

Aprendizaje profundo

REDES RECURRENTES

Gibran Fuentes-Pineda

Septiembre 2019

Modelando secuencias

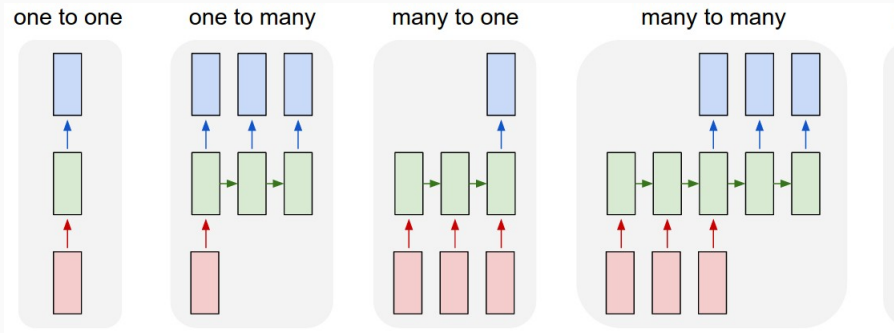


Imagen tomada de Karpathy 2015 (<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>)

- Red (o capas) con retro-alimentación en sus conexiones

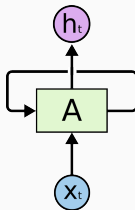


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

Elementos básicos

1. Entradas en cada instante de tiempo t ($\mathbf{x}^{(t)}$)
2. Estados en cada instante de tiempo t ($\mathbf{C}^{(t)}$)
3. Salidas en cada instante de tiempo t ($\mathbf{h}^{(t)}$)

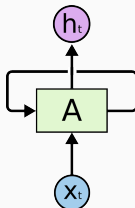


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

Tipos de retroalimentación

1. Salida anterior ($\mathbf{h}^{(t-1)}$)
2. Estado anterior ($\mathbf{C}^{(t-1)}$)
3. Salida y estado anterior ($\mathbf{h}^{(t-1)}$ y $\mathbf{C}^{(t-1)}$)

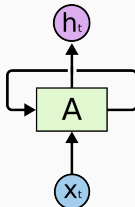


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

- Su estado en cada paso t está dado por

$$\mathbf{C}^{(t)} = \tanh(\mathbf{W}_C \cdot [\mathbf{C}^{(t-1)}, \mathbf{x}^{(t)}] + \mathbf{b}_C)$$

Red neuronal con celdas recurrentes

- Típicamente contiene celdas recurrentes en conjunto con otras capas
- La salida de una celda puede alimentar otras capas u otras celdas
- Un clasificador simple

$$\mathbf{h}^{(t)} = \mathbf{W}_h \mathbf{C}^{(t)} + \mathbf{b}_h$$

$$\mathbf{y}^{(t)} = \text{softmax}(\mathbf{h}^{(t)})$$

Ejemplo: modelo de lenguaje a nivel símbolo

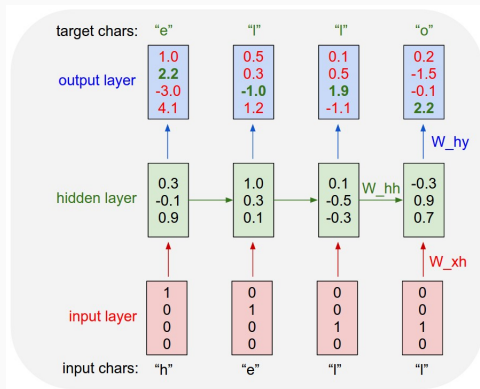


Imagen tomada de Karpathy 2015 (<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>)

Despliegue de celdas

- Una celda recurrente para una secuencia de t valores, se puede desplegar en t capas con parámetros idénticos

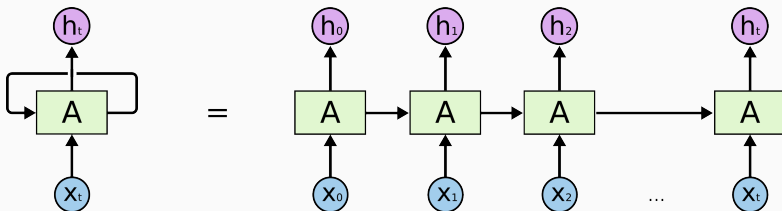


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

Modelando dependencias a corto plazo

- En teoría una red recurrente básica puede modelar dependencias a corto y largo plazo

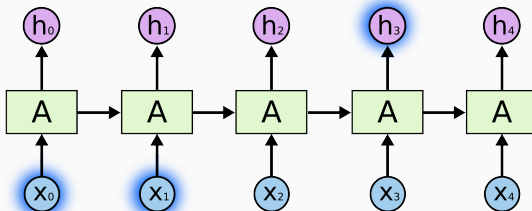


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

El problema de la memoria a largo plazo

- En práctica es muy difícil entrenarlas para tareas con dependencias a largo plazo

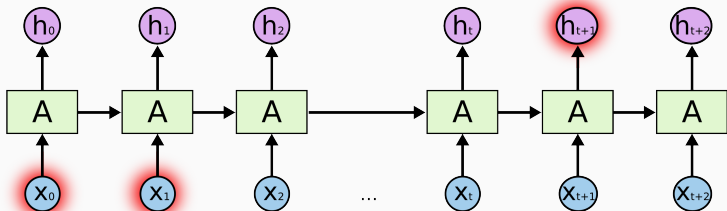


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

Elementos de una celda recurrente estándar

- Una celda básica desplegada recibe como entrada $\mathbf{x}^{(t)}$ y el estado anterior $\mathbf{C}^{(t-1)}$ y produce el nuevo estado $\mathbf{C}^{(t)}$

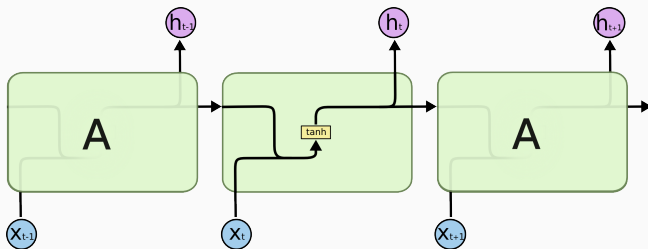


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

Memorias a corto y largo plazo

- Agregan elementos internos a la celda básica que permiten capturar dependencias a corto y largo plazo

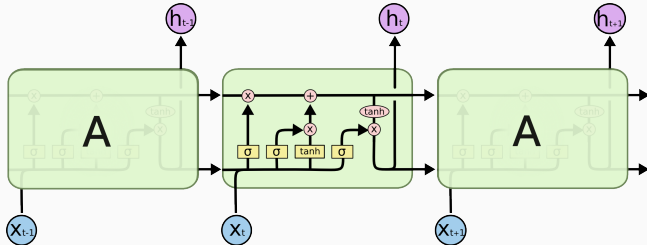


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

LSTM: salida de la capa anterior

- Agrega o elimina elementos del estado anterior de la celda $\mathbf{C}^{(t-1)}$ basado en transformación de la entrada actual $\mathbf{x}^{(t)}$ y el estado oculto anterior $\mathbf{h}^{(t-1)}$

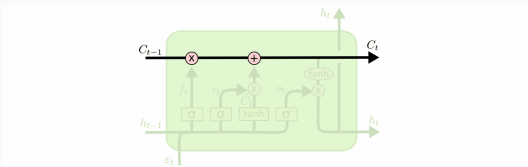


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

LSTM: compuerta de olvido

- Determina qué olvidar del estado de la celda $\mathbf{C}^{(t-1)}$ y en qué proporción a partir de la entrada actual $\mathbf{x}^{(t)}$ y estado oculto anterior $\mathbf{h}^{(t-1)}$

$$\mathbf{f}^{(t)} = \sigma(\mathbf{W}_f \cdot [\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}] + \mathbf{b}_f)$$

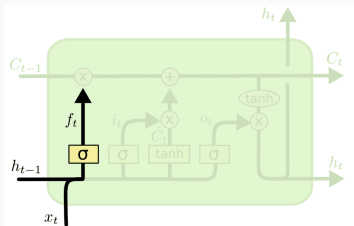


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

LSTM: computerta de entrada

- Determina qué agregar al estado de la celda $\mathbf{C}^{(t-1)}$ y en qué proporción a partir de la entrada actual $\mathbf{x}^{(t)}$ y estado oculto anterior $\mathbf{h}^{(t-1)}$

$$\mathbf{i}^{(t)} = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{(t-1)}, \mathbf{x}^{(t)}] + \mathbf{b}_i)$$

$$\hat{\mathbf{C}}^{(t)} = \tanh(\mathbf{W}_C \cdot [\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}] + \mathbf{b}_C)$$

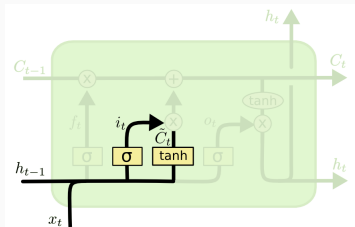


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

LSTM: nuevo estado

- El nuevo estado de la celda se obtiene como una combinación de la salida de la compuerta de olvido $f^{(t)}$ y las salidas $i^{(t)}$ y $\hat{C}^{(t)}$ de la compuerta de entrada

$$C^{(t)} = f^{(t)} * C^{(t-1)} + i^{(t)} * \hat{C}^{(t)}$$

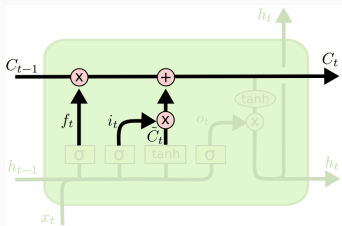


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

LSTM: computerta de salida

- El siguiente estado oculto $\mathbf{h}^{(t)}$ se obtiene como una combinación de la entrada actual $\mathbf{x}^{(t)}$, el estado oculto anterior $\mathbf{h}^{(t-1)}$ y el nuevo estado de la celda $\mathbf{C}^{(t)}$

$$\mathbf{o}^{(t)} = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o)$$

$$\mathbf{h}^{(t)} = \mathbf{o}^{(t)} * \tanh(\mathbf{C}^{(t)})$$

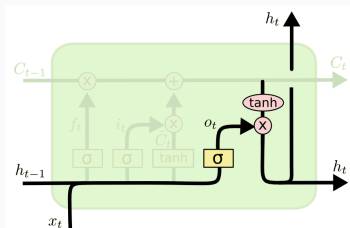


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

Gated Recurrent Unit

- Combina compuertas de olvido y entrada en una sola y agrega otras conexiones y elementos

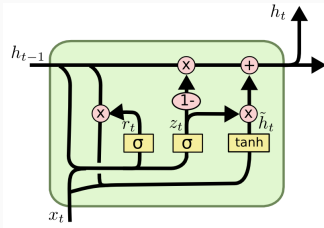


Imagen tomada de Colah 2015 (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>)

[io/posts/2015-08-Understanding-LSTMs/](http://colah.github.io/posts/2015-08-Understanding-LSTMs/))

$$\begin{aligned} \mathbf{z}^{(t)} &= \sigma(\mathbf{W}_z \cdot [\mathbf{h}_{(t-1)}, \mathbf{x}^{(t)}] + \mathbf{b}_z) \\ \mathbf{r}^{(t)} &= \sigma(\mathbf{W}_r \cdot [\mathbf{h}_{(t-1)}, \mathbf{x}^{(t)}] + \mathbf{b}_r) \\ \tilde{\mathbf{h}}^{(t)} &= \tanh(\mathbf{W} \cdot [\mathbf{r}^{(t)} * \mathbf{h}_{(t-1)}, \mathbf{x}^{(t)}] + \mathbf{b}) \\ \mathbf{h}^{(t)} &= (1 - \mathbf{z}^{(t)}) * \mathbf{h}^{(t-1)} + \mathbf{z}^{(t)} * \tilde{\mathbf{h}}^{(t-1)} \end{aligned}$$

Retropropagación en el tiempo

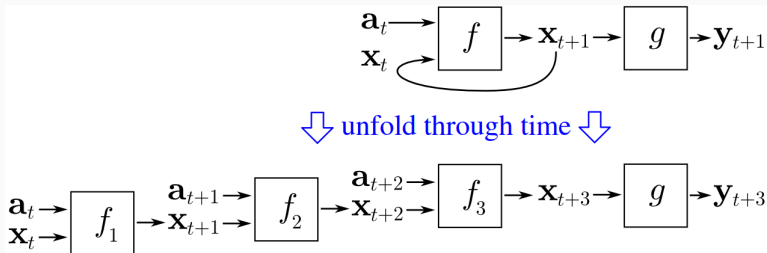


Imagen tomada de Wikipedia (https://en.wikipedia.org/wiki/Backpropagation_through_time)

Redes recurrentes apiladas

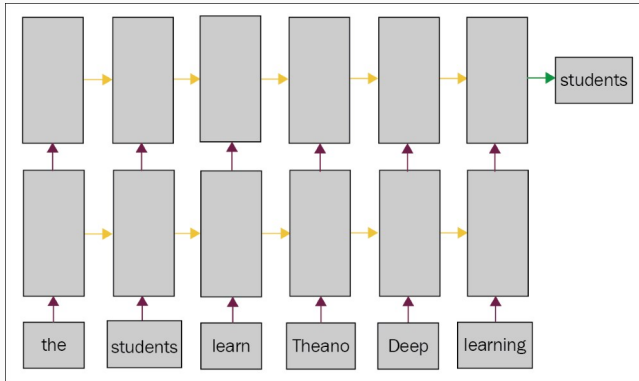


Imagen tomada de: Christopher Bourez. *Deep Learning with Theano*, 2017.

RNR Bidireccional

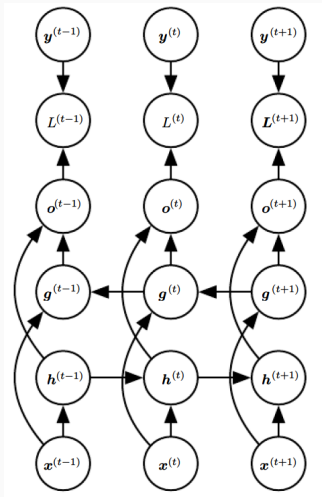


Imagen tomada de Goodfellow et al. 2016 (<http://www.deeplearningbook.org/contents/rnn.html>)

Modelos secuencia a secuencia

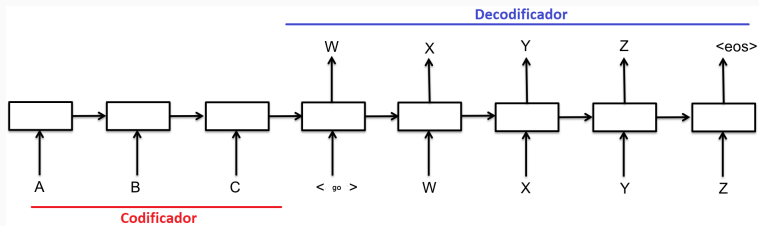


Imagen derivada de <https://www.tensorflow.org/tutorials/seq2seq>