Atención en RNNs

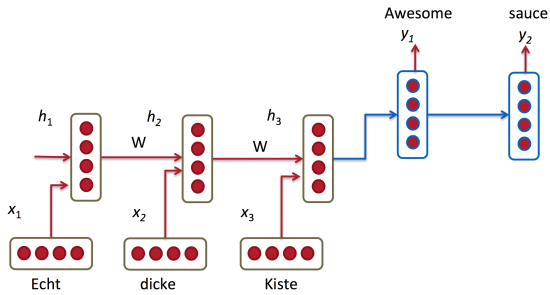
Víctor Mijangos

Facultad de Ciencias

Lingüística computacional

RNNs Encoder-Decoder

Cuando queremos obtener una cadena de longitud T_y dada una cadena de entrada de longitud T_x, tal que $T_x \neq T_y$ utilizamos los modelos de **encoder-decoder**.



Estas arquitecturas son útiles en PLN, pues aparecen en tareas como generación de texto, traducción automática, reinflexión, etc.

Encoder-Decoder

El objetivo de una RNN encoder-decoder es obtener la probabilidad:

$$p(y^{(1)}...y^{(T_y)}|x^{(1)}...x^{(T_x)})$$

Donde $x^{(1)}...x^{(T_x)}$ es una cadena de entrada y $y^{(1)}...y^{(T_y)}$ es una cadena esperada.
Por ejemplo, la cadena de entrada puede ser una oración en una lengua y la de salida su traducción.

Encoder

El encoder se enfoca en codificar una cadena de entrada dentro de un vector c; es decir, es un mapeo:

$$x^{(1)}...x^{(T_x)} \mapsto c \in \mathbb{R}^m$$

El vector c depende de las celdas de la RNN determinada:

$$c = q(h^{(1)}, ..., h^{(T_x)}) \tag{1}$$

Cada celda del Encoder guarda información de los estados en celdas de la forma:

$$h^{(t)} = g(Vh^{(t-1)} + Ux^{(t)} + b)$$

Decoder

El decoder busca obtener la probabilidad de una cadena de salida en los términos siguientes:

$$p(y^{(1)}...y^{(T_y)}) = \prod_{t=1}^{T_y} p(y^{(t)}|y^{(t-1)}...y^{(1)}, c)$$

Donde

$$p(y^{(t)}|y^{(t-1)}...y^{(1)}, c) = g(y^{(t-1)}, s^{(t)}, c)$$

Aquí $s^{(t)}$ representa las celdas recurrentes de decoder.

Decoder

Las celdas del decoder se pueden definir como:

$$s^{(t)} = g\left(\overline{V}s^{(t-1)} + \overline{U}y^{(t-1)} + \overline{b}\right)$$

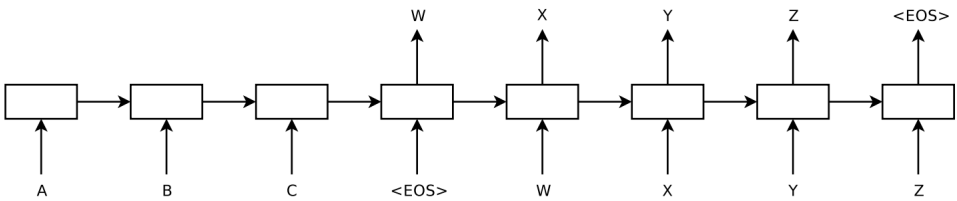
Sin embargo, se debe incorporar el vector de codificación, por lo que tendremos que $s^{(0)} = c$ y $y^{(0)} = x^{(T_x+1)}$

$$s^{(1)} = g\left(\overline{V}c + \overline{U}x^{(T_x+1)} + \overline{b}\right)$$

Decoder

Se puede asumir que c es sólo la última celda del encoder (Sutskever et al., 2014):

$$c = q(h^{(1)}, ..., h^{(T_x)}) = h^{(T_x)}$$



Problema con este enfoque

Las arquitecturas así definidas, estiman una probabilidad de la forma:

$$p(y^{(t)}|y^{(t-1)}, ..., y^{(1)}, x^{(1)}, ..., x^{(T_x)}) = p(y^{(t)}|y^{(t-1)}, ..., y^{(1)}, c)$$

Sin embargo, el decoder envía toda la cadena de entrada para cada salida. Pero, por ejemplo, sabemos que en traducción hay elementos que influyen más en una salida.
Si nos fijamos en la palabra **cat**, veremos que no todas las palabras de entrada influyen igual:

El gato amarillo mira la ventana $\mapsto$ *The yellow* ***cat*** *looks at the window*

Atención

Se busca, entonces, que en cada estado t del decoder, ponga **atención** [1] principalmente a los elementos de entrada que más influyen en la decisión de esa salida.
Se realiza una codificación, para cada estado t de la salida. La probabilidad se expresa como:

$$p(y^{(t)}|y^{(t-1)}, ..., y^{(1)}, x^{(1)}, ..., x^{(T_x)}) = p(y^{(t)}|y^{(t-1)}, ..., y^{(1)}, c^{(t)})$$

Entonces, cada estado del decoder será una función que dependa de los tres elements:

$$s^{(t)} = f(s^{(t-1)}, y^{(t-1)}, c^{(t)})$$

Atención

Se tienen T_y vectores de codificación c_t, determinados por:

$$c^{(t)} = \sum_{s=1}^{T_x} \alpha_{ts} h^{(s)}$$

Donde α_{ts} ponderá las celdas del encoder dependiendo del estado t que se observe en el decoder, donde:

$$\alpha_{ts} = \frac{e^{\epsilon_{ts}}}{\sum_r e^{\epsilon_{tr}}} \tag{2}$$

Y ϵ_{ts} ponderá las celdas del decoder con las del encoder:

$$\epsilon_{ts} = a\left(s^{(t-1)}, h^{(s)}\right)$$

Attention

El valor de la función $a(\cdot)$ puede obtenerse de diferentes formas:

1. Bahdanau (2014):

$$\epsilon_{ts} = a(s^{(t-1)}, h^{(s)}) = v^T \tanh(W[s^{(t-1)}; h^{(s)}])$$

2. Luong (2015):

$$\epsilon_{ts} = a(s^{(t-1)}, h^{(s)}) = s^{(t-1)} W h^{(s)}$$

Otras propuestas se han realizado para obtener este factor.

Resumen de atención

Endoder:

$$h^{(t)} = g(Vh^{(t-1)} + Ux^{(t)} + b)$$

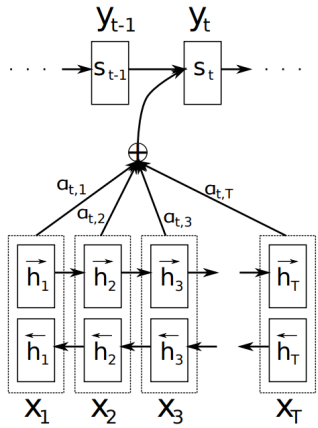
Decoder: Se obtienen los vectores $c^{(t)}$, $t = 1, .., T_y$, como sigue:

1. $\epsilon_{ts} = a(s^{(t-1)}, h^{(s)})$
2. $\alpha_{ts} = \frac{e^{\epsilon_{ts}}}{\sum_r e^{\epsilon_{tr}}}$
3. $c^{(t)} = \sum_{s=1}^{T_x} \alpha_{ts} h^{(s)}$

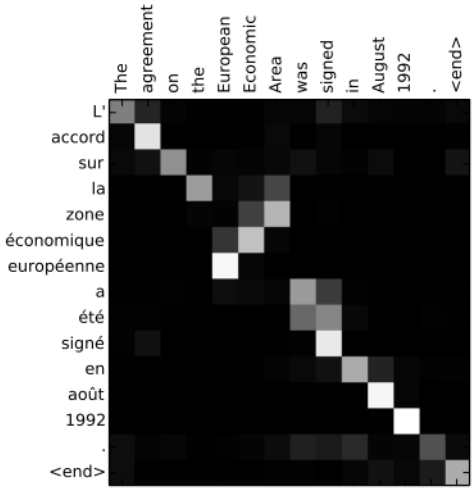
Se obtiene el estado actual:

$$s^{(t)} = g(\overline{V}s^{(t-1)} + \overline{U}y^{(t-1)} + \overline{W}c^{(t)} + \overline{b})$$

Resumen de atención



Matriz de atención



Soft vs Hard attention

Se distinguen dos tipos de atención con respecto a si los pesos de $c^{(t)}$ son distribuidos o discretos:

Soft attention: Los pesos de $c^{(t)}$ representan una probabilidad (como lo hemos definido):

$$c^{(t)} = \sum_{s=1}^{T_x} \alpha_{ts} h^{(s)}$$

Hard attention: Los pesos de $\hat{c}^{(t)}$ son sólo 0s y 1s. Así, se tiene:

$$\hat{c}_i^{(t)} = \begin{cases} 1 & \text{si } i = \arg\max c^{(t)} \\ 0 & \text{en otros casos} \end{cases}$$

Global vs Local attention

Se consideran dos tipos de atención según el contexto (elementos del encoder) que se tome en cuenta:

Global attention: $c^{(t)}$ es una función de todos las celdas del encoder $h^{(1)} \ldots h^{(T_x)}$:

$$c^{(t)} = \sum_{s=1}^{T_x} \alpha_{ts} h^{(s)}$$

Local attention: $c^{(t)}$ es función de un número $2k$ de celdas del encoder. Es decir, se toma una ventana de $2k$ elementos. Se dice que es **monotónico**, si en el estado t (decoder) se considera esta ventana con centro en $h^{(t)}$:

$$c^{(t)} = \sum_{s=0}^{k} \alpha_{ts} h^{(t+s)} + \sum_{s=0}^{k} \alpha_{ts} h^{(t-s)}$$

References


Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio.
Neural machine translation by jointly learning to align and translate.
arXiv preprint arXiv:1409.0473, 2014.

The End