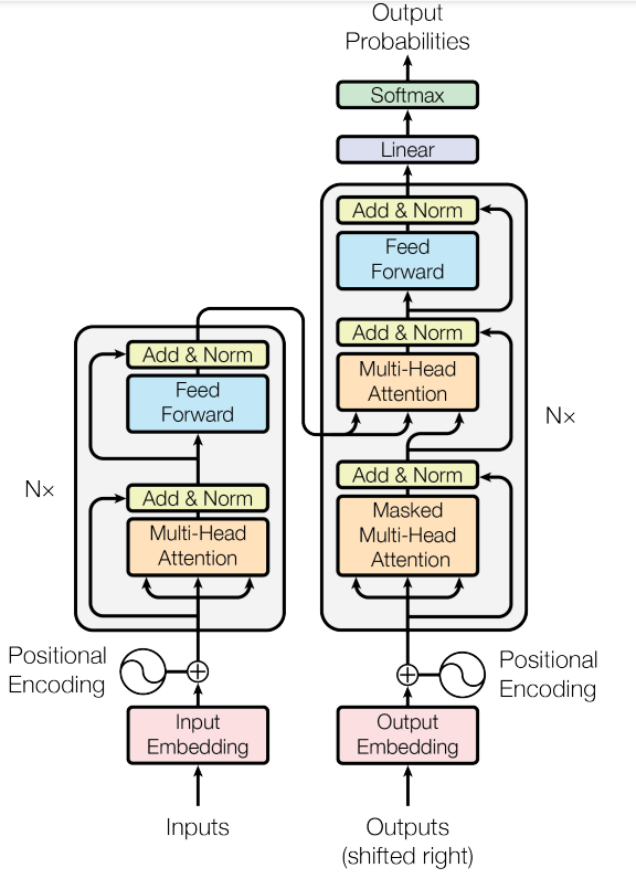


Imagen tomada de Vaswani et al, 2017.

- Teacher forcing: se cuantifica la diferencia usando *cross-entropy loss*.

Masked multi-head attention module

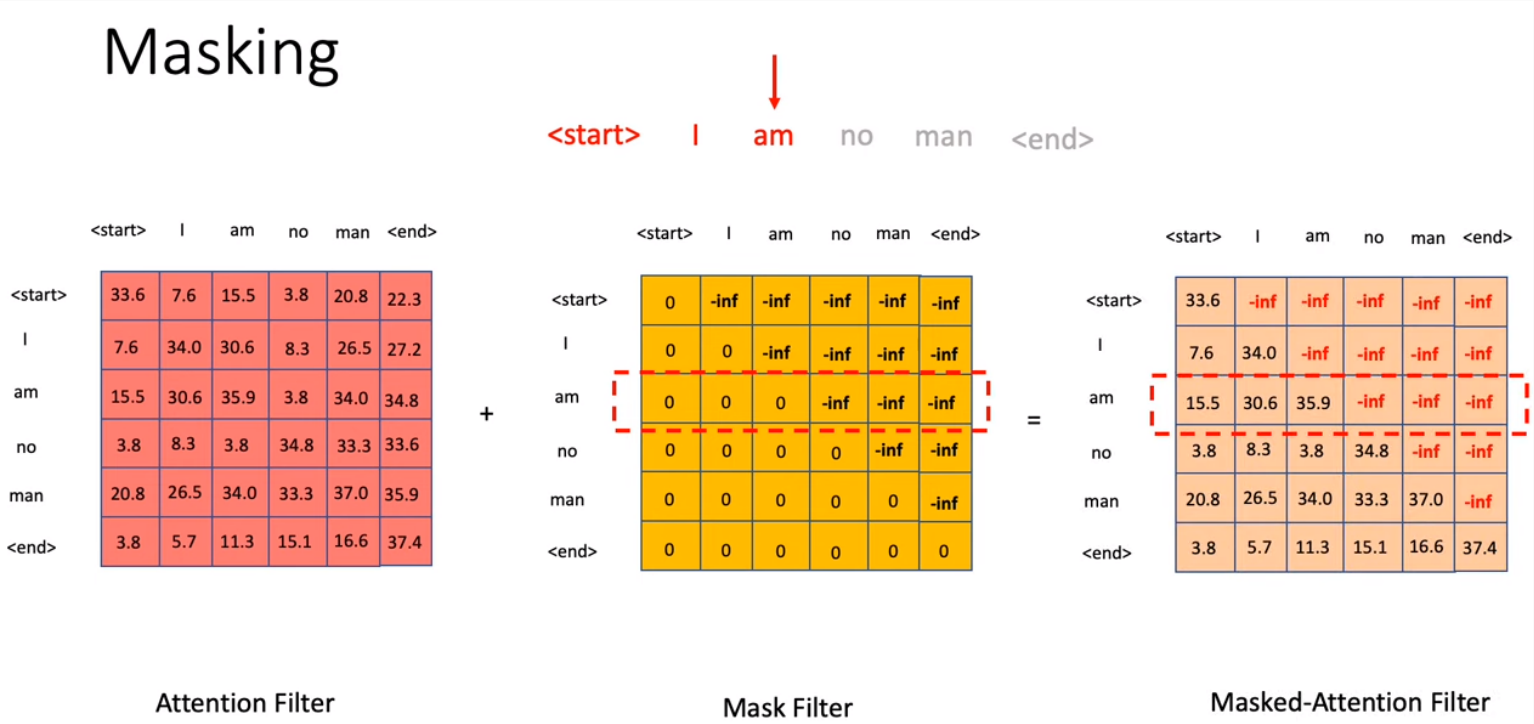


Imagen tomada de Haider, 2021.

Masked multi-head attention module

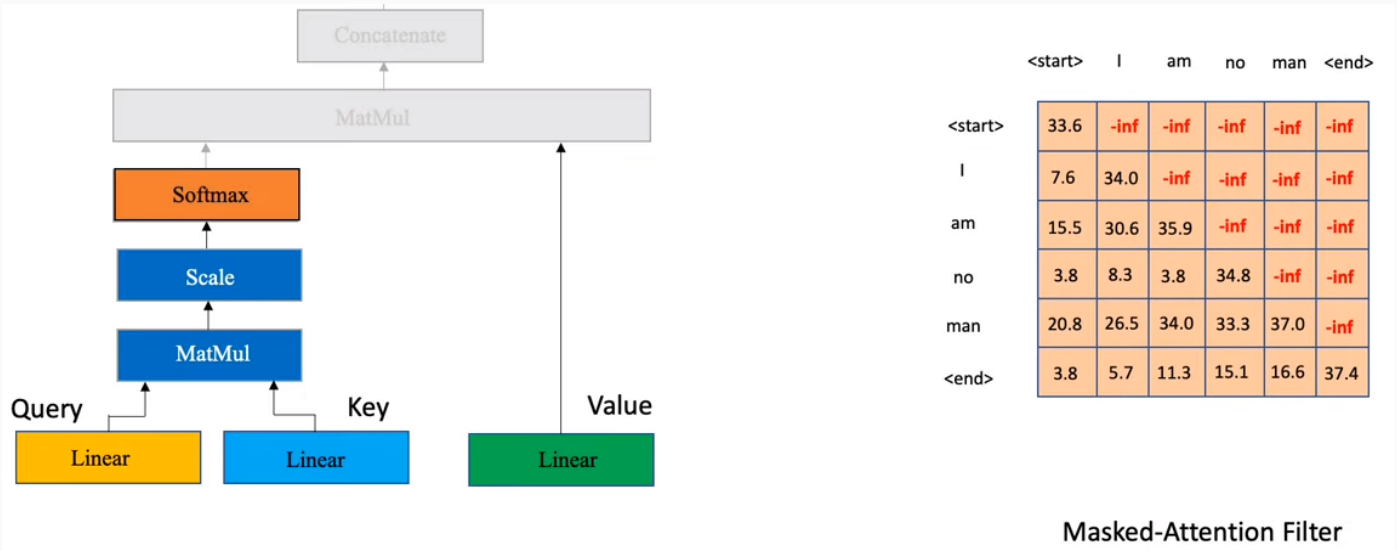


Imagen tomada de Haider, 2021.

Masked multi-head attention module

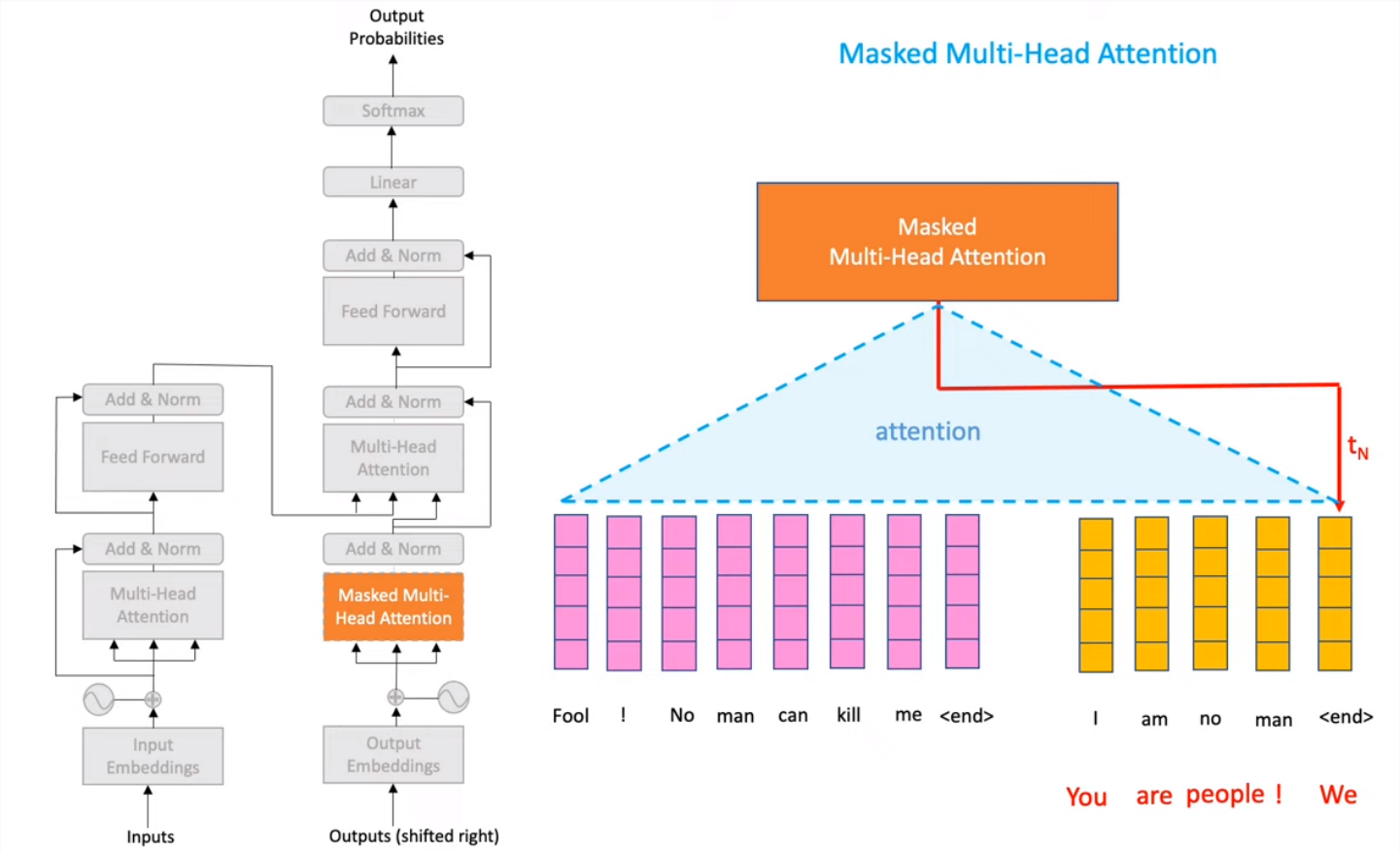


Imagen tomada de Haider, 2021.

Visual transformers

Published as a conference paper at ICLR 2021

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy[*,†], **Lucas Beyer**[*], **Alexander Kolesnikov**[*], **Dirk Weissenborn**[*], **Xiaohua Zhai**[*], **Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby**[*,†]

[*]equal technical contribution, [†]equal advising

Google Research, Brain Team

{adosovitskiy, neilhoulsby}@google.com

Arquitectura Visual transformers

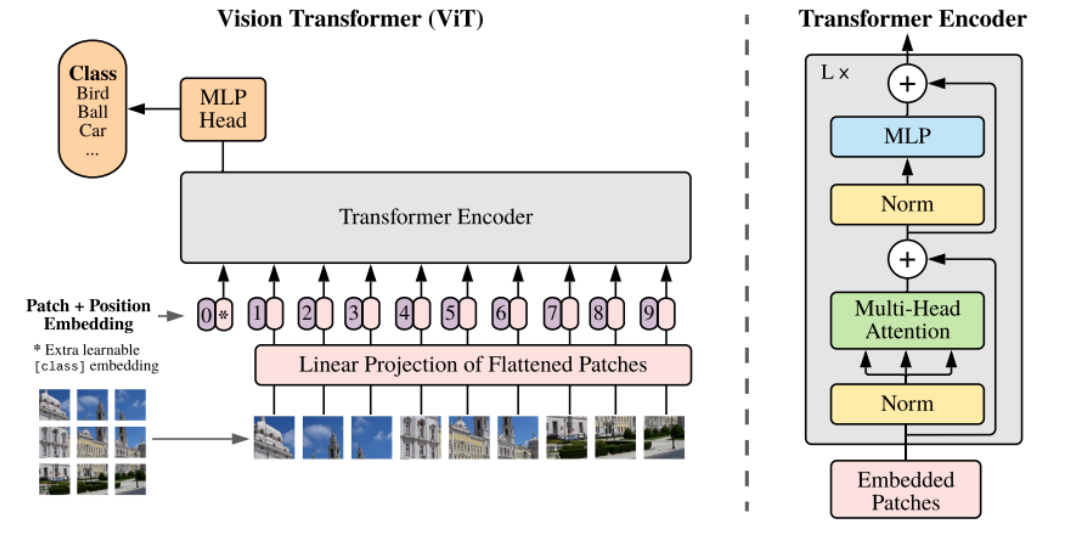


Imagen tomada de Dosovitskiy, 2021.

Arquitectura Visual transformers

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.

Imagen tomada de Dosovitskiy, 2021.

Rendimiento

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Table 2: Comparison with state of the art on popular image classification benchmarks. We report mean and standard deviation of the accuracies, averaged over three fine-tuning runs. Vision Transformer models pre-trained on the JFT-300M dataset outperform ResNet-based baselines on all datasets, while taking substantially less computational resources to pre-train. ViT pre-trained on the smaller public ImageNet-21k dataset performs well too. *Slightly improved 88.5% result reported in Touvron et al. (2020).

Imagen tomada de Dosovitskiy, 2021.

Atención

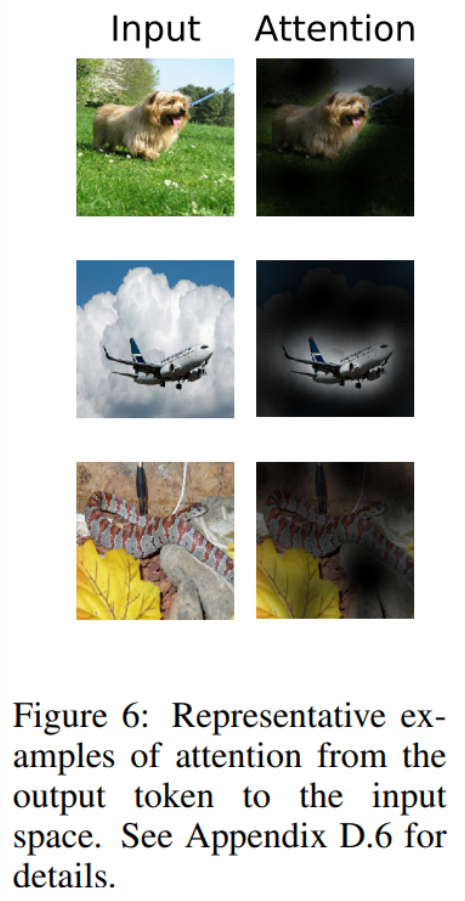


Figure 6: Representative examples of attention from the output token to the input space. See Appendix D.6 for details.

Imagen tomada de Dosovitskiy, 2021.

Atención

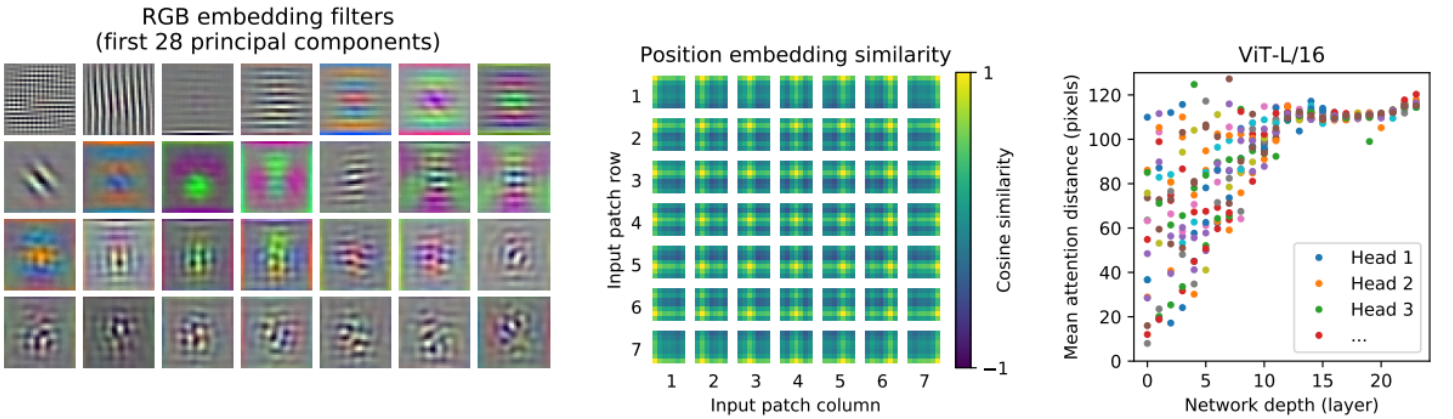


Figure 7: **Left:** Filters of the initial linear embedding of RGB values of ViT-L/32. **Center:** Similarity of position embeddings of ViT-L/32. Tiles show the cosine similarity between the position embedding of the patch with the indicated row and column and the position embeddings of all other patches. **Right:** Size of attended area by head and network depth. Each dot shows the mean attention distance across images for one of 16 heads at one layer. See Appendix D.6 for details.

Imagen tomada de Dosovitskiy, 2021.

Atención

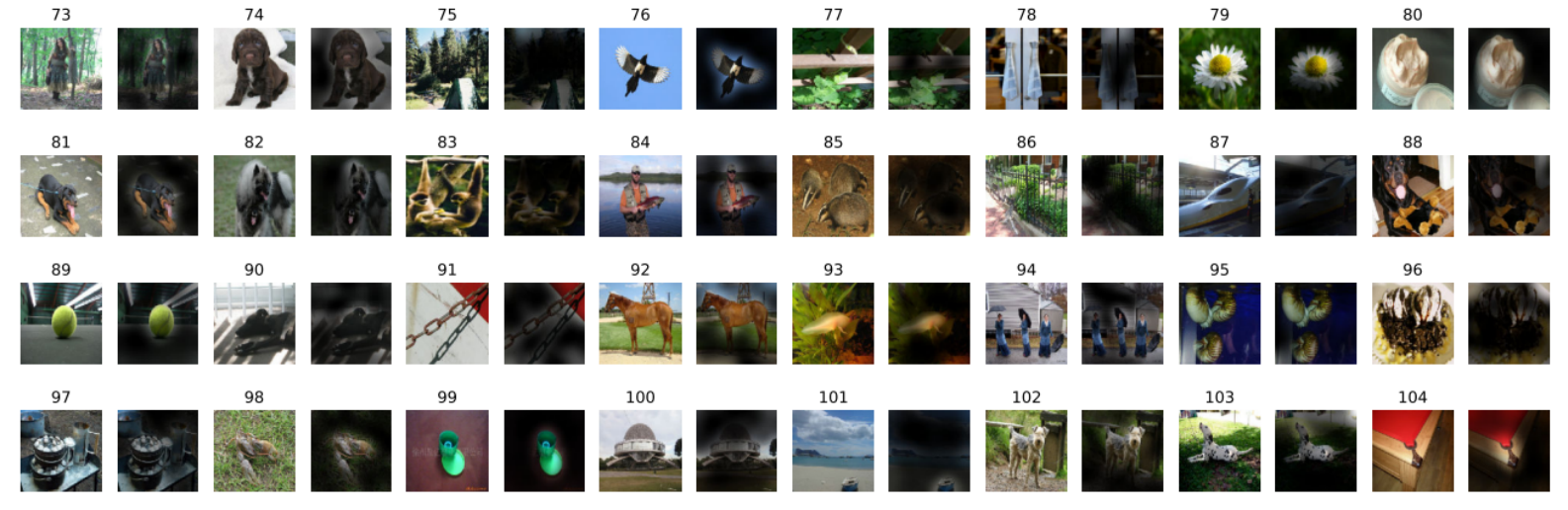


Imagen tomada de Dosovitskiy, 2021.

Vision-Transformer Keras Tensorflow Pytorch Examples

https://github.com/ashishpatel26/Vision-Transformer-Keras-Tensorflow-Pytorch-Examples/tree/main

Bibliografía

- Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv, 2016.
- Azad, R., Asadi-Aghbolaghi, M., Fathy, M., Escalera, S. (2020). Attention Deeplabv3+: Multi-level Context Attention Mechanism for Skin Lesion Segmentation. In: Bartoli, A., Fusiello, A. (eds) Computer Vision – ECCV 2020 Workshops. ECCV 2020. Lecture Notes in Computer Science(), vol 12535. Springer, Cham.
- Jing Li, Kan Jin, Dalin Zhou, Naoyuki Kubota, Zhaojie Ju, Attention mechanism-based CNN for facial expression recognition, Neurocomputing, Volume 411, 2020, Pages 340-350, ISSN 0925-2312.
- Batool Haider, Visual Guide to Transformer Neural Networks, 2021.

Bibliografía

- Ashish Vaswani and Noam Shazeer and Niki Parmar and Jakob Uszkoreit and Llion Jones and Aidan N. Gomez and Lukasz Kaiser and Illia Polosukhin, Attention Is All You Need, arXiv, 2017.

Ecuaciones para medir la similitud

$$Dot\_product = S_t^T \bar{h}_t$$

$$Concat = v_a^T tanh(W_a[S_t^T \bar{h}_t])$$

$$Location = softmax(W_a S_t)$$