Encoder-Decoder con Mecanismo de Atención

Orlando Ramos Flores

Contenido

- Encoder-Decoder Básico
  - Problemas del vector de contexto
- Mecanismo de Atención
  - Atención de producto punto
  - Encoder-Decoder con atención
-

Encoder-Decoder

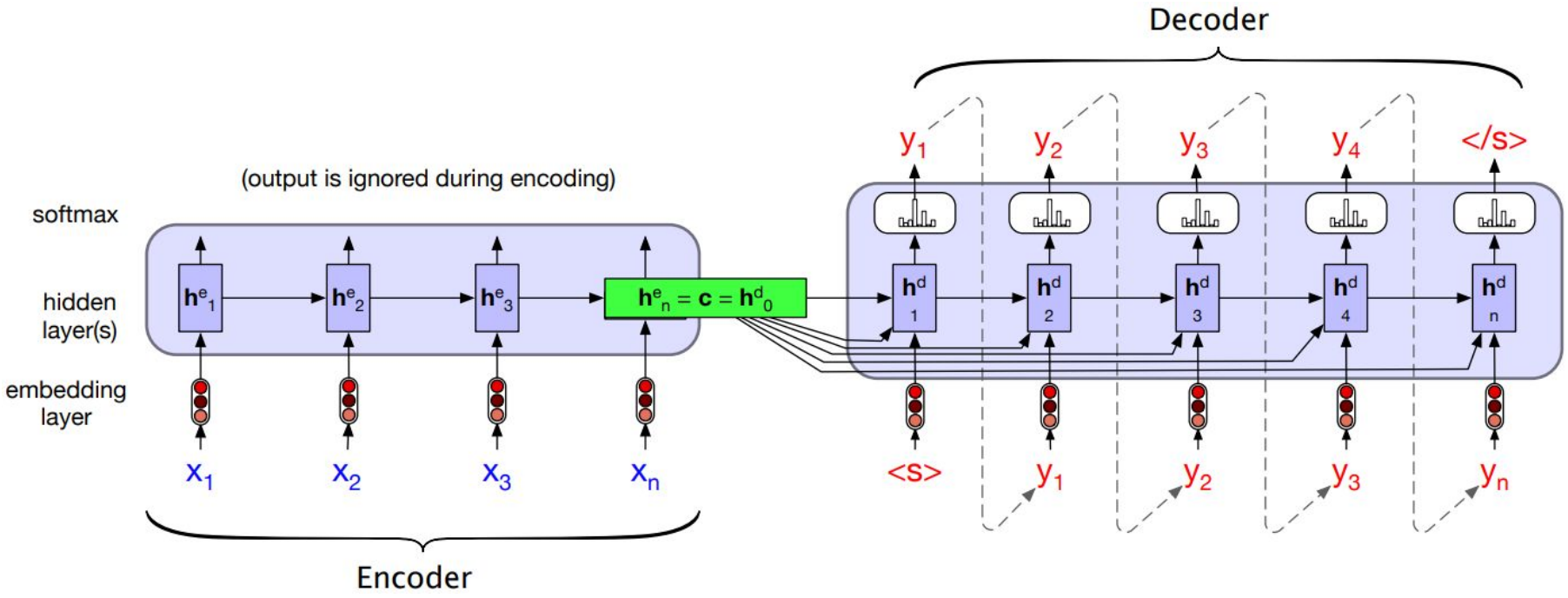
Encoder-Decoder Básico

- La simplicidad del modelo Encoder-Decoder es su clara separación del **encoder** (que crea una representación del texto de origen) del **decoder**, que utiliza este *contexto* para generar un texto de destino.

- En este modelo el *vector de contexto* es h_n, el último (n-ésimo) estado oculto del paso de tiempo del texto fuente.

Jurafsky, D., & Martin, J. H. (2021). Speech and language processing. Vol. 3. *US: Prentice Hall*.

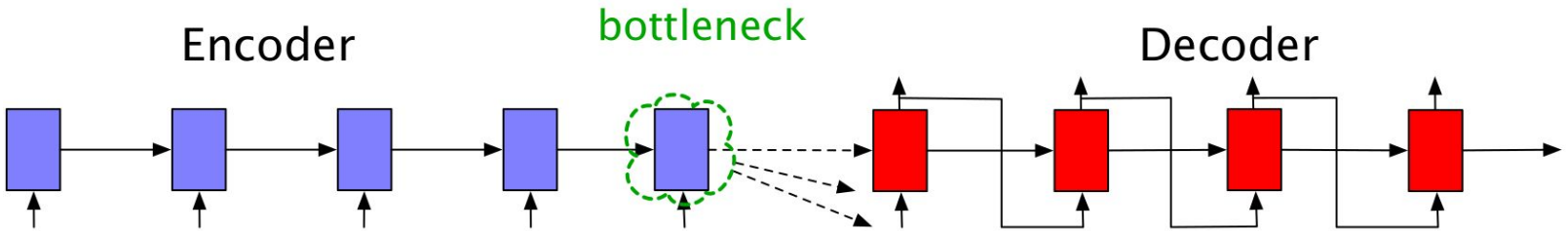
Encoder-Decoder Básico: Arquitectura



Jurafsky, D., & Martin, J. H. (2021). Speech and language processing. Vol. 3. *US: Prentice Hall*.

Encoder-Decoder Básico

- Este estado oculto final actúa como un cuello de botella: debe representar absolutamente todo sobre el significado del texto fuente, ya que lo único que sabe el decoder sobre el texto fuente es lo que hay en este *vector de contexto*.



Jurafsky, D., & Martin, J. H. (2021). Speech and language processing. Vol. 3. *US: Prentice Hall*.

Encoder-Decoder Básico

- La información al comienzo de la oración, especialmente para oraciones largas, puede no estar igualmente bien representada en el *vector de contexto*.
- El **mecanismo de atención** es una solución al problema del cuello de botella, una forma de permitir que el **decoder** obtenga información de todos los estados ocultos del **encoder**, no solo del último estado oculto.

Jurafsky, D., & Martin, J. H. (2021). Speech and language processing. Vol. 3. *US: Prentice Hall*.

Mecanismo de Atención

Mecanismo de Atención

- Para abordar este problema, *Bahdanau et al.* presentaron una extensión del modelo Encoder-Decoder que aprende a *alinear y traducir conjuntamente*.
- Cada vez que el modelo propuesto genera una palabra en una traducción, busca (suavemente) un conjunto de posiciones en una oración fuente donde se concentra la información más *relevante*.
- Luego, el modelo predice una palabra objetivo en función de los *vectores de contexto* asociados con estas posiciones de origen y todas las palabras objetivo generadas anteriormente.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Mecanismo de Atención

- La característica distintiva más importante de este enfoque (Mecanismo de Atención) del Encoder-Decoder básico es que no intenta codificar una oración de entrada completa en un solo vector de longitud fija.
- En cambio, codifica la oración de entrada en una secuencia de vectores y elige un subconjunto de estos vectores de forma adaptativa mientras decodifica la traducción.
- Esto libera a un modelo de traducción neuronal de tener que comprimir toda la información de una oración fuente, independientemente de su longitud, en un vector de longitud fija.
- Además permite que un modelo se adapte mejor a oraciones largas.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Mecanismo de Atención

- La idea de la **atención** es crear el vector de longitud fija c tomando una suma ponderada de todos los estados ocultos del **encoder**.
- Los pesos se enfocan en (“atender a”) una parte particular del texto fuente que es *relevante* para el token que el **decoder** está produciendo actualmente.
- Por lo tanto, la **atención** reemplaza el *vector de contexto estático* con uno que se deriva *dinámicamente* de los estados ocultos del **encoder**, diferentes para cada token en la decodificación.

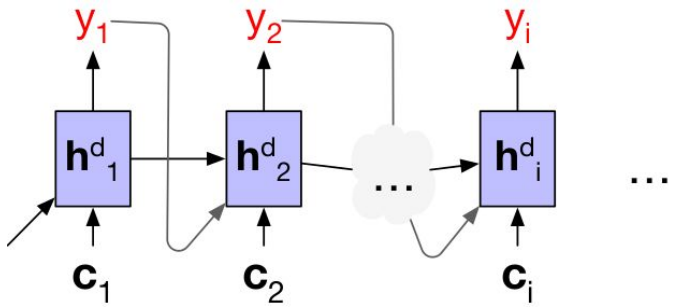
Jurafsky, D., & Martin, J. H. (2021). Speech and language processing. Vol. 3. *US: Prentice Hall*.

Mecanismo de Atención

- En el mecanismo de atención, como en el modelo Encoder-Decoder básico, el *vector de contexto* c es un vector que es una función de los estados ocultos del **encoder**, es decir, $c = f(h^e_1 \dots h^e_n)$.
- Debido a que la cantidad de estados ocultos varía con el tamaño de la entrada, no se puede usar el tensor completo de los vectores de estado oculto del **encoder** directamente como contexto para el **decoder**.
- Este *vector de contexto*, c_i, se genera de nuevo con cada paso de decodificación i y tiene en cuenta todos los estados ocultos del **encoder** en su derivación.

- Luego, este *vector de contexto* estará disponible durante la decodificación al condicionar el cálculo del estado oculto actual del **decoder**, junto con el estado oculto anterior y la salida anterior generada por el **decoder**, con la ecuación:

$$\mathbf{h}_i^d = g(\hat{y}_{i-1}, \mathbf{h}_{i-1}^d, \mathbf{c}_i)$$



Jurafsky, D., & Martin, J. H. (2021). Speech and language processing. Vol. 3. *US: Prentice Hall*.

Mecanismo de Atención: Dot-product Attention

- El primer paso para calcular el vector c_i es calcular cuánto enfocarse en cada estado del **encoder**, qué tan *relevante* es cada estado del **encoder** para el estado del **decoder** capturado en h^d_{i-1}.
- Se captura la *relevancia* calculando (en cada estado i durante la decodificación) un $score(h^d_{i-1}, h^e_j)$ para cada estado del **encoder** j.
- El *score* más simple de este tipo, llamado atención de producto punto, implementa la relevancia como una similitud: mide cuán similar es el estado oculto del **decoder** a un estado oculto del codificador, calculando el producto punto entre ellos:

$$score(\mathbf{h}^d_{i-1}, \mathbf{h}^e_j) = \mathbf{h}^d_{i-1} \cdot \mathbf{h}^e_j$$

Jurafsky, D., & Martin, J. H. (2021). Speech and language processing. Vol. 3. *US: Prentice Hall*.

Mecanismo de Atención: Dot-product Attention

- El resultado del *score* del producto punto es un escalar que refleja el grado de similitud entre los dos vectores.
- El vector de estas puntuaciones (scores) en todos los estados ocultos del **encoder** nos da la *relevancia* de cada estado del **encoder** para el paso actual del **decoder**.
- Para hacer uso de estas puntuaciones, se normalizan con una *softmax* para crear un vector de pesos α_{ij}, que indica la *relevancia proporcional* de cada estado oculto del **encoder** j con respecto al estado oculto anterior del **decoder** h^{d}_{i-1}.

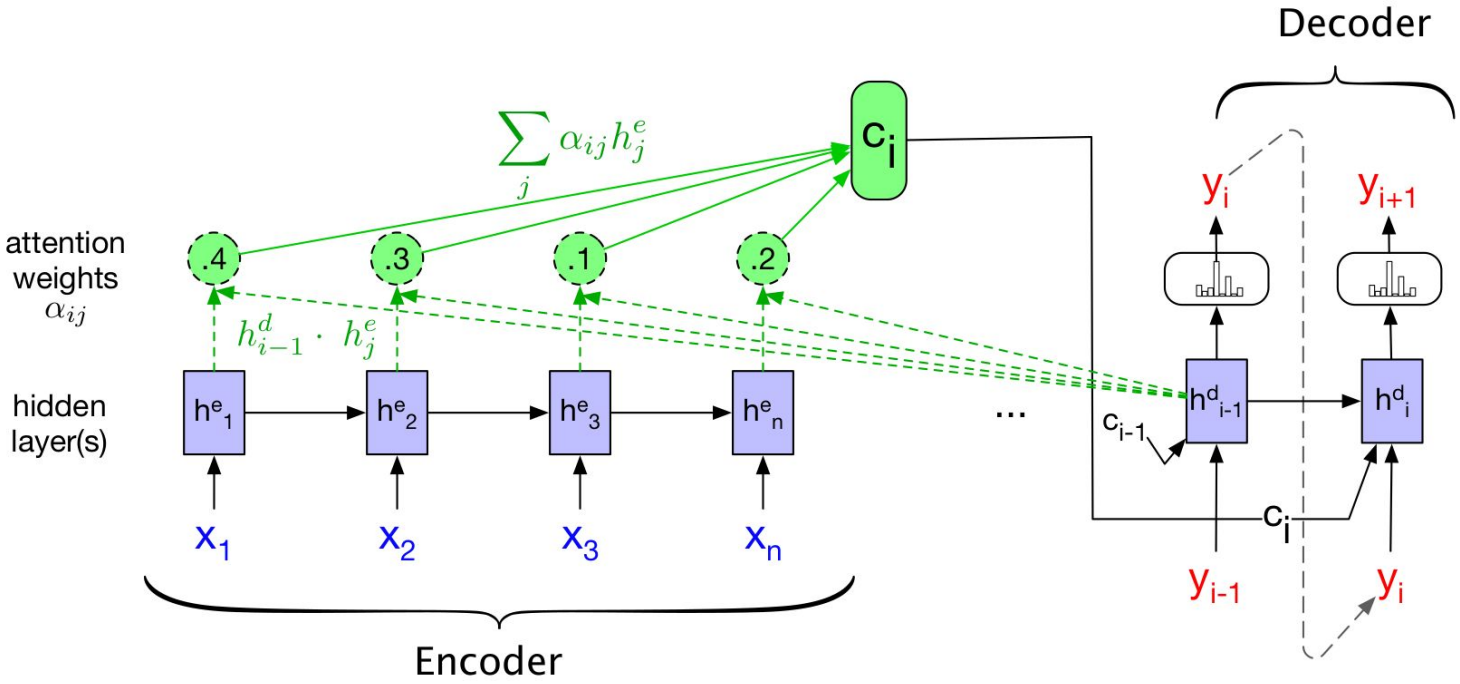
$$\alpha_{ij} = \text{softmax}(score(\mathbf{h}^d_{i-1}, \mathbf{h}^e_j) \ \forall j \in e)$$
$$= \frac{\exp(score(\mathbf{h}^d_{i-1}, \mathbf{h}^e_j)}{\sum_k \exp(score(\mathbf{h}^d_{i-1}, \mathbf{h}^e_k))}$$

- Finalmente, dada la distribución en α, se puede calcular un *vector de contexto* de longitud fija para el estado actual del **decoder** tomando un promedio ponderado de todos los estados ocultos del **encoder**.

$$\mathbf{c}_i = \sum_j \alpha_{ij} \mathbf{h}^e_j$$

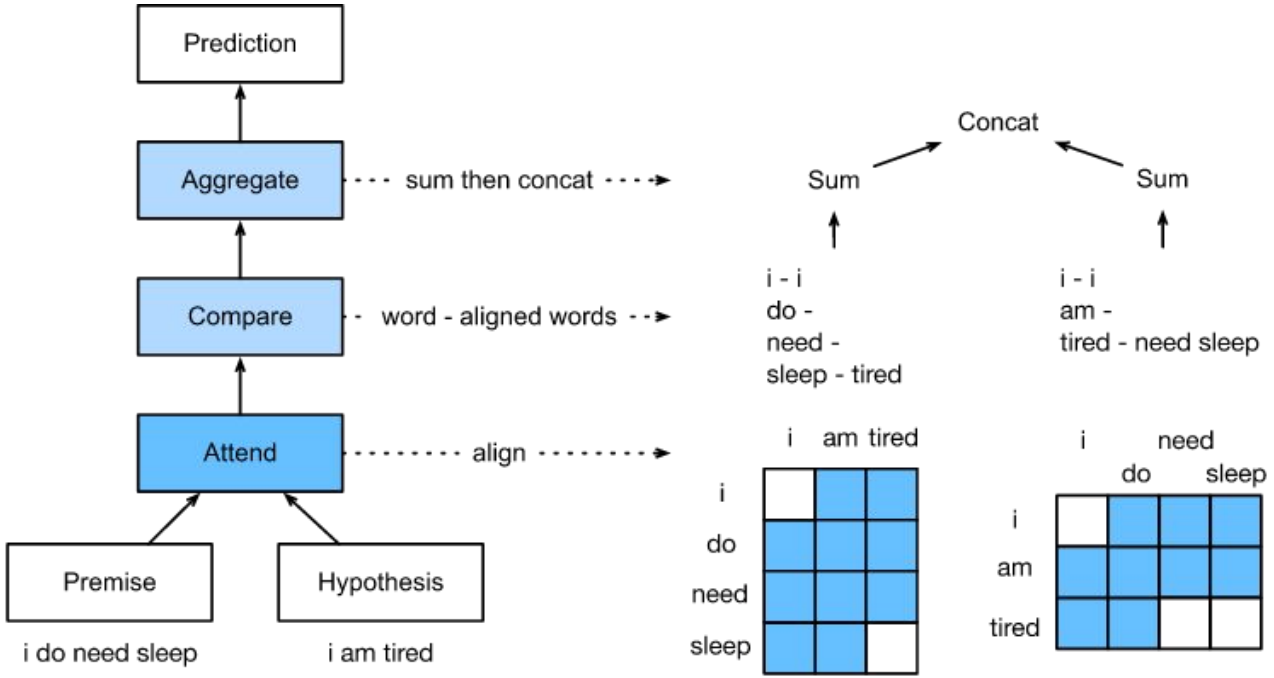
Jurafsky, D., & Martin, J. H. (2021). Speech and language processing. Vol. 3. *US: Prentice Hall*.

Encoder-Decoder con Atención



Jurafsky, D., & Martin, J. H. (2021). Speech and language processing. Vol. 3. *US: Prentice Hall*.

Encoder-Decoder con Atención



- Los cuadrados blancos denotan pesos de atención activos, los cuadrados azules están inactivos.
- En este ejemplo se asume una hard attention.

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. arXiv preprint arXiv:2106.11342.

Encoder-Decoder con Atención: Attending

- El primer paso es alinear los tokens en una secuencia de texto con cada token en la otra secuencia.
- Supongamos que la premisa es "I do need sleep" y la hipótesis es "I am tired".
- Debido a la similitud semántica, es posible alinear "i" en la hipótesis con "i" en la premisa, y alinear "tired" en la hipótesis con "sleep" en la premisa.
- Del mismo modo, podemos desear alinear "i" en la premisa con "i" en la hipótesis, y alinear "need" y "sleep" en la premisa con "tired" en la hipótesis.
- Dicha alineación es suave utilizando un promedio ponderado, donde idealmente se asocian pesos grandes con los tokens que se alinearán.

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. arXiv preprint arXiv:2106.11342.

Encoder-Decoder con Atención: Comparing

- En el siguiente paso, comparar un token en una secuencia con la otra secuencia que está suavemente alineada con ese token.
- En la alineación suave, todos los tokens de una secuencia (aunque probablemente tengan diferentes pesos de atención) se compararán con un token en la otra secuencia.

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. arXiv preprint arXiv:2106.11342.

Encoder-Decoder con Atención: Aggregate

- Con dos conjuntos de vectores de comparación $v_{A,i}$ (i=1,…,m) y $v_{B,j}$ (j=1,…,n), en el último paso, se agrega dicha información para inferir la relación lógica.
- Los vectores se representan como:

$$\mathbf{v}_A = \sum_{i=1}^{m} \mathbf{v}_{A,i}, \quad \mathbf{v}_B = \sum_{j=1}^{n} \mathbf{v}_{B,j}$$

- La concatenación se realiza con ambos resultados de la función **h** para obtener el resultado de clasificación de la relación lógica:

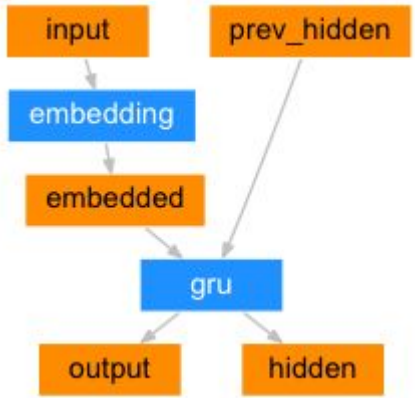
$$\hat{\mathbf{y}} = h([\mathbf{v}_A, \mathbf{v}_B])$$

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into deep learning. arXiv preprint arXiv:2106.11342.

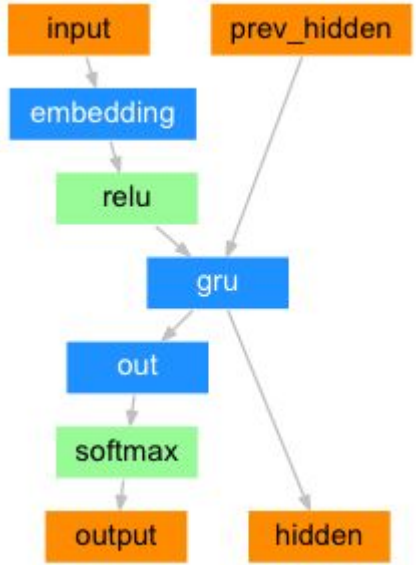
Encoder-Decoder con Atención: Notebook

- Traducción automática del Francés al Inglés

- Encoder

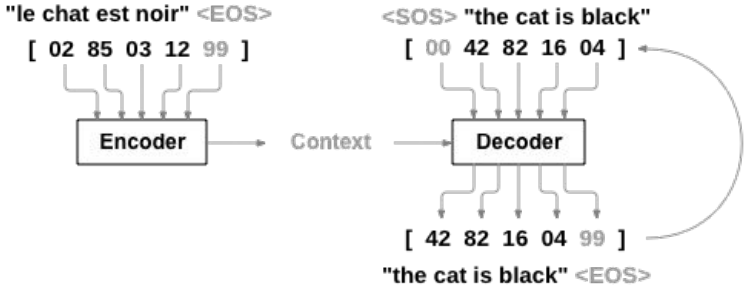


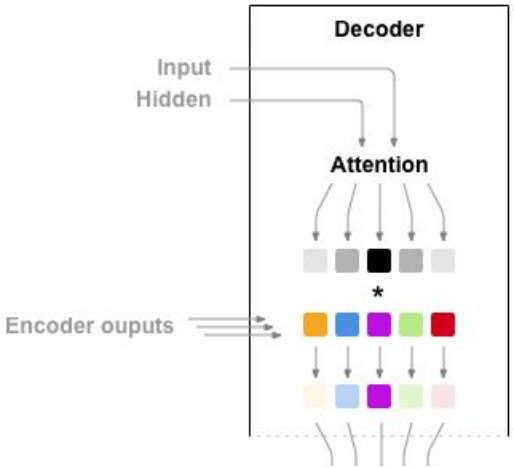
- Decoder

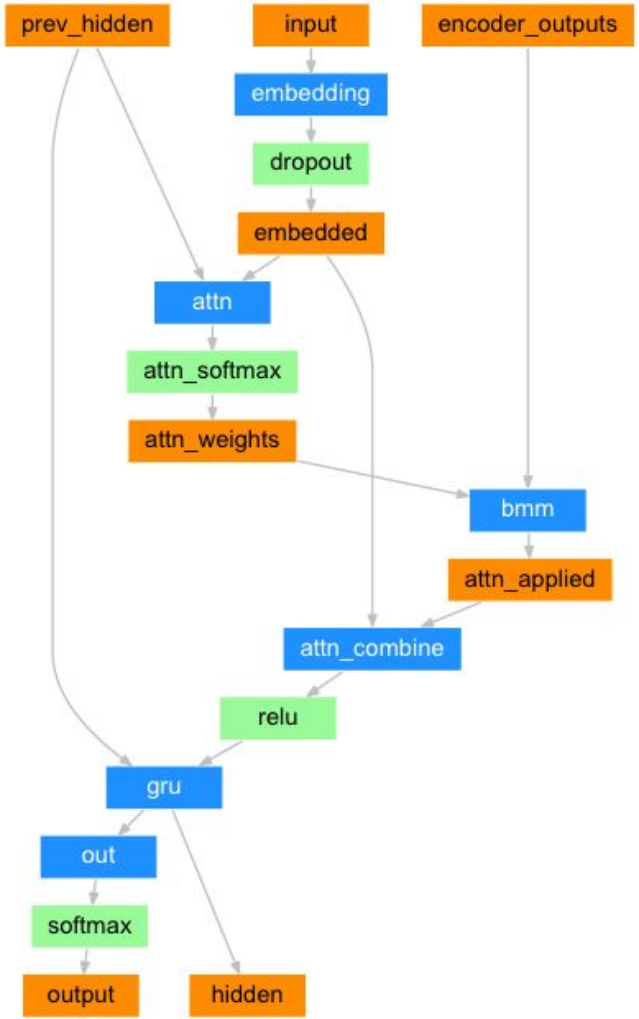


[Translation with a Sequence to Sequence Network and Attention — PyTorch Tutorials]

Encoder-Decoder con Atención: Notebook







[Translation with a Sequence to Sequence Network and Attention — PyTorch Tutorials]