Encoder-Decoder

Orlando Ramos Flores

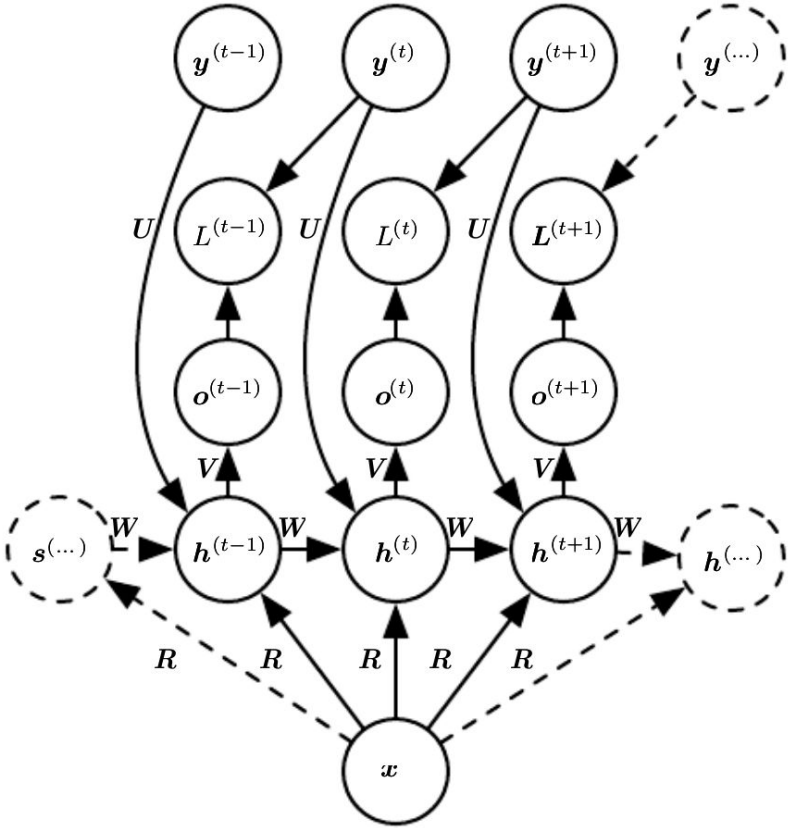
Contenido

- Introducción
  - Vec2seq
  - Seq2Vec
  - Seq2Seq
- Encoder-Decoder
  - Machine Translation
  - Encoder-Decoder

Introducción

Introducción: Vec2Seq (generación de secuencias)

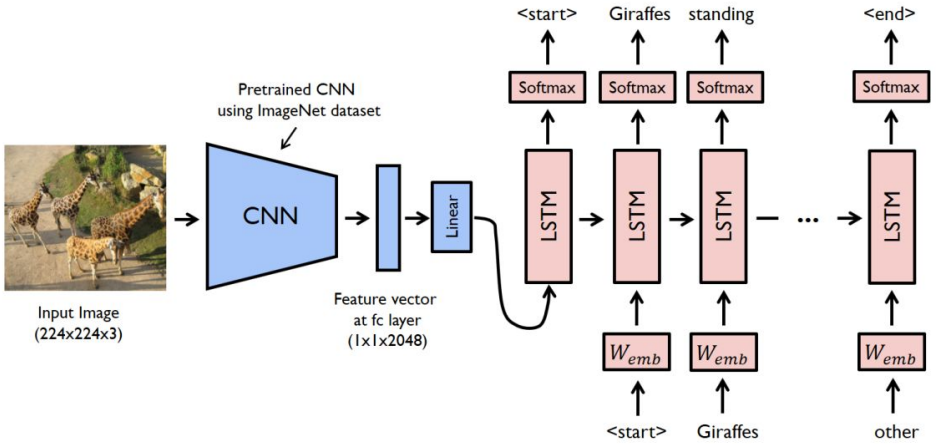
- Un RNN que mapea un vector **x** de longitud fija a una distribución sobre secuencias **Y**.
- Esta RNN es apropiada para tareas como el subtitulado de imágenes, donde una sola imagen se usa como entrada para un modelo que luego produce una secuencia de palabras que describen la imagen.
- Cada elemento $\mathbf{y}^{(t)}$ de la secuencia de salida observada sirve como entrada (para el paso de tiempo actual) **y**, durante el entrenamiento, como objetivo (para el paso de tiempo anterior).



Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

Vec2Seq: Aplicaciones

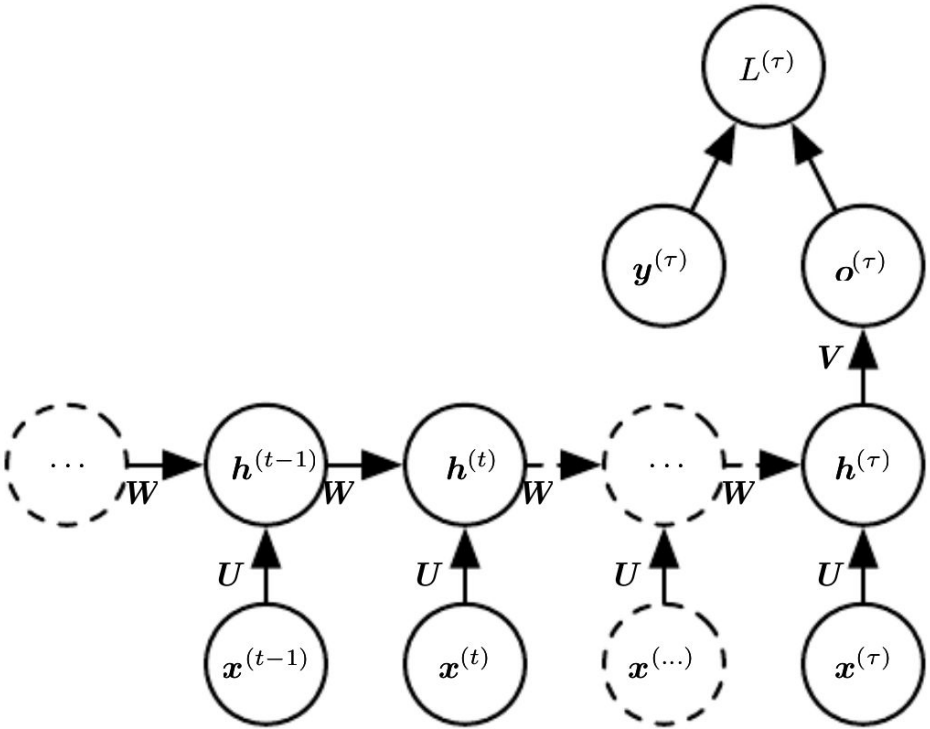
- Generar texto usando una RNN a nivel de caracter. Dado un vector: “amistad”, “el”, “noche” así se calcula el carácter más probable en cada paso y luego se retroalimenta al modelo.
- La Figura de la derecha es un modelo CNN-RNN para subtítulos de imágenes.



Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

Introducción: Seq2Vec (clasificación de secuencias)

- RNN en el tiempo **t** con una sola salida al final de la secuencia.
- Esta red se puede usar para resumir una secuencia y producir una representación de tamaño fijo que se usa como entrada para un procesamiento posterior.
- Puede haber un objetivo justo al final (como se muestra aquí) o el gradiente en la salida $\mathbf{o}^{(t)}$ se puede obtener de la retropropagación desde otros módulos posteriores.



Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

Seq2Vec: aplicaciones

- Clasificación de sentimientos (binaria)

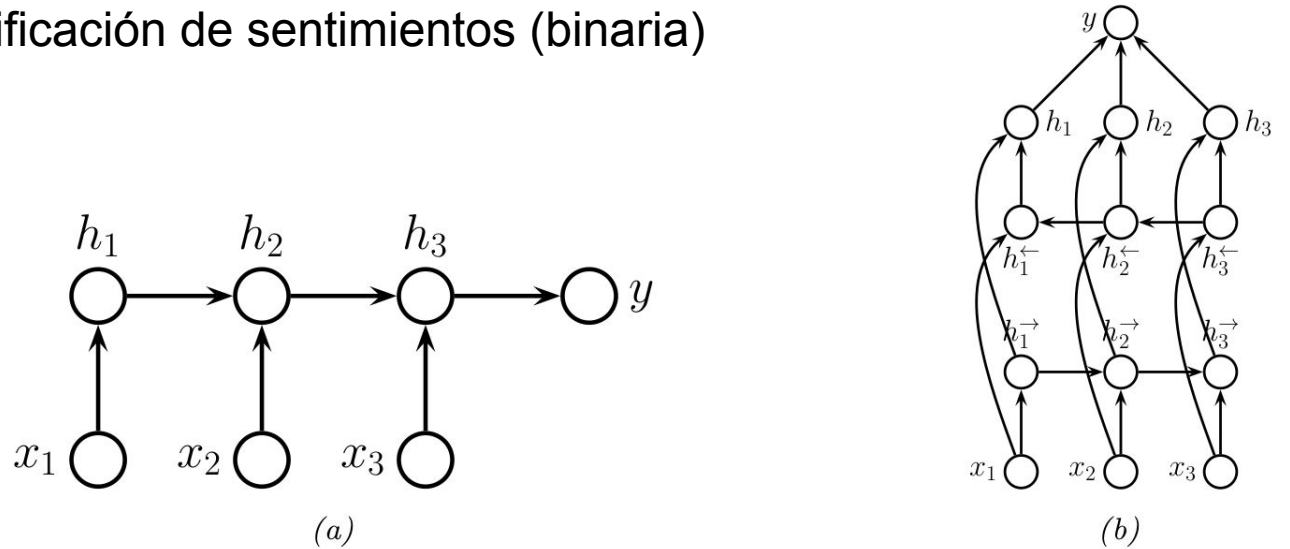
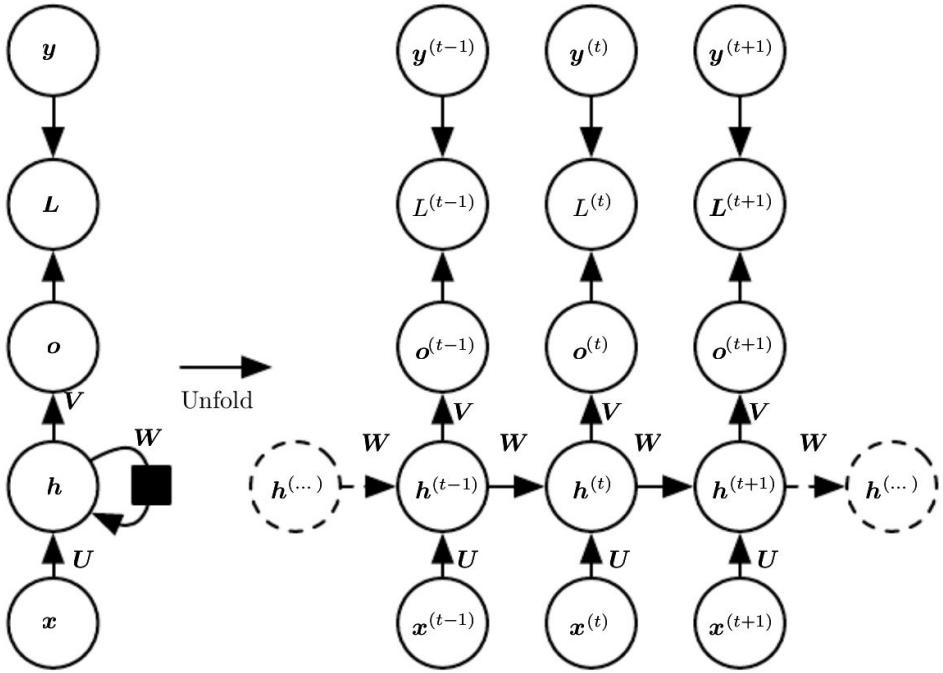


Figure 15.4: (a) RNN for sequence classification. (b) Bi-directional RNN for sequence classification.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

Introducción: Seq2Seq (traducción de secuencias)

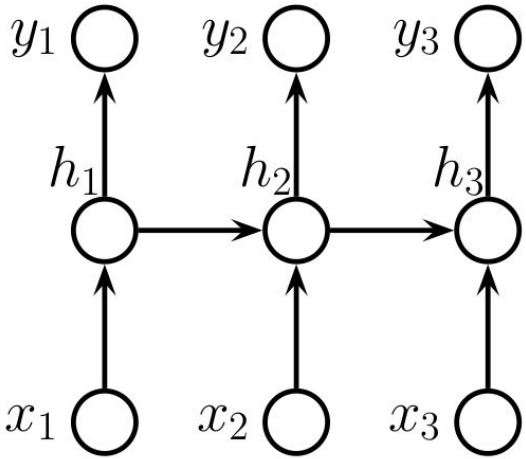
- Se consideran dos casos:
- T’ = T, las secuencias de entrada y salida con la misma longitud, por lo tanto están alineadas.
- T’ ≠ T, las secuencias de entrada y salida tienen diferentes longitudes, esta última es llamado problema **seq2seq**.



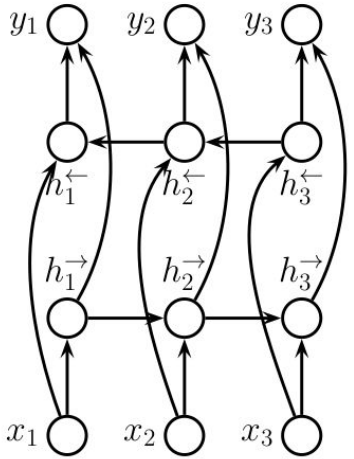
Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

Seq2Seq: aplicaciones - secuencias alineadas

- Reconocimiento de entidades nombradas
- Los vectores de entrada y salida tienen la misma longitud



(a)

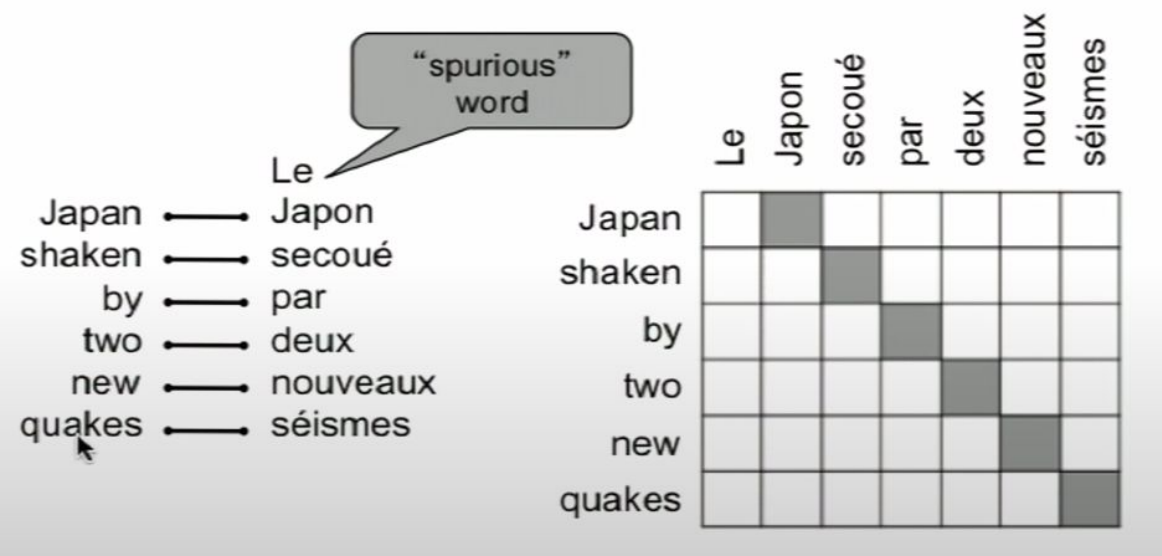


(b)

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
Murphy, K. P. (2022). *Probabilistic machine learning: an introduction*. MIT press.

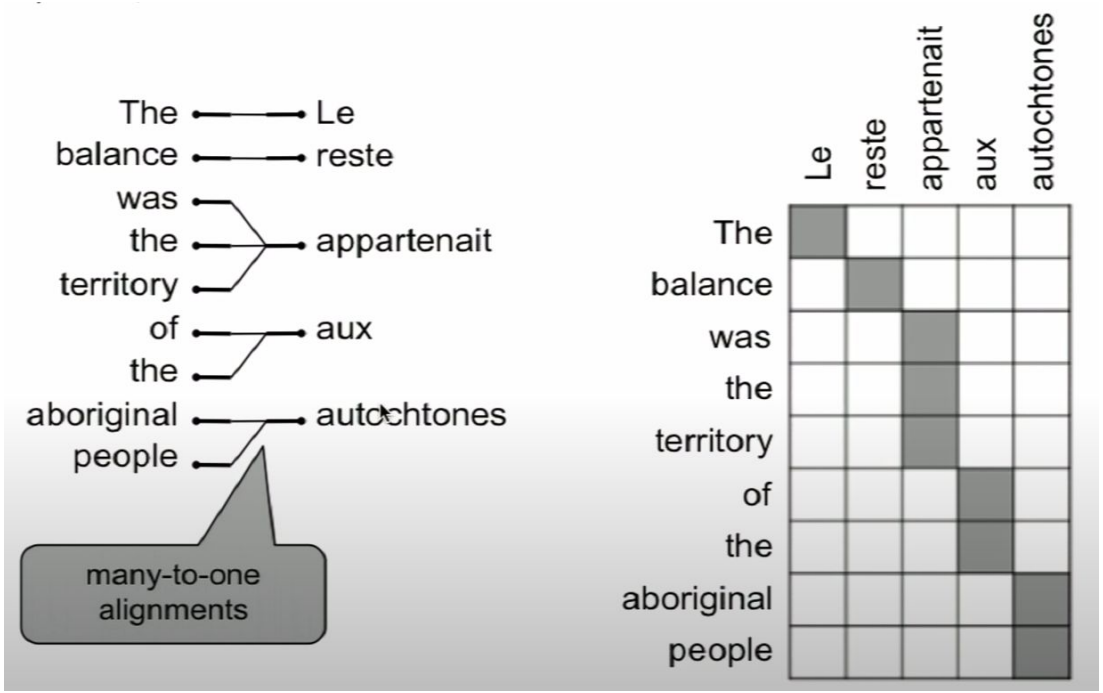
Machine Translation:
¿Cómo alinear los datos (modelos de lenguaje)?

Machine Translation (MT) (Inglés-Francés)



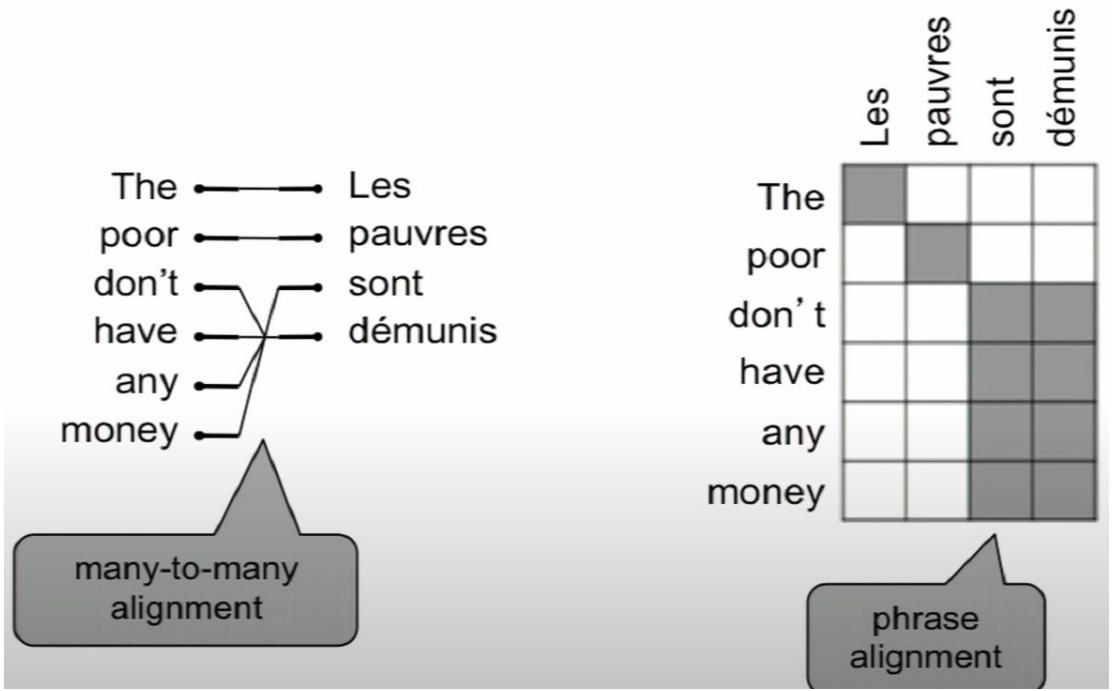
Lecture 9: Machine Translation and Advanced Recurrent LSTMs and GRUs - YouTube

MT: Inglés-Francés



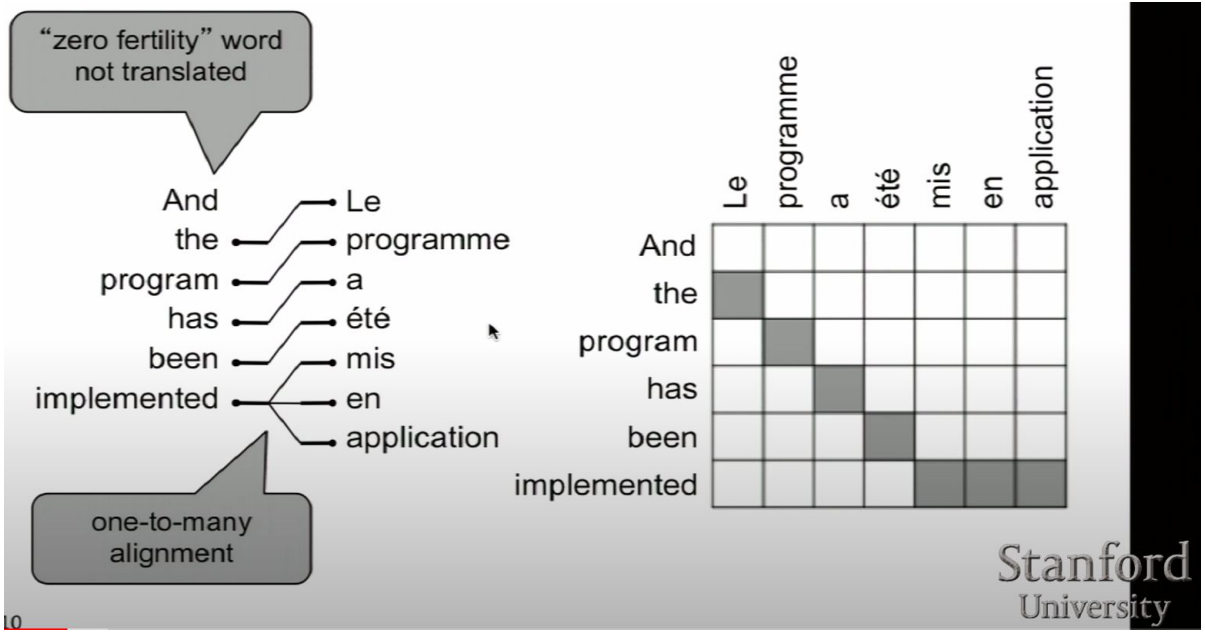
Lecture 9: Machine Translation and Advanced Recurrent LSTMs and GRUs - YouTube

MT: Inglés-Francés



Lecture 9: Machine Translation and Advanced Recurrent LSTMs and GRUs - YouTube

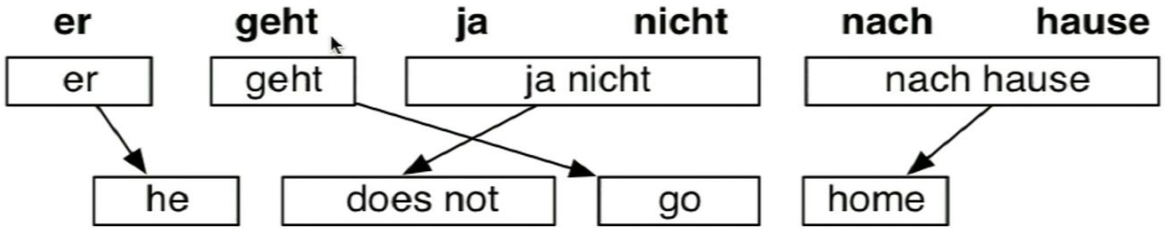
MT: Inglés-Francés



Lecture 9: Machine Translation and Advanced Recurrent LSTMs and GRUs - YouTube

MT: Inglés-Francés

- Podríamos dedicar una conferencia completa a la alineación de modelos
- No solo palabras sueltas, sino que se podrían usar frases, sintaxis, etc.
- También se puede considerar el reordenamiento de las frases traducidas

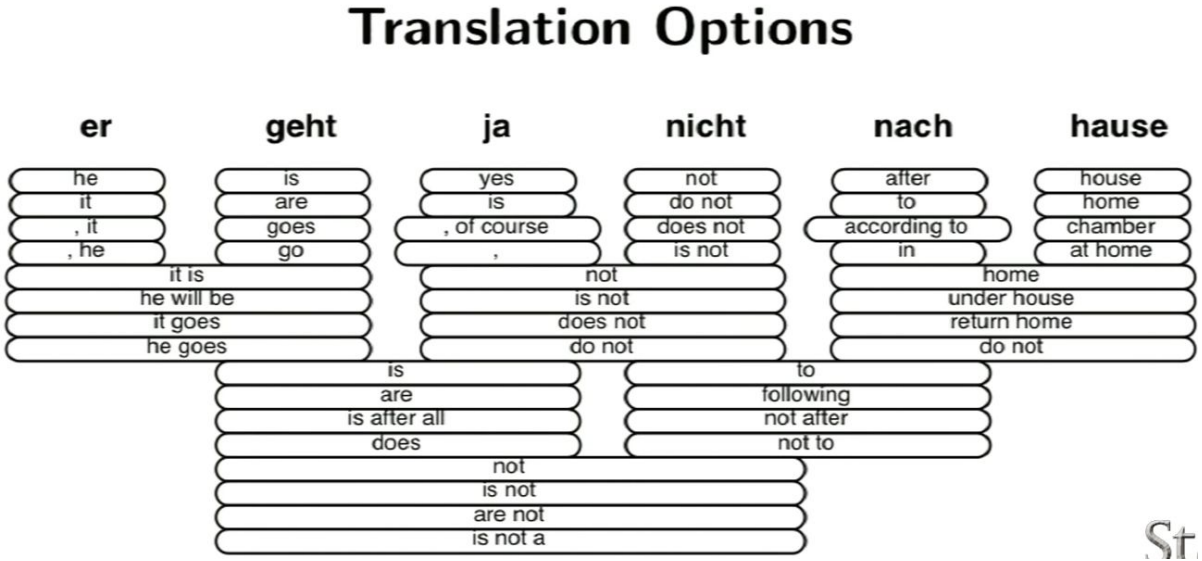


Example from Philipp Koehn

Lecture 9: Machine Translation and Advanced Recurrent LSTMs and GRUs - YouTube

MT: Inglés-Francés

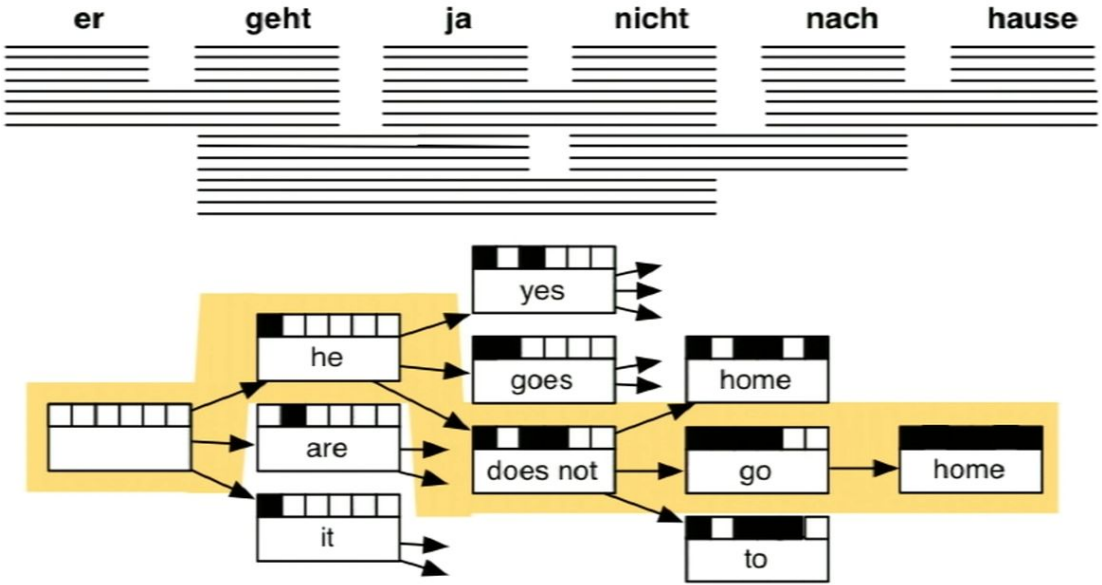
- Cada frase del idioma de origen tiene muchas posibles traducciones, lo que da como resultado un gran espacio de búsqueda.



Lecture 9: Machine Translation and Advanced Recurrent LSTMs and GRUs - YouTube

MT: Inglés-Francés. Buscar la mejor de muchas hipótesis

- Problema de búsqueda difícil que también incluye modelos de lenguaje



Lecture 9: Machine Translation and Advanced Recurrent LSTMs and GRUs - YouTube

Tradicional MT

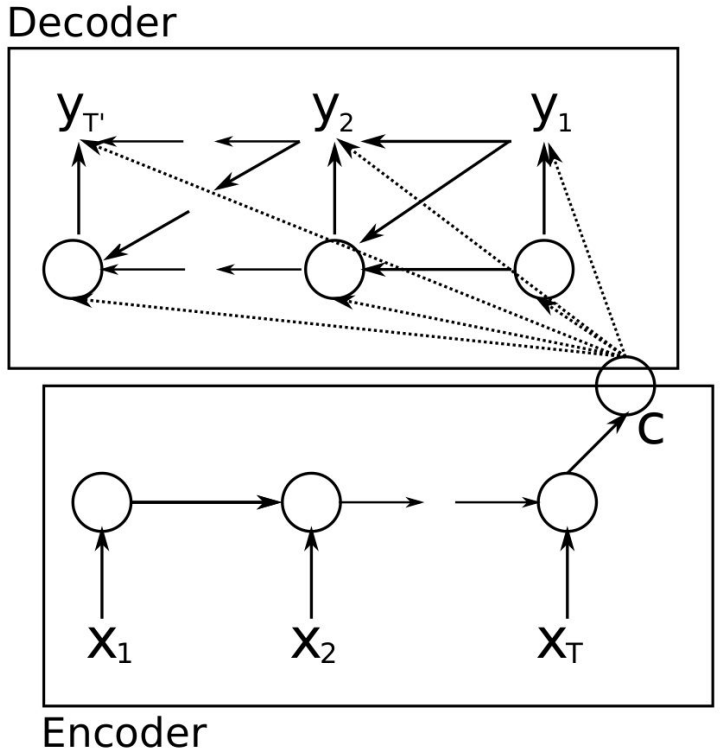
- Se omiten centenares de detalles importantes
- Mucha ingeniería de características hecha a mano
- Sistemas muy complejos
- Muchos problemas diferentes de aprendizaje automático entrenados de forma independiente.

Lecture 9: Machine Translation and Advanced Recurrent LSTMs and GRUs - YouTube

Encoder-Decoder

Encoder-Decoder: Primeros enfoques

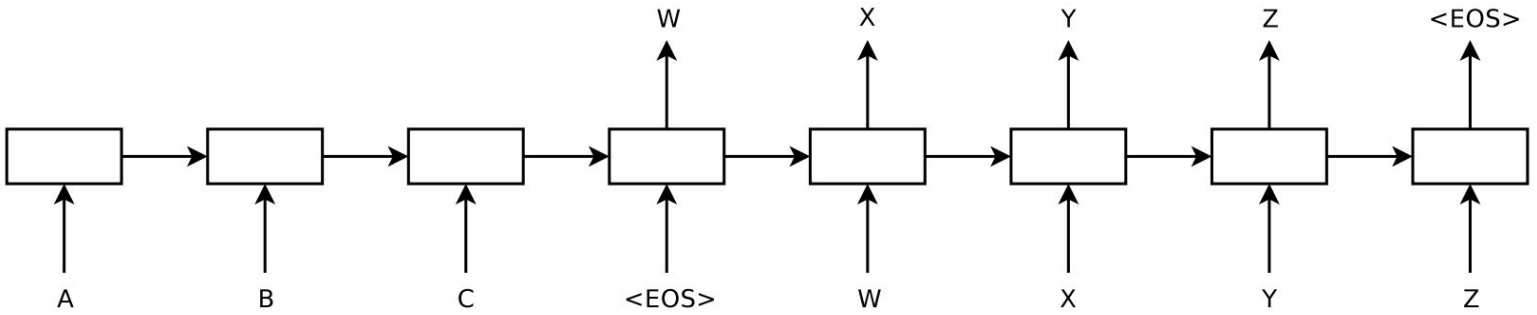
- “Una arquitectura de red neuronal que aprende a **codificar** una secuencia de longitud variable en una representación vectorial de longitud fija y a **decodificar** una representación vectorial de longitud fija determinada en una secuencia de longitud variable”.



Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Encoder-Decoder: Primeros enfoques

- La arquitectura propuesta lee una oración de entrada "ABC" y produce "WXYZ" como oración de salida. El modelo deja de hacer predicciones después de generar el token de final de oración.
- La LSTM lee la oración de entrada al revés porque al hacerlo introduce muchas dependencias a corto plazo en los datos que facilitan mucho el problema de optimización.



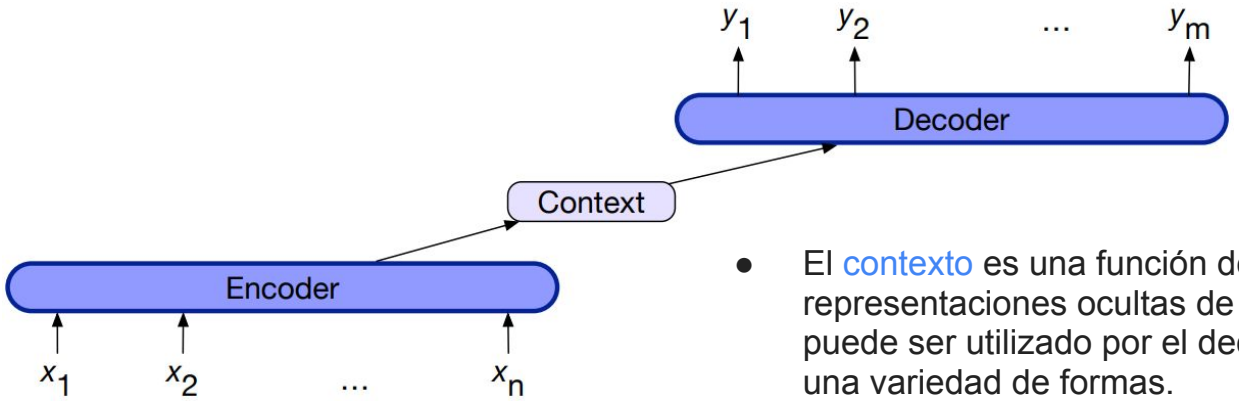
Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, *27*.

Encoder-Decoder (Seq2Seq)

- Las redes de **Encoder-Decoder**, o redes de **Secuencia a Secuencia**, son modelos capaces de generar secuencias de salida de longitud arbitraria apropiadas al contexto.
- Las redes de **Encoder-Decoder** se han aplicado a una amplia gama de aplicaciones que incluyen traducción automática (MT: Machine Translation), resumen (AS: Automatic Summarization), respuesta a preguntas (Q&A: Question Answering) y diálogos.

Encoder-Decoder (Seq2Seq)

- La idea clave que subyace a estas redes es el uso de una **red codificadora** que toma una secuencia de entrada y crea una representación contextualizada de la misma, a menudo denominada contexto.



- Esta representación luego se pasa a un **decodificador** que genera una secuencia de salida específica de la tarea.
- El contexto es una función de las representaciones ocultas de la entrada y puede ser utilizado por el decodificador en una variedad de formas.

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

Encoder

- El encoder del modelo Seq2Seq extrae información del texto de entrada y la codifica en un solo vector, es decir, un vector de “contexto”.
- Básicamente, para cada palabra de entrada, el codificador genera un estado oculto y un vector, utilizando este estado oculto para la siguiente palabra de entrada.
- Se emplean RNN para esta tarea, ya sean GRU, LSTM o Bi-LSTM para el encoder (encoder layer) con el fin de capturar dependencias a largo plazo, mitigando el problema del desvanecimiento/explosión del gradiente que se encuentra al trabajar con RNNs estándar.

Decoder

- El decoder de un modelo seq2seq utiliza el último estado oculto del encoder, es decir, el vector de “contexto”, y genera las palabras de salida.
- El proceso de decodificación comienza una vez que la oración ha sido codificada y el decodificador recibe un estado oculto y un token de entrada en cada paso/tiempo.
- En la marca de tiempo/estado inicial, el primer estado oculto es el vector de contexto y el vector de entrada es <SOS> (inicio de cadena).
- El proceso de decodificación finaliza cuando se alcanza <EOS> (final de oración).
- Los tokens <SOS> y <EOS> se agregan explícitamente al comienzo y al final de cada oración, respectivamente.

Encoder-Decoder

Las redes Encoder-Decoder consisten de tres componentes:

1. Un **encoder** que acepta secuencias de entrada x^n_1, y genera una secuencia correspondiente de representaciones contextualizadas h^n_1. Redes LSTM, CNN, y los Transformers se pueden emplear como Encoders.
2. Un **vector de contexto** c, que es una función de h^n_1 y transmite la esencia de la entrada al decoder.
3. Un **decoder** que acepta a c como entrada y genera una secuencia de longitud arbitraria de estados ocultos h^m_1, a partir de la cual se puede obtener una secuencia correspondiente de estados de salida y^m_1. Al igual que con los encoders, los decoders se pueden realizar mediante cualquier tipo de arquitectura de secuencia.

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

Encoder-Decoder

- En un momento particular t, se pasa el prefijo de $t-1$ tokens a través del modelo de lenguaje, usando inferencia directa para producir una secuencia de estados ocultos, que termina con el estado oculto correspondiente a la última palabra del prefijo.
- Luego se usa el estado oculto final del prefijo como el punto de partida para generar el siguiente token.
- Más formalmente, si g es una función de activación como $tanh$ o $ReLU$, una función de la entrada en el tiempo t y el estado oculto en el tiempo $t-1$, y f es una $softmax$ sobre el conjunto de posibles elementos del vocabulario, entonces en el tiempo t la salida y_t y el estado oculto h_t se calculan como:

$$\mathbf{h}_t = g(\mathbf{h}_{t-1}, \mathbf{x}_t)$$
$$\mathbf{y}_t = f(\mathbf{h}_t)$$

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

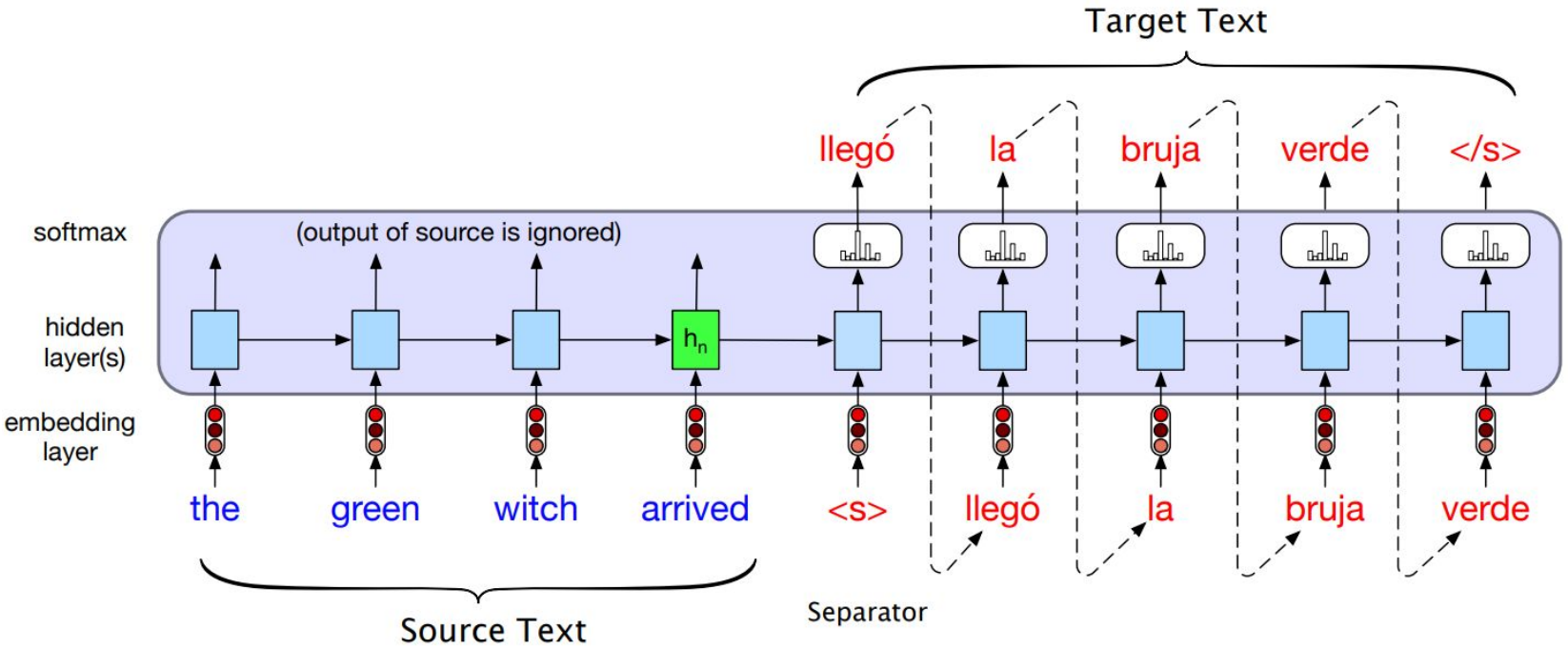
Encoder-Decoder: Ejemplo

Traducir la oración del inglés al español

- *the green witch arrived*
- *llegó la bruja verde*

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

Encoder-Decoder: Arquitectura



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

Encoder-Decoder

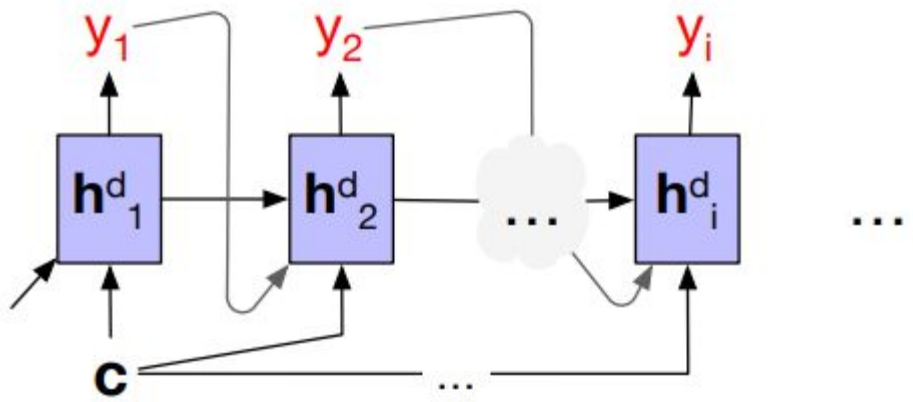
- El propósito del **encoder** es generar una representación contextualizada de la entrada.
- Esta representación se materializa en el estado oculto final del **encoder** h^e_n.
- La representación, también llamada c (contexto), se pasa luego al **decoder**.
- La red del **decoder** de la derecha toma este estado y lo usa para inicializar el primer estado oculto del **decoder**.
- Es decir, la primera celda RNN del **decoder** usa c como su estado oculto previo h^d_0.
- El **decoder** genera auto-regresivamente una secuencia de salidas, un elemento a la vez hasta que se genera un marcador de fin de secuencia.
- Cada estado oculto está condicionado por el estado oculto anterior y la salida generada en el estado anterior.

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

Encoder-Decoder

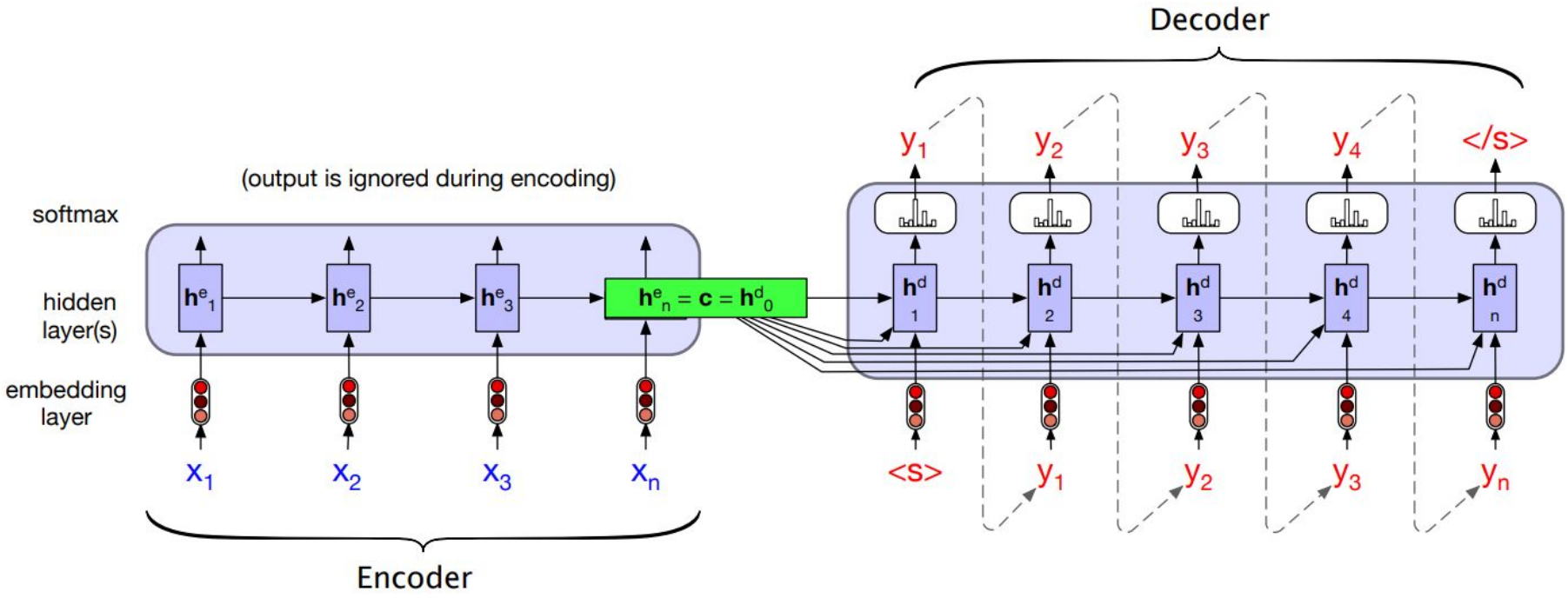
- Una debilidad de este enfoque descrito hasta ahora es que la influencia del vector de contexto c, disminuirá a medida que se genere la secuencia de salida.
- Una solución es hacer que el vector de contexto c esté disponible en cada paso del proceso de decodificación añadiendolo como parámetro al cálculo del estado oculto actual, usando la siguiente ecuación:

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



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

Encoder-Decoder: vector contexto



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

Encoder-Decoder

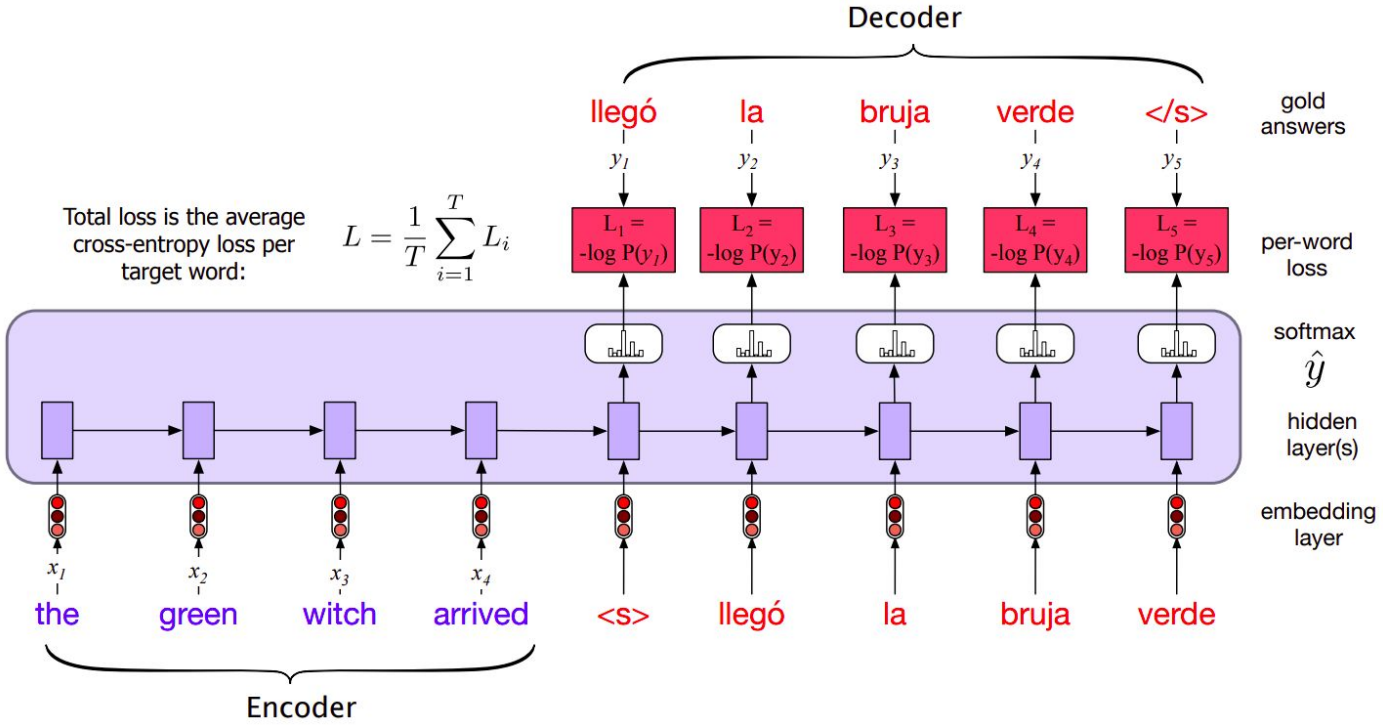
- Las ecuaciones completas para esta versión del decoder en el modelo básico del Encoder-Decoder, con contexto disponible en cada paso de tiempo de decodificación.
- Recordar que g es un sustituto de algún tipo de RNN y $\hat{y}_{t-1}$ es la incrustación de la salida muestreada de la función *softmax* en el paso anterior:
- Finalmente, la salida y en cada paso de tiempo consiste en un cálculo *softmax* sobre el conjunto de posibles salidas (el vocabulario, en el caso del modelado de lenguaje o MT).
- Se calcula la salida más probable en cada paso de tiempo tomando el *argmax* sobre la salida *softmax*:

$$\begin{aligned}
\mathbf{c} &= \mathbf{h}_n^e \\
\mathbf{h}_0^d &= \mathbf{c} \\
\mathbf{h}_t^d &= g(\hat{y}_{t-1}, \mathbf{h}_{t-1}^d, \mathbf{c}) \\
\mathbf{z}_t &= f(\mathbf{h}_t^d) \\
y_t &= \text{softmax}(\mathbf{z}_t)
\end{aligned}$$

$$\hat{y}_t = \text{argmax}_{w \in V} P(w | x, y_1 \ldots y_{t-1})$$

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

Encoder-Decoder: Entrenamiento



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