

## Redes recurrentes

Jocelyn Dunstan Escudero

jdunstan@uc.cl

Departamento de Ciencia de la Computación  
& Instituto de Matemática Computacional

Pontificia Universidad Católica de Chile



17 de noviembre de 2025

- Introducir las redes recurrentes y entender sus limitaciones
- Comprender el concepto de atención
- Conocer los transformers



# Entregar texto a una red feed forward

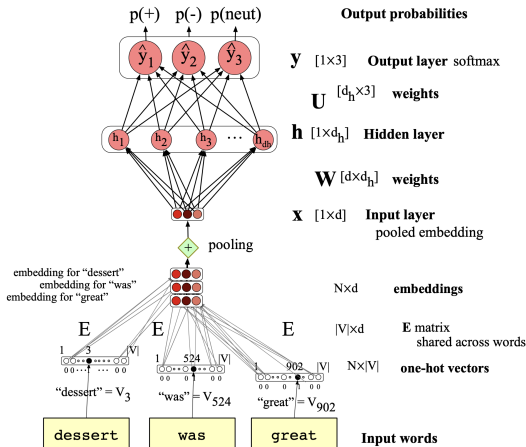


Multiplicando un vector *one-hot* por una matriz de embedding, obtenemos el vector de esa palabra

$$\begin{matrix} & 3 & & |V| \\ 1 & \boxed{0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \dots 0 \ 0 \ 0 \ 0} \end{matrix} \times \begin{matrix} & d \\ 3 & \boxed{\phantom{0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \dots 0 \ 0 \ 0 \ 0}} \\ & |V| \end{matrix} \begin{matrix} E \\ \\ \end{matrix} = \begin{matrix} & d \\ 1 & \boxed{\phantom{0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \dots 0 \ 0 \ 0 \ 0}} \end{matrix}$$



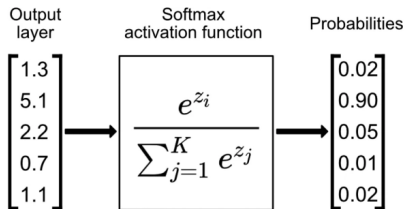
# Clasificando el sentimiento de una frase promediando vectores



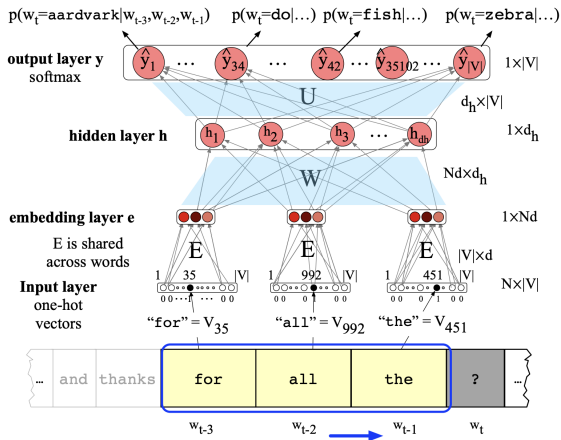
<https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf>



# Recordemos que era la softmax



# Predecir la siguiente palabra concatenando vectores



**¿Qué pasa cuando la  
entrada no tiene un  
tamaño fijo?**

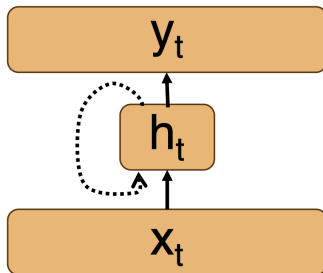




# Redes recurrentes



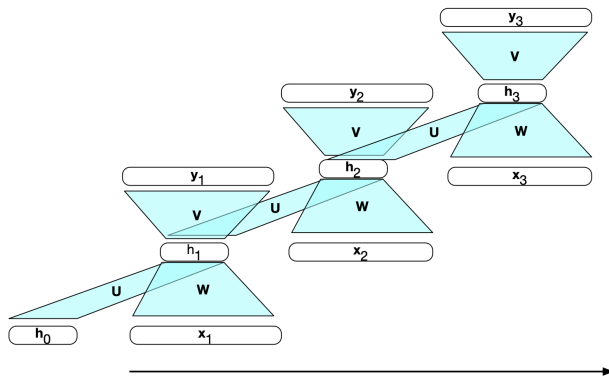
# Red recurrente



<https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf>



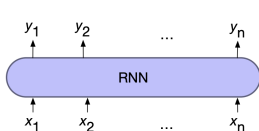
# Red recurrente desenrollada (*unrolled*)



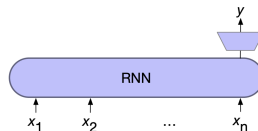
<https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf>



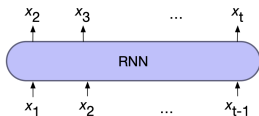
# Distintas tareas que pueden hacer las redes recurrentes



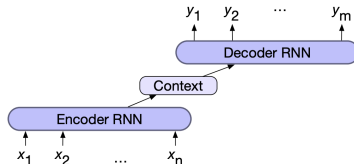
a) sequence labeling



b) sequence classification



c) language modeling

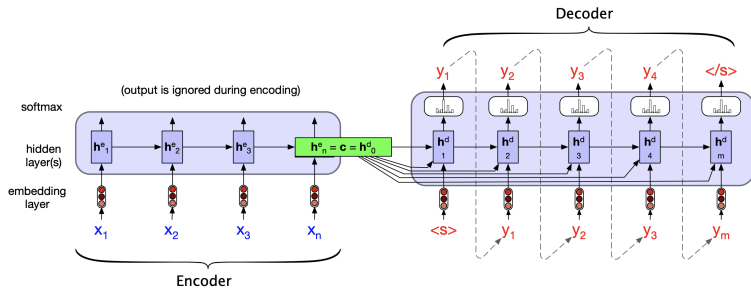


d) encoder-decoder

<https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf>



# Encoder-decoder



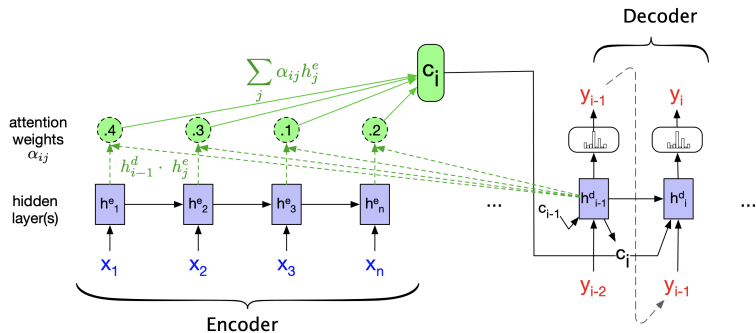
<https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf>



# Attention



# Attention



<https://web.stanford.edu/~jurafsky/slp3/slides/vectorsemantics2024.pdf>



# Mecanismo de Atención: Cálculo de Relevancia

Crearemos un puntaje que nos indique cuánto debemos enfocarnos en cada estado del codificador, es decir, cuán *relevante* es cada estado del codificador para el estado del decodificador:

$$\text{score}(h_{i-1}^d, h_j^e) = h_{i-1}^d \cdot h_j^e$$





# Mecanismo de Atención: Cálculo de Relevancia

Crearemos un puntaje que nos indique cuánto debemos enfocarnos en cada estado del codificador, es decir, cuán *relevante* es cada estado del codificador para el estado del decodificador:

$$\text{score}(h_{i-1}^d, h_j^e) = h_{i-1}^d \cdot h_j^e$$

Normalizamos estos puntajes con un *softmax* para crear pesos  $\alpha_{i,j}$ , que indican la relevancia del estado oculto  $j$  del codificador para el estado oculto  $h_{i-1}^d$  del decodificador:

$$\alpha_{i,j} = \text{softmax}(\text{score}(h_{i-1}^d, h_j^e))$$



# Mecanismo de Atención: Cálculo de Relevancia

Crearemos un puntaje que nos indique cuánto debemos enfocarnos en cada estado del codificador, es decir, cuán *relevante* es cada estado del codificador para el estado del decodificador:

$$\text{score}(h_{i-1}^d, h_j^e) = h_{i-1}^d \cdot h_j^e$$

Normalizamos estos puntajes con un *softmax* para crear pesos  $\alpha_{i,j}$ , que indican la relevancia del estado oculto  $j$  del codificador para el estado oculto  $h_{i-1}^d$  del decodificador:

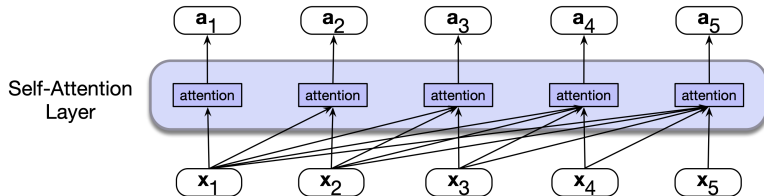
$$\alpha_{i,j} = \text{softmax}(\text{score}(h_{i-1}^d, h_j^e))$$

Luego usamos estos pesos para crear un promedio ponderado:

$$c_i = \sum_j \alpha_{i,j} h_j^e$$



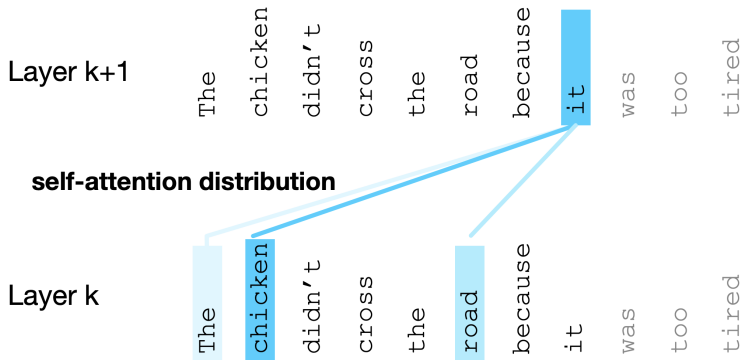
# Attention layer



<https://web.stanford.edu/~jurafsky/slp3/slides/transformer25aug.pdf>



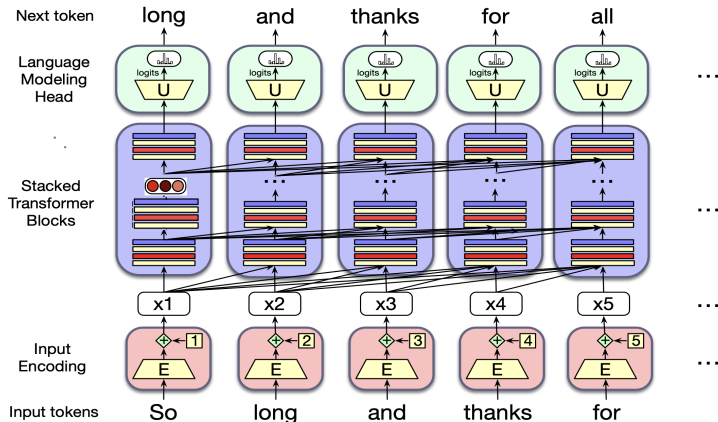
# Attention layer para saber en qué palabra fijarme



<https://web.stanford.edu/~jurafsky/slp3/slides/transformer25aug.pdf>



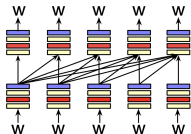
# Transformers: Attention is all you need



<https://web.stanford.edu/~jurafsky/slp3/slides/transformer25aug.pdf>

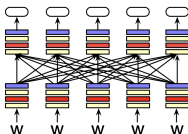


# Formas en que podemos usar los transformers



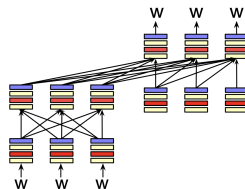
## Decoders

GPT, Claude,  
Llama  
Mixtral



## Encoders

BERT family,  
HuBERT



## Encoder-decoders

Flan-T5, Whisper

<https://web.stanford.edu/~jurafsky/slp3/slides/11m25aug.pdf>

