Clase 5: Redes Neuronales Recurrentes

Long Short-Term Memory

Dr. Juan Bekios Calfa

Magíster en Inteligencia Artificial
Facultad de Ingeniería y Ciencias Geológicas

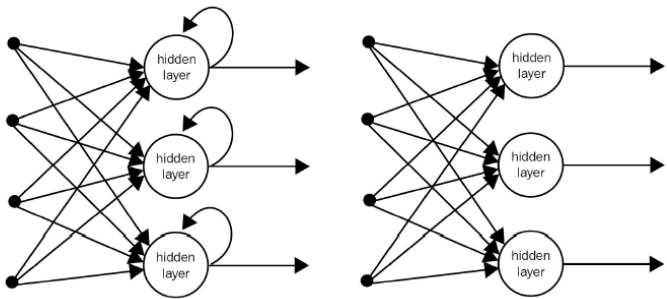
UNIVERSIDAD ADOLFO IBÁÑEZ

Información de Contacto

- Juan Bekios Calfa
  - email: juan.bekios@edu.uai.cl
  - Web page: http://jbekios.ucn.cl
  - Teléfono: 235(5162) - 235(5125)

Definición

- Las **redes neuronales recurrentes**, son redes neuronales que presentan uno o más ciclos en el grafo definido por las interconexiones de sus unidades de procesamiento.



(a) Recurrent neural network

(b) Forward neural network

Definición

- **Redes Neuronales Recurrentes** (RNN en inglés), es un tipo de aprendizaje supervisado.
- La existencia de estos ciclos les permite trabajar de forma innata con **secuencias temporales**.
- Las redes recurrentes son sistemas dinámicos no lineales capaces de descubrir regularidades temporales en las secuencias procesadas y pueden aplicarse, por lo tanto, a multitud de tareas de procesamiento de este tipo de secuencias.

Predicción de series temporales

A partir de la historia pasada de una o más variables, la red neuronal debeproporcionar una predicción lo más correcta posible de su valor futuro.

https://conceptosclaros.com/serie-temporal/

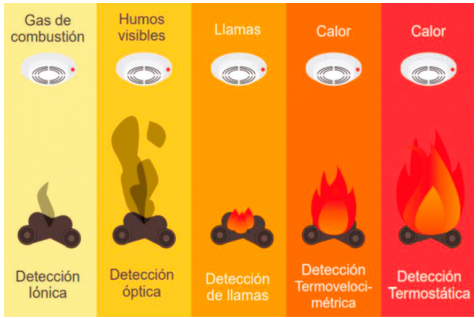
Procesamiento del lenguaje humano

El análisis sintáctico de frases o el estudio de regularidades en el lenguaje son algunas tareas relacionadas con el lenguaje (escrito) aplicadas a las recurrentes.

https://decidesoluciones.es/procesamiento-del-lenguaje-natural-pln-o-nlp-que-es-y-para-que-se-utiliza/

Detección de anomalías

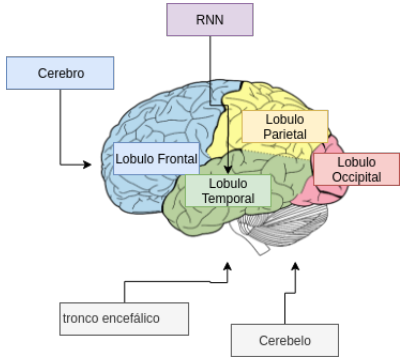
Identificación de eventos anómalos acontencidos sobre equipos. Para la identificación se utilizan datos históricos de operación, con la intención de predecirlos en el futuro. Generación de alertas tempranas.



https://revistainnovacion.com/nota/10467/nociones_basicas_de_un_sistema_de_deteccion_de_incendios/

Conceptos generales

- Nuestro cerebro no solo procesa patrones de manera independiente. Utiliza el lóbulo temporal que se encarga de nuestra memoria a largo plazo.



Conceptos generales

- **Los seres humanos**, normalmente, no resolvemos problemas desde cero. Por ejemplo, a medida que leemos este párrafo comprendemos el contexto de cada palabra según la comprensión de las palabras anteriores.

Conceptos generales

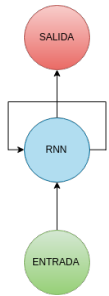
- **Los seres humanos**, normalmente, no resolvemos problemas desde cero. Por ejemplo, a medida que leemos este párrafo comprendemos el contexto de cada palabra según la comprensión de las palabras anteriores.
- Las **arquitecturas neuronales tradicionales** tienen la desventaja de no recordar (memoria) los estados anteriores o predicciones anteriores realizadas. En general, reciben una entrada (atributos) y calculan su predicción por medio de un modelo optimizado. Cada predicción es independiente de la anterior.

Conceptos generales

- **Los seres humanos**, normalmente, no resolvemos problemas desde cero. Por ejemplo, a medida que leemos este párrafo comprendemos el contexto de cada palabra según la comprensión de las palabras anteriores.
- Las **arquitecturas neuronales tradicionales** tienen la desventaja de no recordar (memoria) los estados anteriores o predicciones anteriores realizadas. En general, reciben una entrada (atributos) y calculan su predicción por medio de un modelo optimizado. Cada predicción es independiente de la anterior.
- Por ejemplo: Si vemos una película, es muy difícil entender contexto general de ella solo analizando los elementos individuales de cada imagen.

Conceptos generales

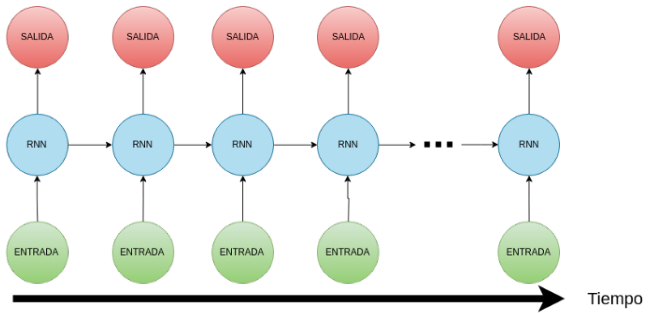
- Las **redes neuronales recurrentes**, son modelos, que se alimentan a si mismas.
- Cada salida de activación retro-alimenta nuevamente a la neurona en su entrada.



https://medium.com/@jcrispis56/introducci%C3%B3n-al-deep-learning-parte-3-redes-neuronales-recurrentes-7da543c3b181

Concepto inspirado en nuestro cerebro

- Un modelo más moderno de redes neuronales recurrentes está compuesta de una secuencia de redes neuronales las cuales se retro-alimentan unas con otras.

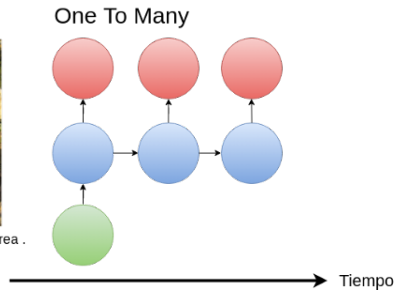


https://medium.com/@jcrispis56/introducci%C3%B3n-al-deep-learning-parte-3-redes-neuronales-recurrentes-7da543c3b181

Arquitecturas de Redes Neuronales Recurrentes

- Arquitectura **ONE TO MANY**: Una entrada, múltiples salidas. Por ejemplo: Una imagen de entrada y como salida una descripción de la imagen.

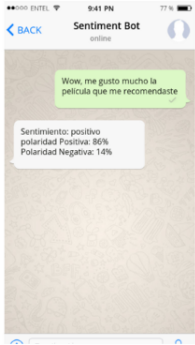
a group of giraffes standing in a grassy area .
Fuente: https://www.analyticsvidhya.com

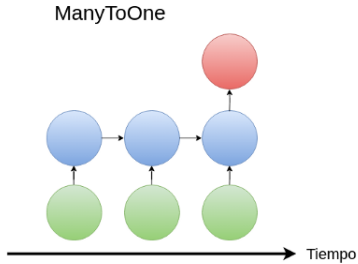


https://medium.com/@jcrispis56/introducci%C3%B3n-al-deep-learning-parte-3-redes-neuronales-recurrentes-7da543c3b181

Arquitecturas de Redes Neuronales Recurrentes

- Arquitectura **MANY TO ONE**: Un conjunto de entradas, una salida. Por ejemplo: A partir de un texto de entrada predecir, como salida, es estado de ánimo del texto.

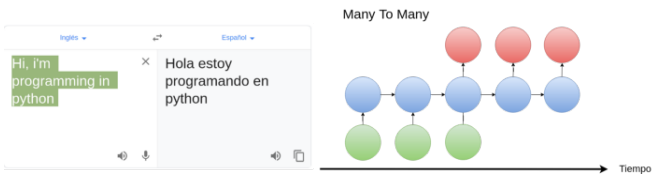




https:

Arquitecturas de Redes Neuronales Recurrentes

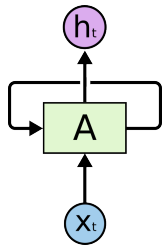
- Arquitectura **MANY TO MANY**: Un conjunto de entradas, un conjunto de salidas. Por ejemplo: La traducción automática es un ejemplo de este tipo de arquitectura. Como entrada tenemos un texto en un idioma (español) y como salida obtenemos otro texto traducido (inglés).



https://medium.com/@jcrispis56/introducci%C3%B3n-al-deep-learning-parte-3-redes-neuronales-recurrentes-7da543c3b181

Una simple red neuronal recurrente

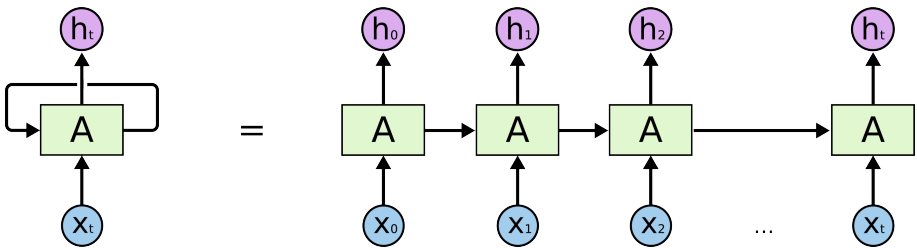
- Una red neuronal recurrente A, procesa alguna entrada x_t y obtiene como salida el valor h_t. La red A permite que la información se retro-alimente en el siguiente paso a través de bucle.



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Encadenamiento de redes neuronales recurrentes

- Una red neuronal recurrente moderna crea copias múltiples de la misma red, cada una de las cuales pasa un mensaje a un sucesor.



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Encadenamiento de redes neuronales recurrentes

- Al encadenar redes neuronales recurrentes revelamos su naturaleza para resolver problemas relacionados con secuencias y listas de datos.

Encadenamiento de redes neuronales recurrentes

- Al encadenar redes neuronales recurrentes revelamos su naturaleza para resolver problemas relacionados con secuencias y listas de datos.
- Son la arquitectura natural de la red neuronal para usar para dichos datos.

Encadenamiento de redes neuronales recurrentes

- Al encadenar redes neuronales recurrentes revelamos su naturaleza para resolver problemas relacionados con secuencias y listas de datos.
- Son la arquitectura natural de la red neuronal para usar para dichos datos.
- Esta implementación permite generar una especie de memoria a corto plazo.

Encadenamiento de redes neuronales recurrentes

- Al encadenar redes neuronales recurrentes revelamos su naturaleza para resolver problemas relacionados con secuencias y listas de datos.
- Son la arquitectura natural de la red neuronal para usar para dichos datos.
- Esta implementación permite generar una especie de memoria a corto plazo.
- Se puede convertir problemas de procesamiento secuencial en problemas que no son secuenciales.

Encadenamiento de redes neuronales recurrentes

- Al encadenar redes neuronales recurrentes revelamos su naturaleza para resolver problemas relacionados con secuencias y listas de datos.
- Son la arquitectura natural de la red neuronal para usar para dichos datos.
- Esta implementación permite generar una especie de memoria a corto plazo.
- Se puede convertir problemas de procesamiento secuencial en problemas que no son secuenciales.
- El éxito de estas redes se debe a la implementación de un tipo de redes neuronales recurrentes llamadas LSTM (*Long Short-Term Memory*).

Dependencias largo plazo

- Uno de los atractivos de los RNN es la idea de que podrían conectar información previa a la tarea actual, como el uso de fotogramas de video anteriores para informar la comprensión del presente cuadro.

Dependencias largo plazo

- Uno de los atractivos de los RNN es la idea de que podrían conectar información previa a la tarea actual, como el uso de fotogramas de video anteriores para informar la comprensión del presente cuadro.
- Dependiendo de la naturaleza del problema se necesita recordar información a **corto plazo** pero en otras ocasiones se necesita recordar información a **largo plazo**.

Dependencias largo plazo

- Uno de los atractivos de los RNN es la idea de que podrían conectar información previa a la tarea actual, como el uso de fotogramas de video anteriores para informar la comprensión del presente cuadro.
- Dependiendo de la naturaleza del problema se necesita recordar información a **corto plazo** pero en otras ocasiones se necesita recordar información a **largo plazo**.
- Consideramos la oración "**las nubes están en el ..**", sabemos que lo más probable que la próxima palabra sea "cielo". No es necesario más contexto que las esas palabras.

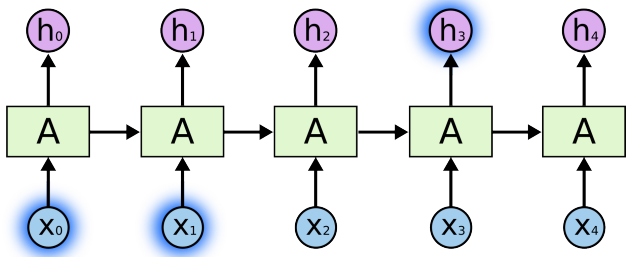
Dependencias largo plazo

- Uno de los atractivos de los RNN es la idea de que podrían conectar información previa a la tarea actual, como el uso de fotogramas de video anteriores para informar la comprensión del presente cuadro.
- Dependiendo de la naturaleza del problema se necesita recordar información a **corto plazo** pero en otras ocasiones se necesita recordar información a **largo plazo**.
- Consideramos la oración "**las nubes están en el ..**", sabemos que lo más probable que la próxima palabra sea "cielo". No es necesario más contexto que las esas palabras.

Dependencias largo plazo

- Uno de los atractivos de los RNN es la idea de que podrían conectar información previa a la tarea actual, como el uso de fotogramas de video anteriores para informar la comprensión del presente cuadro.
- Dependiendo de la naturaleza del problema se necesita recordar información a **corto plazo** pero en otras ocasiones se necesita recordar información a **largo plazo**.
- Consideramos la oración "**las nubes están en el ..**", sabemos que lo más probable que la próxima palabra sea "cielo". No es necesario más contexto que las esas palabras.
- En esos casos, donde la brecha entre la **información relevante** y **el lugar que se necesita es pequeña**, los RNN pueden aprender a usar la información pasada.

RNN con memoria de corto plazo



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Dependencias largo plazo (II)

- Pero también hay casos en los que necesitamos más contexto.

Dependencias largo plazo (II)

- Pero también hay casos en los que necesitamos más contexto.
- Supongamos que deseo predecir la próxima palabra de la oración: “Nací en Chile. Me gustan las empanadas y bailo cueca todos los días. Entonces, yo habló ...”. (“chileno”, lenguaje parecido al español ☺)

Dependencias largo plazo (II)

- Pero también hay casos en los que necesitamos más contexto.
- Supongamos que deseo predecir la próxima palabra de la oración: “Nací en Chile. Me gustan las empanadas y bailo cueca todos los días. Entonces, yo habló ...”. (“chileno”, lenguaje parecido al español ☺)
- La información sugiere que la siguiente palabra es probablemente el nombre de algún idioma, pero si queremos limitar el idioma, necesitamos el contexto del país (Chile), cuya información está al comienzo de la oración.

Dependencias largo plazo (II)

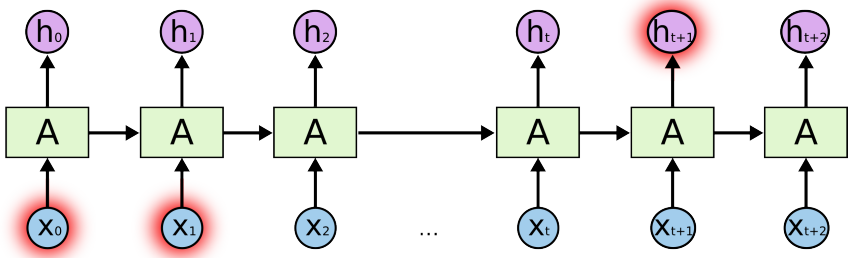
- Pero también hay casos en los que necesitamos más contexto.
- Supongamos que deseo predecir la próxima palabra de la oración: “Nací en Chile. Me gustan las empanadas y bailo cueca todos los días. Entonces, yo habló ...”. (“chileno”, lenguaje parecido al español ☺)
- La información sugiere que la siguiente palabra es probablemente el nombre de algún idioma, pero si queremos limitar el idioma, necesitamos el contexto del país (Chile), cuya información está al comienzo de la oración.
- Desafortunadamente, a medida que crece la brecha de recuerdo (memoria). Las RNN no podrían aprender a conectar esa la información.

Doris y su memoria de largo plazo

https://memegenerator.net/instance/81831683/forgetful-dory-when-you-try-to-retrieve-something-from-your-stored-memory-but-you-cant-remember-what

El problema de la dependencia a largo plazo

- En teoría, los RNN son absolutamente capaces de manejar tales “dependencias a largo plazo”.
- Sin embargo, en la práctica esta aproximación no funciona.



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Redes *Long Short Term Memory* (LSTM)

- Es un tipo de **red neuronal recurrente** que es capaz de aprender **términos a largo plazo**.

Redes *Long Short Term Memory* (LSTM)

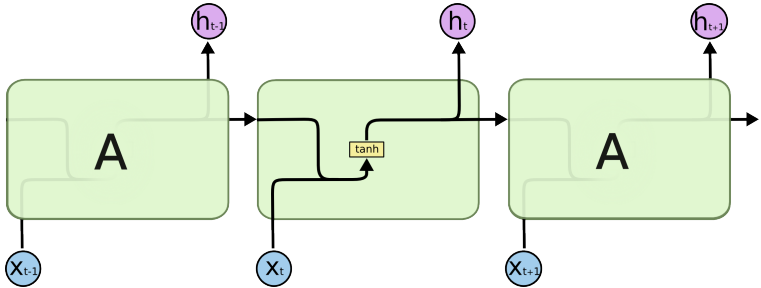
- Es un tipo de **red neuronal recurrente** que es capaz de aprender **términos a largo plazo**.
- Las redes LSTM fueron definidas por los trabajos de *Hochreiter y Schmidhuber* (1997).La nueva propuesta trabajo bien en una gran cantidad de problemas.

Redes *Long Short Term Memory* (LSTM)

- Es un tipo de **red neuronal recurrente** que es capaz de aprender **términos a largo plazo**.
- Las redes LSTM fueron definidas por los trabajos de *Hochreiter y Schmidhuber* (1997).La nueva propuesta trabajo bien en una gran cantidad de problemas.
- Los LSTM están diseñados explícitamente para evitar el problema de dependencia a largo plazo.

Módulo de encadenamiento de RNN estándar.

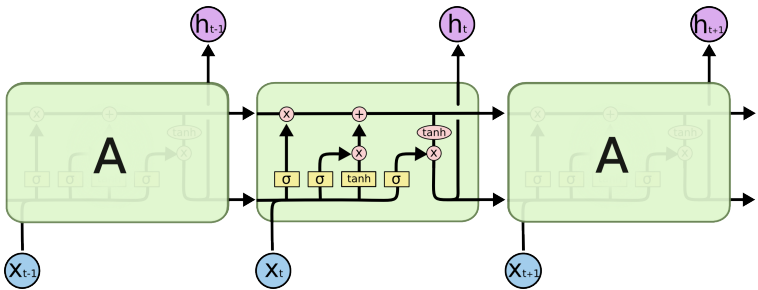
- Todas las **redes neuronales recurrentes** tienen la forma de una cadena de módulos repetitivos de red neuronal.
- Este módulo repetitivo tendrá una **estructura muy simple**, con una sola capa de $tanh$.



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Módulos de encadenamiento en una LSTM.

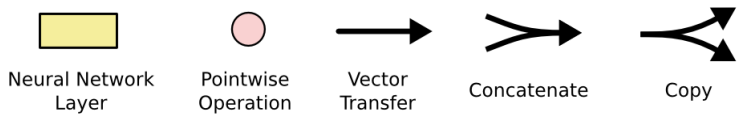
- Los LSTM también tienen esta estructura tipo cadena.
- La diferencia está en el módulo que se repite, este tiene diferente estructura.
- En lugar de tener una sola capa de red neuronal, hay cuatro.



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Notación de una LSTM.

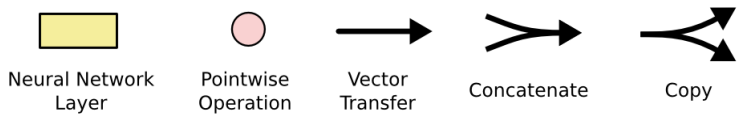
- En el diagrama anterior, cada línea transporta un vector completo, desde la salida de un nodo hasta las entradas de otros.



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Notación de una LSTM.

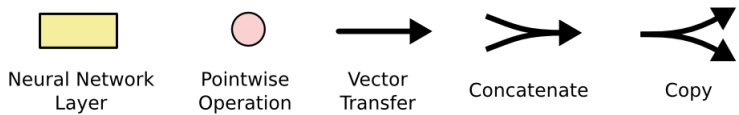
- En el diagrama anterior, cada línea transporta un vector completo, desde la salida de un nodo hasta las entradas de otros.
- **Los círculos rosados**, representan operaciones puntuales, como la suma de vectores.



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Notación de una LSTM.

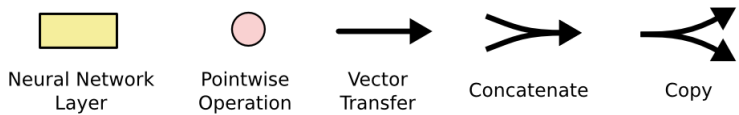
- En el diagrama anterior, cada línea transporta un vector completo, desde la salida de un nodo hasta las entradas de otros.
- **Los círculos rosados**, representan operaciones puntuales, como la suma de vectores.
- **Los cuadros amarillos** se aprenden capas de redes neuronales.



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Notación de una LSTM.

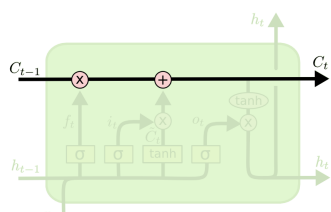
- En el diagrama anterior, cada línea transporta un vector completo, desde la salida de un nodo hasta las entradas de otros.
- **Los círculos rosados**, representan operaciones puntuales, como la suma de vectores.
- **Los cuadros amarillos** se aprenden capas de redes neuronales.
- Las **líneas de fusión** denotan concatenación. Mientras, que una **línea de bifurcación** denota que su contenido se copia y las copias van a diferentes ubicaciones.



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Estado celda en una LSTM

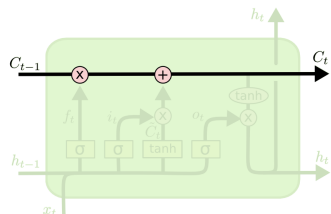
- La clave para los LSTM es el **estado de la celda** (*cell state*), la línea horizontal que pasa por la parte superior del diagrama.



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Estado celda en una LSTM

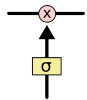
- La clave para los LSTM es el **estado de la celda** (*cell state*), la línea horizontal que pasa por la parte superior del diagrama.
- El **estado de la celda** es como una cinta transportadora. Corre directamente por toda la cadena, con solo algunas interacciones lineales menores. Es muy fácil que la información fluya sin cambios.



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Estado celda en una LSTM

- El LSTM tiene la capacidad de eliminar o agregar información al estado de la celda, cuidadosamente regulado por estructuras llamadas **puertas** (*gates*).



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

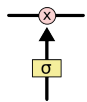
Estado celda en una LSTM

- El LSTM tiene la capacidad de eliminar o agregar información al estado de la celda, cuidadosamente regulado por estructuras llamadas **puertas** (*gates*).
- Las **puertas** son una forma opcional de dejar pasar la información. Se componen de una capa de red neuronal sigmoidea y una operación de multiplicación puntual.

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Estado celda en una LSTM

- El LSTM tiene la capacidad de eliminar o agregar información al estado de la celda, cuidadosamente regulado por estructuras llamadas **puertas** (*gates*).
- Las **puertas** son una forma opcional de dejar pasar la información. Se componen de una capa de red neuronal sigmoidea y una operación de multiplicación puntual.
- **La capa sigmoide** genera números entre cero y uno, que describe la cantidad de cada componente que debe dejarse pasar.



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Estado celda en una LSTM

- Un valor de cero significa “no dejar pasar nada”, mientras que un valor de uno significa “dejar pasar todo”.

Estado celda en una LSTM

- Un valor de cero significa “no dejar pasar nada”, mientras que un valor de uno significa “dejar pasar todo”.
- Un LSTM tiene tres de estas puertas, para proteger y controlar el estado de la célula.

Estado celda en una LSTM

- Un valor de cero significa “no dejar pasar nada”, mientras que un valor de uno significa “dejar pasar todo”.
- Un LSTM tiene tres de estas puertas, para proteger y controlar el estado de la célula.
    1. Capa de filtrado de la celda de estado (*Forget Gate Layer*).

Estado celda en una LSTM

- Un valor de cero significa “no dejar pasar nada”, mientras que un valor de uno significa “dejar pasar todo”.
- Un LSTM tiene tres de estas puertas, para proteger y controlar el estado de la célula.
  1. Capa de filtrado de la celda de estado (*Forget Gate Layer*).
  2. Capa para almacenar información (*Input Gate Layer*).

Estado celda en una LSTM

- Un valor de cero significa “no dejar pasar nada”, mientras que un valor de uno significa “dejar pasar todo”.
- Un LSTM tiene tres de estas puertas, para proteger y controlar el estado de la célula.
  1. Capa de filtrado de la celda de estado (*Forget Gate Layer*).
  2. Capa para almacenar información (*Input Gate Layer*).
  3. Capa de salida (*Output Layer*).

Paso1: Filtrar, o no, información al estado de la celda

- El primer paso en nuestro LSTM es decidir qué información vamos a traspasar al estado de la celda.

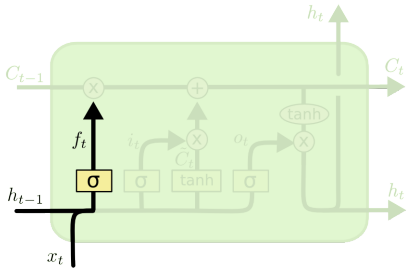
- Examina h_{t-1} y x_t, y genera un número entre 0 y 1 para cada número en el estado de celda C_{t-1}.
- Un **1** representa "**mantener completamente**", mientras que un **0** representa "**deshacerse completamente**".

Paso1: Filtrar, o no, información al estado de la celda

- El primer paso en nuestro LSTM es decidir qué información vamos a traspasar al estado de la celda.
- Esta decisión la toma una capa sigmoidea llamada “capa de puerta olvidada” (*forget gate layer*).
- Examina h_{t-1} y x_t, y genera un número entre 0 y 1 para cada número en el estado de celda C_{t-1}.
- Un **1** representa “**mantener completamente**”, mientras que un **0** representa “**deshacerse completamente**”.

Paso 1: Filtrar, o no, información al estado de la celda

- Si recordamos el ejemplo de un **modelo de lenguaje** que intenta predecir la siguiente palabra en función de todas las anteriores.

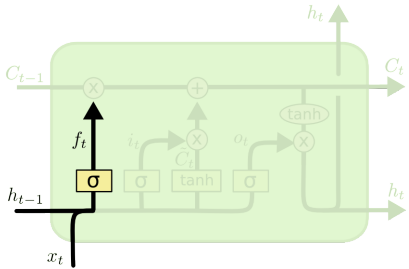


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Paso 1: Filtrar, o no, información al estado de la celda

- Si recordamos el ejemplo de un **modelo de lenguaje** que intenta predecir la siguiente palabra en función de todas las anteriores.
- En tal problema, el **estado de la celda** puede incluir el **género del sujeto** presente, de modo que se puedan usar los **pronombres** correctos. Cuando vemos un tema nuevo, queremos **olvidar** el género del tema anterior.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Paso 2: Almacenar información

- En este paso decidimos qué **nueva información** vamos a **almacenar en el estado de la celda**.

Paso 2: Almacenar información

- En este paso decidimos qué **nueva información** vamos a **almacenar en el estado de la celda**.
- Primero, una capa sigmoidea llamada “capa de puerta de entrada” decide qué valores actualizaremos.

Paso 2: Almacenar información

- En este paso decidimos qué **nueva información** vamos a **almacenar en el estado de la celda**.
- Primero, una capa sigmoidea llamada “capa de puerta de entrada” decide qué valores actualizaremos.
- Luego, una capa $tanh$ crea un vector de nuevos valores candidatos, $\widetilde{C}_t$, que podrían agregarse al estado.

Paso 2: Almacenar información

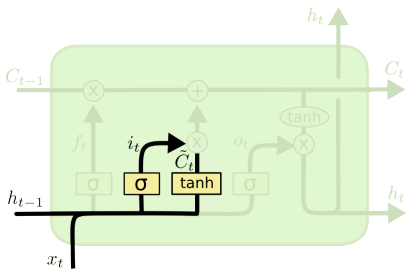
- En este paso decidimos qué **nueva información** vamos a **almacenar en el estado de la celda**.
- Primero, una capa sigmoidea llamada “capa de puerta de entrada” decide qué valores actualizaremos.
- Luego, una capa $tanh$ crea un vector de nuevos valores candidatos, $\widetilde{C}_t$, que podrían agregarse al estado.
- Finalmente, combinaremos estos dos para crear una actualización del estado.

Paso 2: Almacenar información

- En este paso decidimos qué **nueva información** vamos a **almacenar en el estado de la celda**.
- Primero, una capa sigmoidea llamada “capa de puerta de entrada” decide qué valores actualizaremos.
- Luego, una capa $tanh$ crea un vector de nuevos valores candidatos, $\widetilde{C}_t$, que podrían agregarse al estado.
- Finalmente, combinaremos estos dos para crear una actualización del estado.
- En el ejemplo del **modelo de lenguaje**, queremos agregar el género del nuevo sujeto al estado de la celda, para reemplazar el antiguo que estamos olvidando.

Paso 2: Almacenar información

- Ahora es el momento de actualizar el estado de la celda anterior, C_{t-1}, en el nuevo estado de la celda C_t. Los pasos anteriores ya decidieron qué hacer, ahora solo hay que hacerlo.

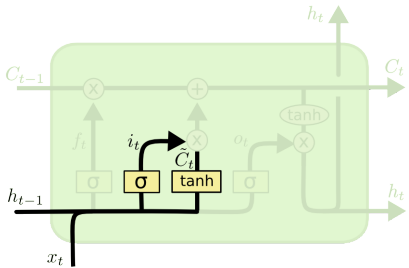


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Paso 2: Almacenar información

- Ahora es el momento de actualizar el estado de la celda anterior, C_{t-1}, en el nuevo estado de la celda C_t. Los pasos anteriores ya decidieron qué hacer, ahora solo hay que hacerlo.
- Multiplicamos el antiguo estado por f_t, olvidando las cosas que decidimos olvidar antes. Luego lo agregamos $i_t \times \widetilde{C}_t$. Estos son los nuevos valores candidatos, escalados según cuánto decidimos actualizar cada valor de estado.

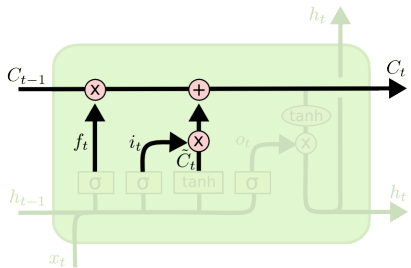


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Paso 2: Almacenar información

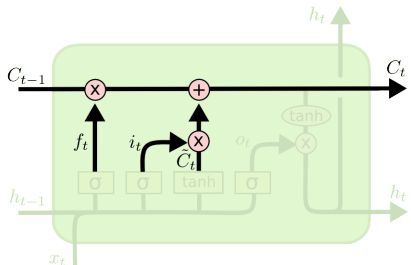
- Ahora es el momento de actualizar el estado de la celda anterior, C_{t-1}, en el nuevo estado de la celda C_t. Los pasos anteriores ya decidieron qué hacer, ahora solo hay que hacerlo.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Paso 2: Almacenar información

- Ahora es el momento de actualizar el estado de la celda anterior, C_{t-1}, en el nuevo estado de la celda C_t. Los pasos anteriores ya decidieron qué hacer, ahora solo hay que hacerlo.
- Multiplicamos el antiguo estado por f_t, olvidando las cosas que decidimos olvidar antes. Luego lo agregamos $i_t \times \widetilde{C}_t$. Estos son los nuevos valores candidatos, escalados según cuánto decidimos actualizar cada valor de estado.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

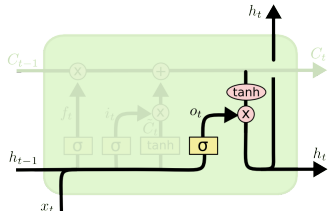
Paso 2: Almacenar información

Aquí es donde desecharíamos la información sobre el género del sujeto antiguo y agregaríamos la nueva información, como decidimos en los pasos anteriores.

(Modelo de predicción de palabras)

Paso 3: Calcular salida

- Finalmente, necesitamos decidir qué vamos a generar.

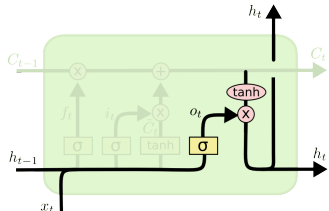


$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Paso 3: Calcular salida

- Finalmente, necesitamos decidir qué vamos a generar.
- Esta salida se basará en nuestro estado de celda, pero será una versión filtrada.

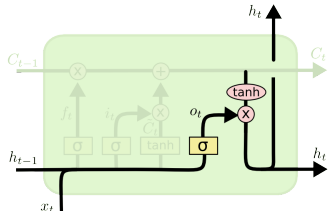


$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Paso 3: Calcular salida

- Finalmente, necesitamos decidir qué vamos a generar.
- Esta salida se basará en nuestro estado de celda, pero será una versión filtrada.
- Primero, ejecutamos una capa sigmoidea que decide qué partes del estado de la celda vamos a generar.

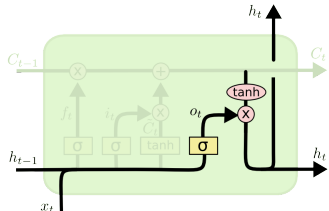


$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

Paso 3: Calcular salida

- Finalmente, necesitamos decidir qué vamos a generar.
- Esta salida se basará en nuestro estado de celda, pero será una versión filtrada.
- Primero, ejecutamos una capa sigmoidea que decide qué partes del estado de la celda vamos a generar.
- Luego, colocamos el estado de la celda a través de $tanh$ (para empujar los valores a estar entre -1 y 1) y lo multiplicamos por la salida de la puerta sigmoidea, de modo que solo produzcamos las partes que decidimos.



$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

UNIVERSIDAD ADOLFO IBÁÑEZ

Paso 3: Calcular salida

Dado que acaba de ver un tema, es posible que se desee generar información relevante para un verbo, en caso de que eso sea lo que viene a continuación. Por ejemplo, podría generar si el sujeto es singular o plural, de modo que sepamos en qué forma se debe conjugar un verbo si eso es lo que sigue a continuación.

(Modelo de predicción de palabras)

UNIVERSIDAD ADOLFO IBÁÑEZ

Referencias

- Tesis doctoral: Juan Pérez Ortiz. Modelos predictivos basados en redes neuronales recurrentes de tiempo discreto. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante. (07-2002).
- *Understanding LSTM Networks*. `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`

UNIVERSIDAD ADOLFO IBÁÑEZ

¿Preguntas?