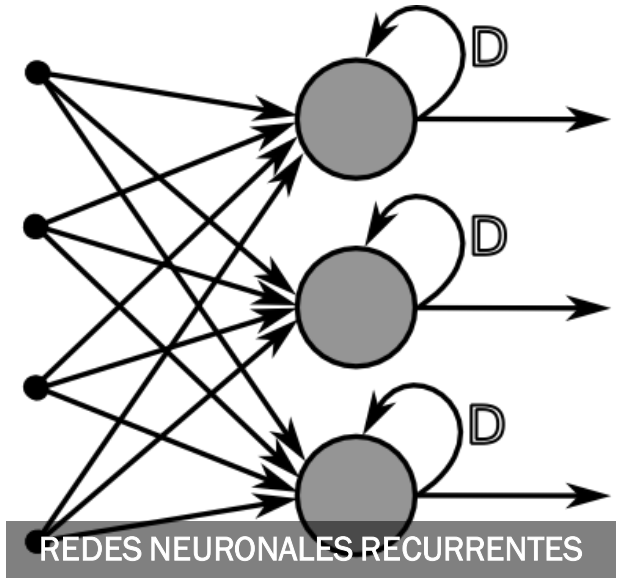
REDES NEURONALES RECURRENTES

INTELIGENCIA ARTIFICIAL – EJEMPLOS AVANZADOS

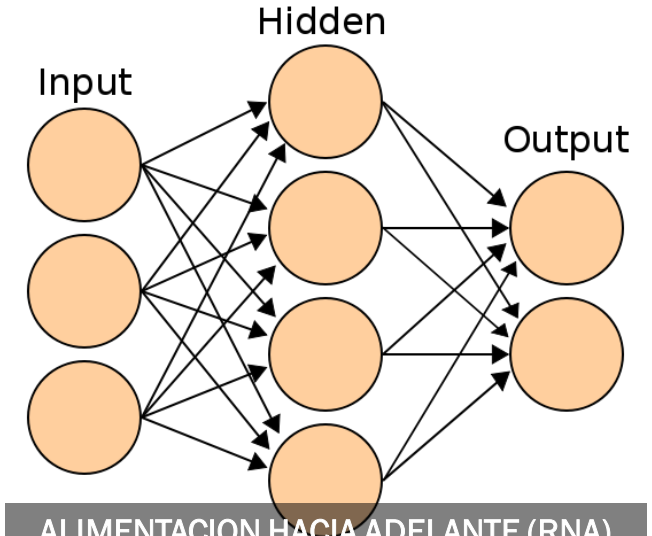
HTTPS://DATADOSIS.COM

REDES NEURONALES RECURRENTES (RNR): ¿QUÉ SON?

- Cubrimos las redes neuronales Feedforward (redes de vainilla) que mapean una entrada de tamaño fijo (como una imagen) a una salida de tamaño fijo (clases o probabilidades).
- Una desventaja de las redes de retroalimentación es que no tienen ningún efecto de dependencia temporal o de memoria.
- Una RNR es un tipo de RNA que está diseñada para tener en cuenta la dimensión temporal al tener una memoria (estado interno) (bucle de retroalimentación).



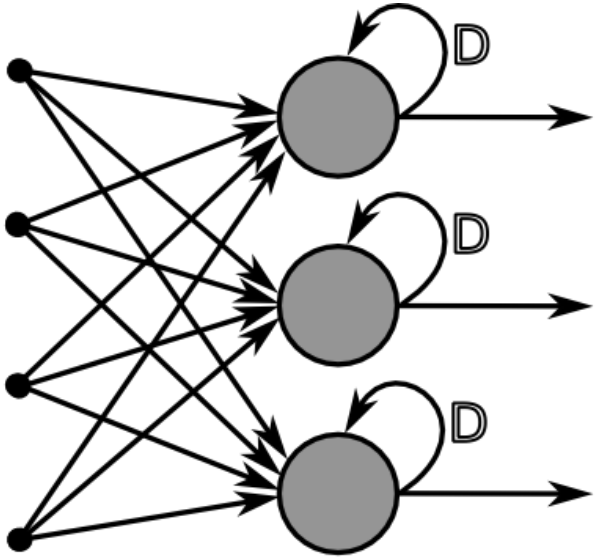
REDES NEURONALES RECURRENTES



ALIMENTACION HACIA ADELANTE (RNA)

Photo Credit: https://commons.wikimedia.org/wiki/File:RecurrentLayerNeuralNetwork_english.png
Photo Credit: https://commons.wikimedia.org/wiki/File:Artificial_neural_network.svg

ARQUITECTURA DE REDES NEURONALES



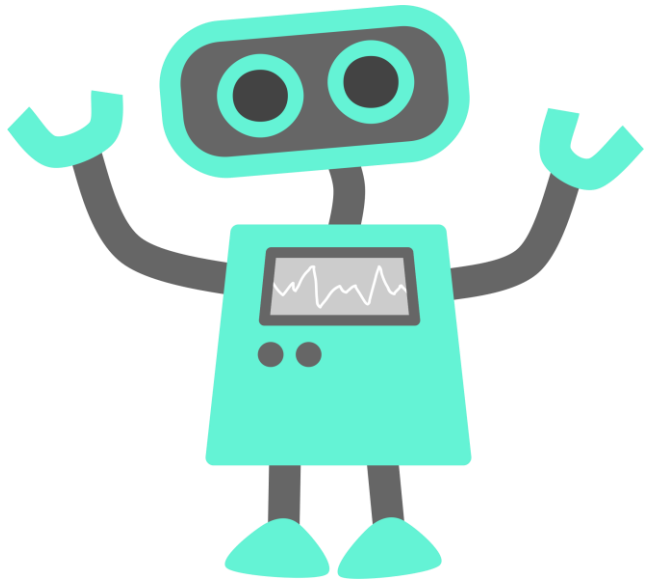
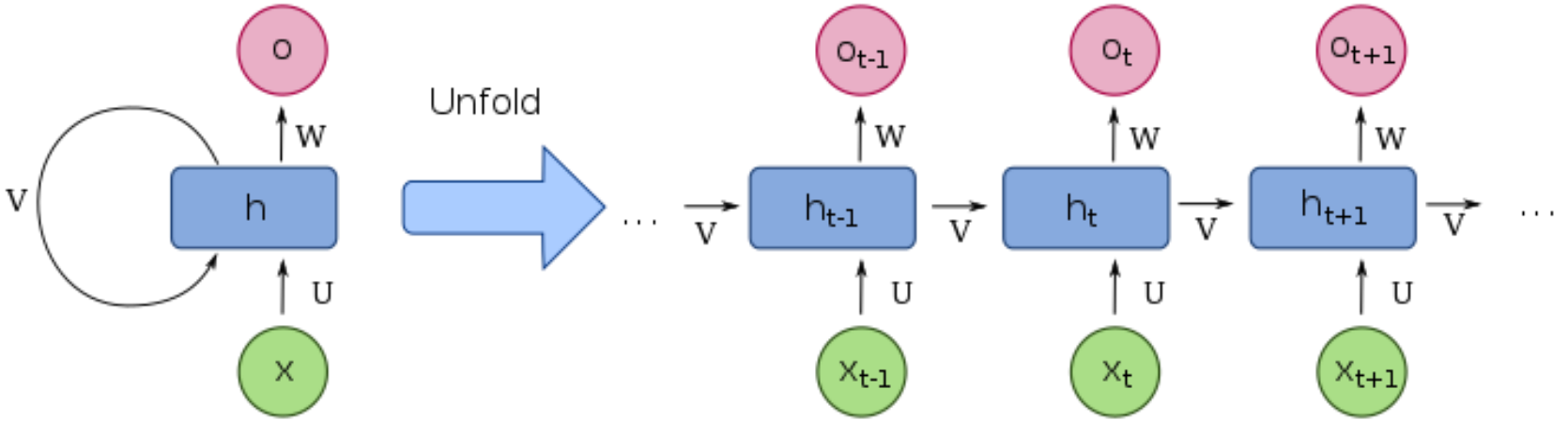


Photo Credit: https://fr.wikipedia.org/wiki/Fichier:Recurrent_neural_network_unfold.svg



ARQUITECTURA DE REDES NEURONALES

- Un RNR contiene un bucle temporal en el que la capa oculta no sólo da una salida sino que se alimenta a sí misma también.
- Se añade una dimensión extra que es el tiempo.
- El RNR puede recordar lo que sucedió en la anterior marca de tiempo, así que funciona muy bien con la secuencia de texto.

¡LOS RNRS TRABAJAN COMO LA MAGIA!

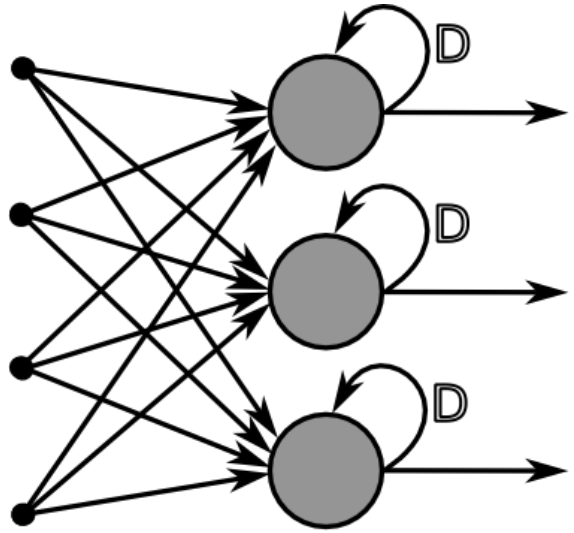
- Entrenaremos a los RNNs para que generen texto carácter por carácter y reflexionen sobre la pregunta "¿cómo es eso posible?“

- Fuente: The Unreasonable Effectiveness of Recurrent Neural Networks por Andrej Karpathy http://karpathy.github.io/2015/05/21/rnn-effectiveness/

¡¿QUÉ HACE QUE LAS REDES NEURONALES RECURRENTES SEAN TAN ESPECIALES?!

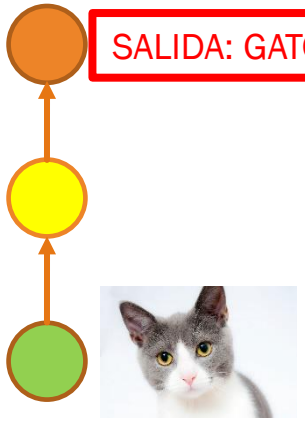
- Las RNA de avance están tan limitadas por su número fijo de entradas y salidas.
- Por ejemplo, una RNA tendrá una imagen de tamaño fijo (28x28) y genera una salida fija (clase o probabilidades).
- Las RNA de avance tienen una configuración fija, es decir, el mismo número de capas y pesos ocultos.
- Las redes neuronales recurrentes ofrecen una gran ventaja sobre las RNA de avance y son mucho más divertidas.
- Las RNN nos permiten trabajar con una secuencia de vectores:
  - Secuencia en las entradas
  - Secuencia en las salidas
  - ¡Secuencia en ambos!

RED NEURONAL RECURRENTE

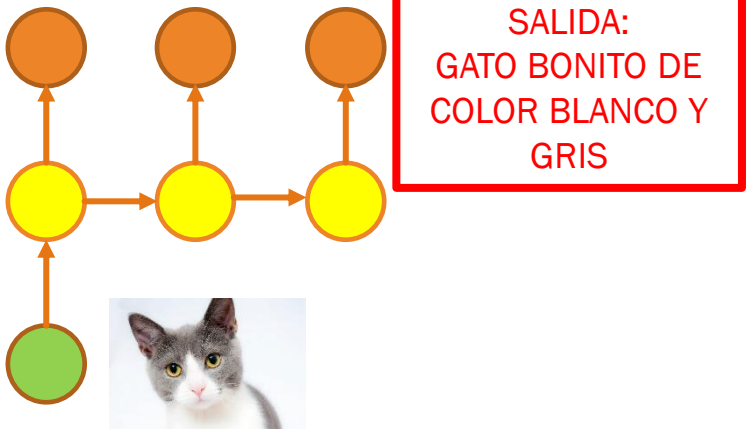


Source: The Unreasonable Effectiveness of Recurrent Neural Networks by Andrej Karpathy
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

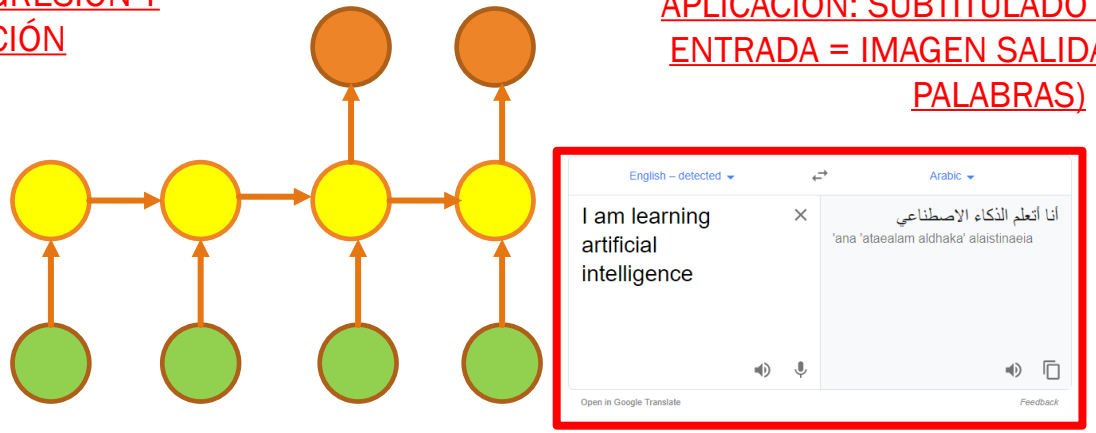
QUE HACE A LAS RNR TAN ESPECIALES



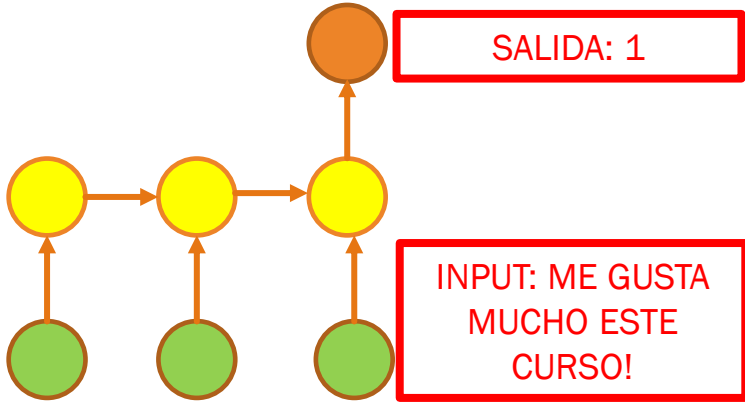
ONE TO ONE (VAINILLA)
APLICACIÓN: REGRESIÓN Y CLASIFICACIÓN



ONE TO MANY (OUTPUT DE SECUENCIA)
APLICACIÓN: SUBTITULADO DE LA IMAGEN, ENTRADA = IMAGEN SALIDAS = FRASE DE PALABRAS)

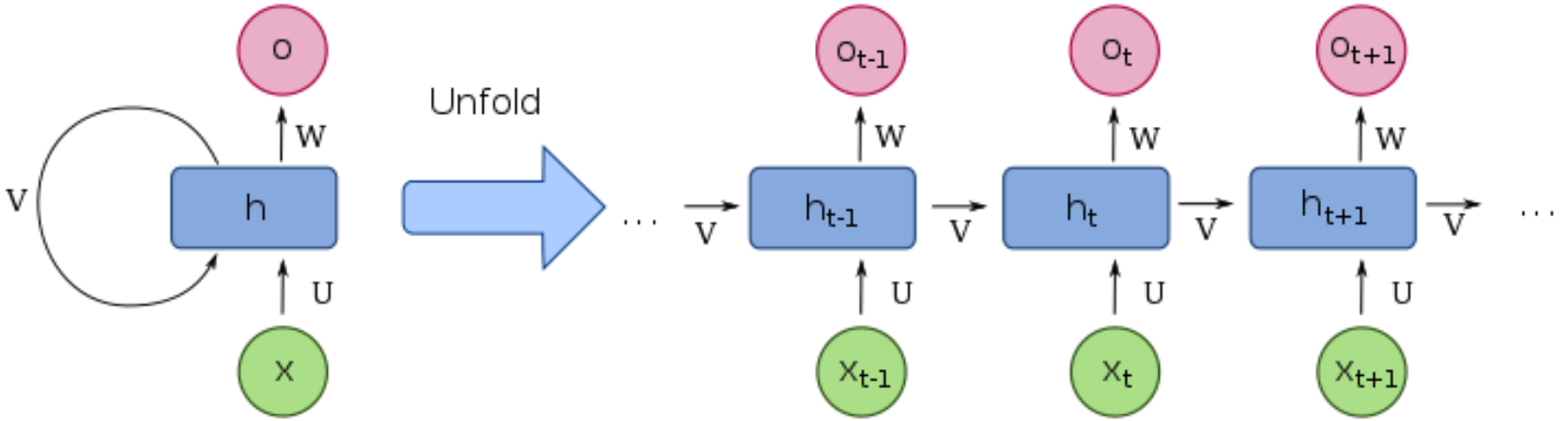


MANY TO MANY (SECUENCIA INPUT Y OUTPUT)
APLICACION: TRADUCCIÓN DEL LENGUAJE,



MANY TO ONE (SEQUENCE INPUT)
APLICACION: ANALISIS DE SENTIMIENTO,
EX: LA RESENA ES POSITIVA O NEGATIVA?

Photo Credit: https://fr.wikipedia.org/wiki/Fichier:Recurrent_neural_network_unfold.svg



MATEMATICA DE REDES RECURRENTES

- Un RNR acepta una entrada x y genera una salida o.
- La salida o no depende sólo de la entrada x, sino que depende del historial completo de las entradas que han sido alimentadas a la red en pasos de tiempo anteriores.
- Dos ecuaciones que rigen el RNN son las siguientes:
  - ACTUALIZACIÓN INTERNA DEL ESTADO:

    h_t=tanh($X\_t * U + h\_(t\text{-}1) * V$)
  - ACTUALIZACIÓN DE LA SALIDA:

    o_t= $softmax(\llbracket W * h \rrbracket\_t)$

DIVIRTIENDONOS CON REDES RECURRENTES!

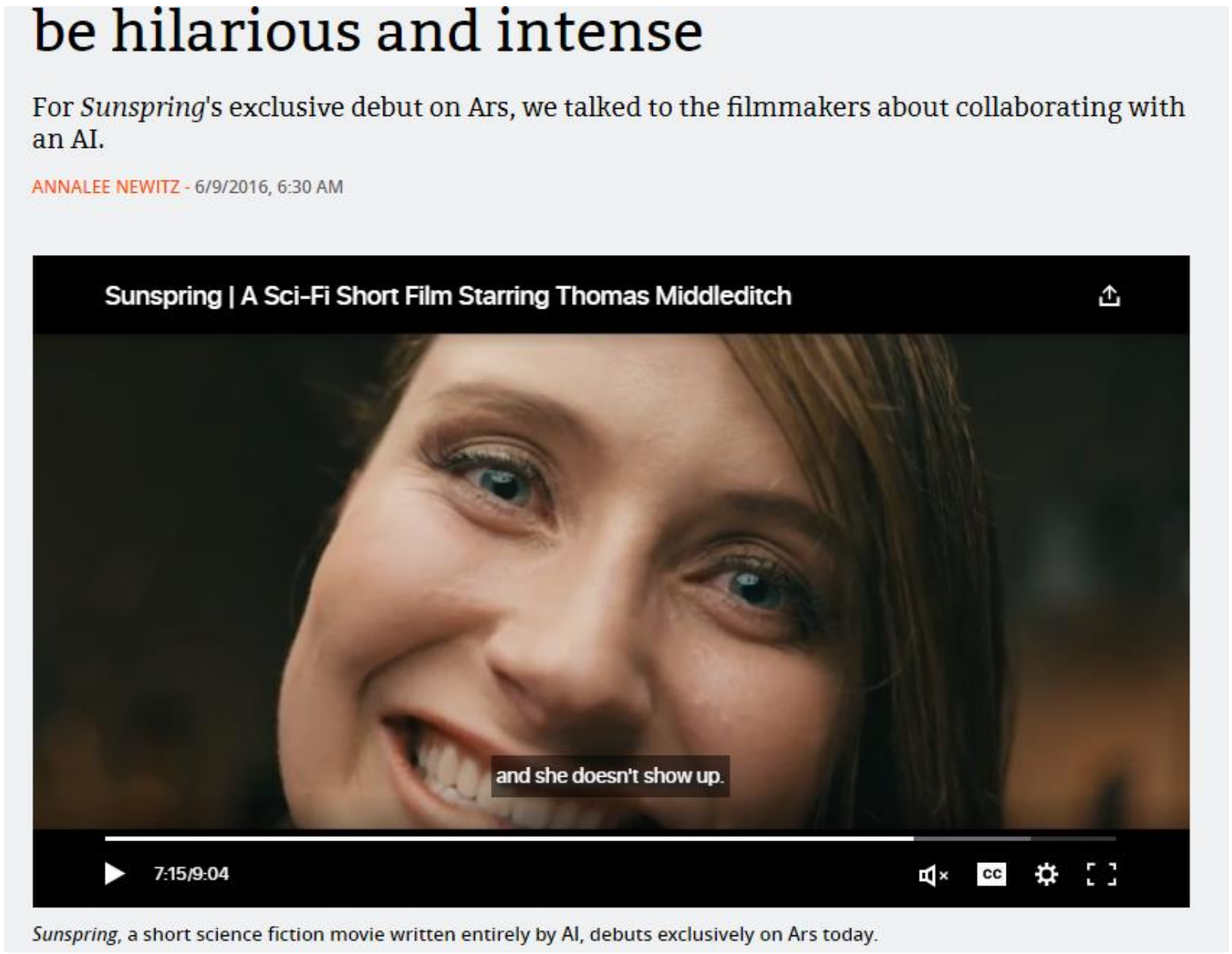
¡VEAMOS ESTA PELÍCULA ESCRITA POR UNA RNR!

- ¡Veamos una película escrita por AI!

- https://arstechnica.com/gaming/2016/06/an-ai-wrote-this-movie-and-its-strangely-moving/

- La película fue escrita por una red neural recurrente de LSTM

- La red LSTM fue entrenada con un corpus de docenas de guiones de ciencia ficción de películas de los años 80 y 90.



be hilarious and intense

For *Sunspring*'s exclusive debut on Ars, we talked to the filmmakers about collaborating with an AI.

ANNALEE NEWITZ - 6/9/2016, 6:30 AM

Sunspring, a short science fiction movie written entirely by AI, debuts exclusively on Ars today.

Photo Credit https://fr.wikipedia.org/wiki/Fichier:Recurrent_neural_network_unfold.svg

PROBLEMA DEL GRADIENTE DE DESAPARICIÓN

PROBLEMA DE GRADIENTE DE DESAPARICION

- Las redes LSTM funcionan mucho mejor comparadas con las RNN de vainilla, ya que superan el problema del gradiente de desaparición.
- El error tiene que propagarse a través de todas las capas anteriores resultando en un gradiente de desaparición.
- A medida que el gradiente se reduce, los pesos de la red ya no se actualizan.
- A medida que se añaden más capas, los gradientes de la función de pérdida se acercan a cero, haciendo que la red sea difícil de entrenar.

CADA CAPA DEPENDE DE LA SALIDA DE LAS CAPAS ANTERIORES, LA "V" SE MULTIPLICA VARIAS VECES RESULTANDO EN UN GRADIENTE QUE SE DESVANECE

0.1 * 0.1 * 0.1 *.....*0.1 =1e-10

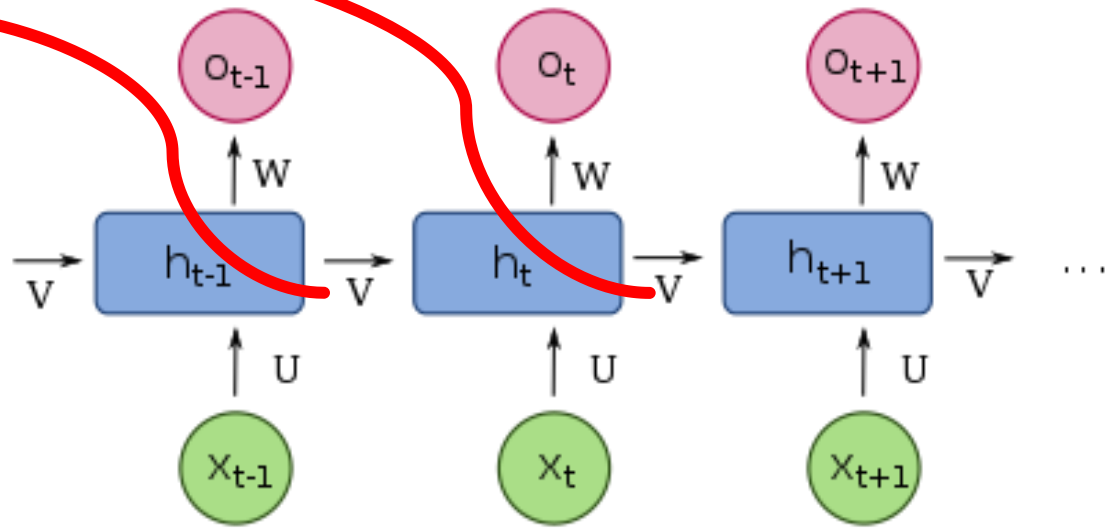
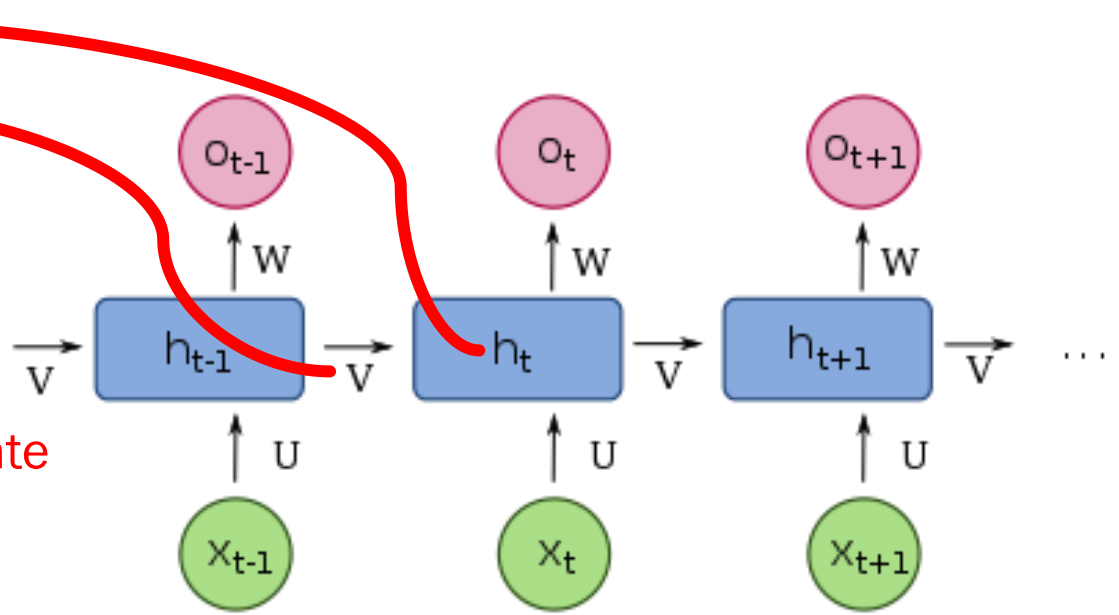


Photo Credit: https://fr.wikipedia.org/wiki/Fichier:Recurrent_neural_network_unfold.svg

PROBLEMA DE GRADIENTE DE DESAPARICION

- Los gradientes de la RNA se calculan durante la retropropagación.
- En la retropropagación, calculamos los derivados de la red pasando de la capa más externa (cerca de la salida) a las capas iniciales (cerca de las entradas).
- Durante este cálculo se utiliza la regla de la cadena, en la que los derivados de las capas finales se multiplican por los derivados de las capas iniciales.
- Los gradientes siguen disminuyendo exponencialmente y, por lo tanto, los pesos y los sesgos ya no se actualizan.

CADA CAPA DEPENDE DE LA SALIDA DE LAS CAPAS ANTERIORES, LA "V" SE MULTIPLICA VARIAS VECES RESULTANDO EN UN GRADIENTE QUE SE DESVANECE, (ex: 0.1 * 0.1 * 0.1 *.....*0.1 =1e-10)

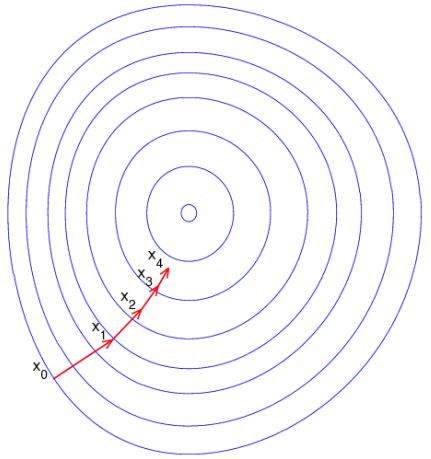


Nuevo Peso = Antiguo Peso – Rango de Aprendizaje * gradiente
9.09999 = 10.1 – 1 * 0.001

Photo Credit: https://fr.wikipedia.org/wiki/Fichier:Recurrent_neural_network_unfold.svg

DESCENSO DE GRADIENTE

- El descenso de gradiente es un algoritmo de optimización que se utiliza para obtener los valores optimizados de peso y sesgo de la red
- Funciona tratando iterativamente de minimizar la función de costo
- Funciona calculando el gradiente de la función de costo y moviéndose en dirección negativa hasta que se alcanza el mínimo local/global
- Si se toma el positivo del gradiente, se alcanza el máximo local/global



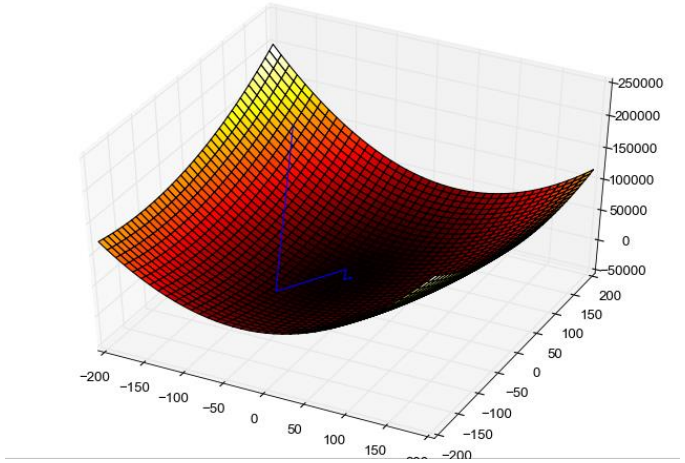


Photo Credit: https://commons.wikimedia.org/wiki/File:Gradient_descent_method.png
Photo Credit: https://commons.wikimedia.org/wiki/File:Gradient_descent.png

RANGO DE APRENDIZAJE

- El tamaño de los pasos que se dan se llama la tasa de aprendizaje
- Si la tasa de aprendizaje aumenta, el área cubierta en el espacio de búsqueda aumentará para que podamos alcanzar el mínimo global más rápido
- Sin embargo, podemos sobrepasar el objetivo
- Para pequeñas tasas de aprendizaje, el entrenamiento tomará mucho más tiempo para alcanzar valores de peso optimizados

DESCENSO DE GRADIENTE

El descenso gradual funciona de la siguiente manera:

- 1. Calcular la derivada (gradiente) de la función de pérdida
- 2. Elija valores aleatorios para los parámetros m, b y sustituya
- 3. Calcular el tamaño del paso (¿cuánto vamos a actualizar los parámetros?)

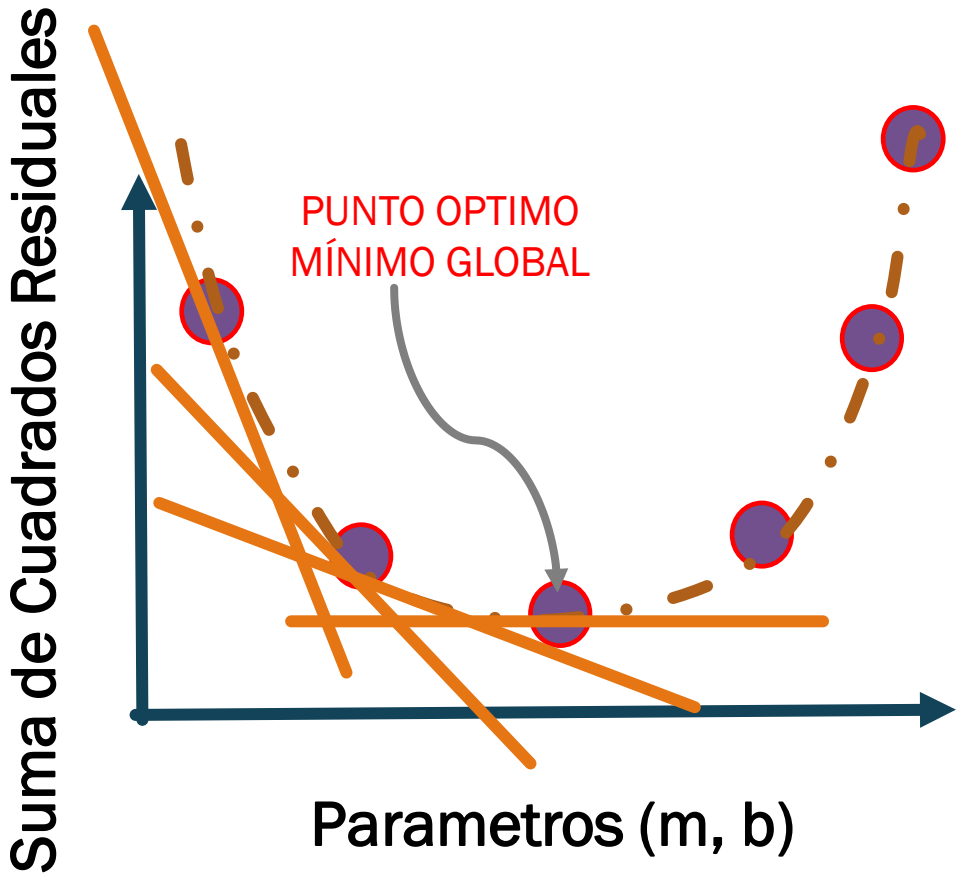
  Tamaño del paso = Pendiente ∗ Rango de Aprendizaje

- 4. Actualice los parámetros y repita

$$y = b + m * x$$

EL OBJETIVO ES ENCONTRAR LOS MEJORES PARÁMETROS

***Nota: en realidad, este gráfico es 3D y tiene tres ejes, uno para m, b y la suma de los residuos cuadrados**



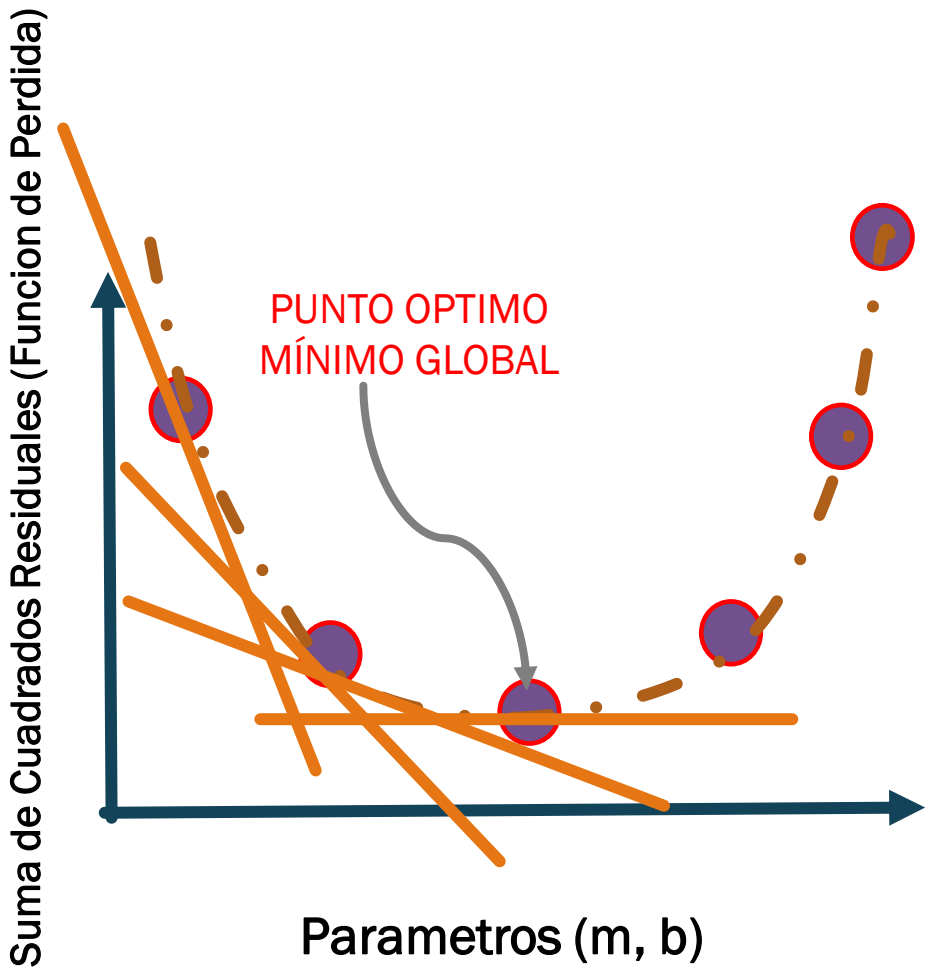
MATEMATICA DEL DESCENSO DE GRADIENTE

$$y = b + m * x$$

EL OBJETIVO ES ENCONTRAR LOS MEJORES PARÁMETROS

$$Funcoin\ de\ Perdida f(m,b) = \frac{1}{N}\sum_{i=1}^{n}\left(y_i - (b + m * x_i)\right)^2$$

$$gradiente\ f'(m,b) = \begin{bmatrix} \frac{df}{dm} \\ \frac{df}{db} \end{bmatrix} = \begin{bmatrix} \frac{1}{N}\sum_{i=1}^{n} -2x_i\left(y_i - (b + m * x_i)\right)^2 \\ \frac{1}{N}\sum_{i=1}^{n} -2\left(y_i - (b + m * x_i)\right)^2 \end{bmatrix}$$



**Nota: en realidad, este gráfico es 3D y tiene tres ejes, uno para m, b y la suma de los residuos cuadrados*

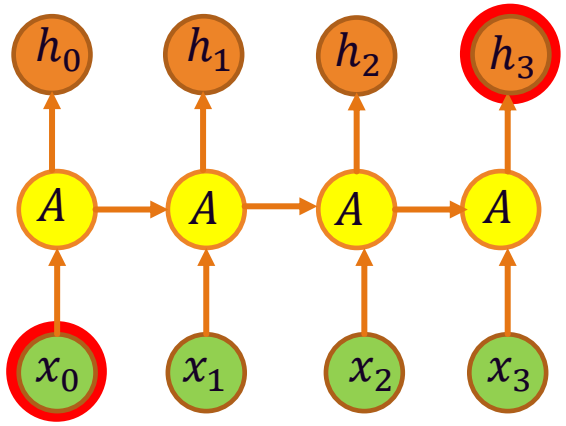
REDES DE MEMORIA A LARGO Y CORTO PLAZO (LSTM)

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

TEORIA LSTM

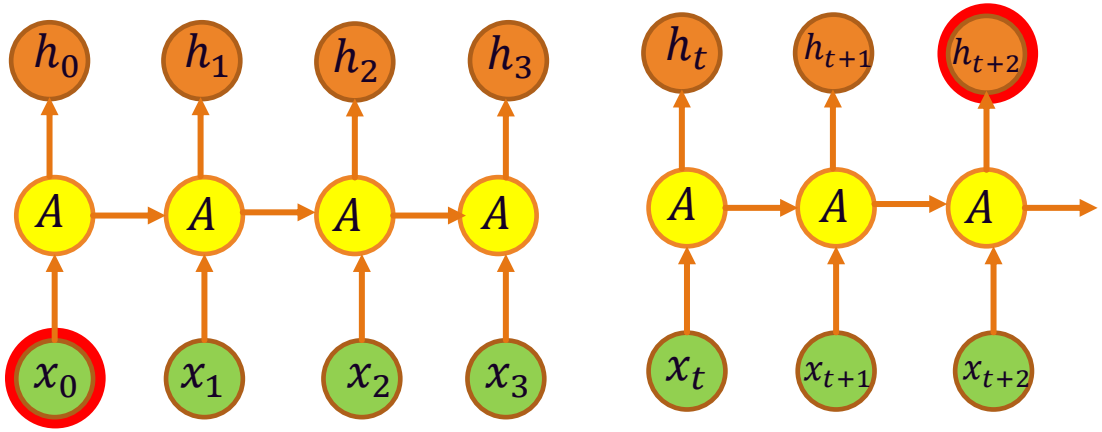
- Las redes LSTM funcionan mejor comparadas con las RNN de vainilla, ya que superan el problema del gradiente de desaparición.
- En la práctica, las RNN no logran establecer dependencias a largo plazo.
- Referencia: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

El color del árbol es "verde"



RNN FUNCIONA BIEN YA QUE LA BRECHA ENTRE LA PREDICCIÓN "VERDE" Y EL "ÁRBOL" DE INFORMACIÓN DE CONTEXTO NECESARIO ES PEQUEÑA

RNN FUNCIONA BIEN YA QUE LA BRECHA ENTRE LA PREDICCIÓN "VERDE" Y EL "ÁRBOL" DE INFORMACIÓN DE CONTEXTO NECESARIO ES PEQUEÑA



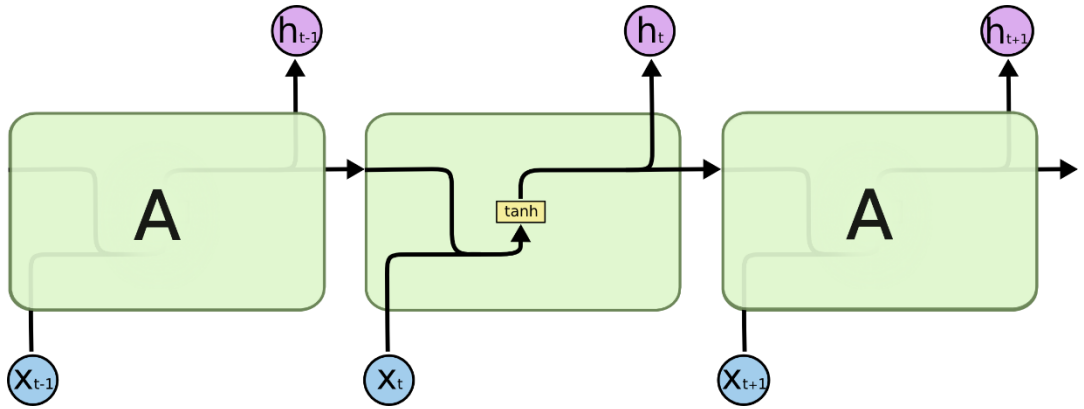
RNR FUNCIONA BIEN YA QUE LA BRECHA ENTRE LA PREDICCIÓN "VERDE" Y EL "ÁRBOL" DE INFORMACIÓN DE CONTEXTO NECESARIO ES PEQUEÑA

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

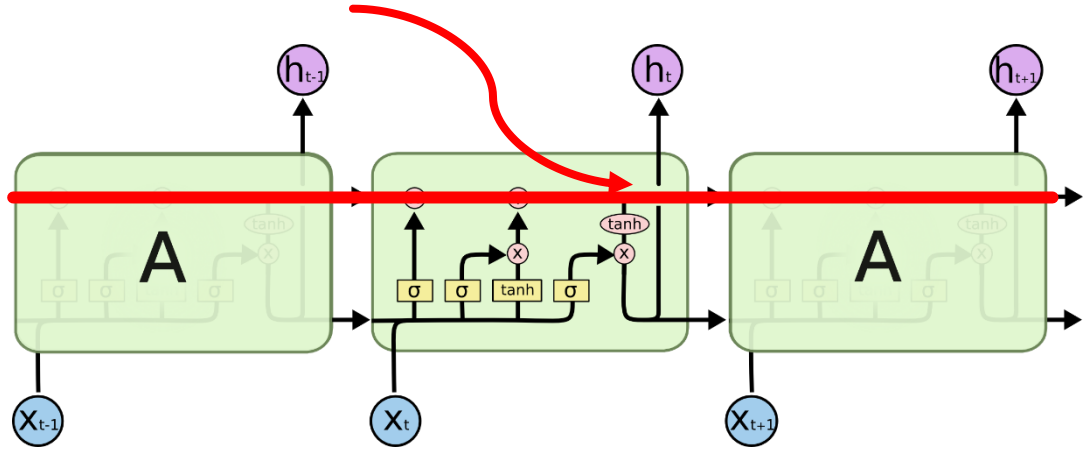
TEORIA LSTM

- Las redes LSTM son un tipo de RNN que están diseñadas para recordar las dependencias a largo plazo por defecto.
- El LSTM puede recordar y recordar información durante un período prolongado de tiempo.
- Recuerda que cada línea representa un vector completo.

ESTA LÍNEA HORIZONTAL (MEMORIA) O ESTADO CELULAR
PERMITE AL LSTM RECORDAR INFORMACIÓN MUY ANTIGUA



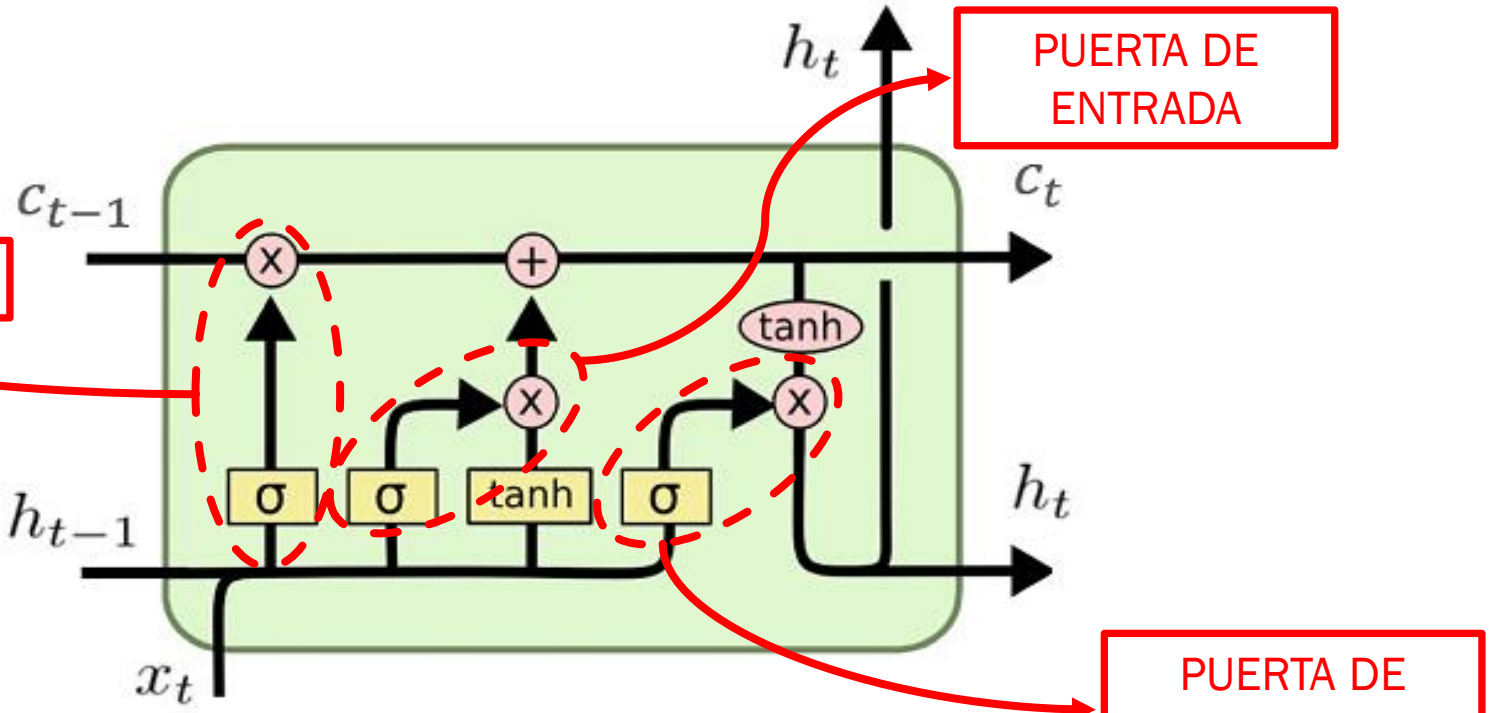
RED NEURONAL RECURRENTE DE VAINILLA



RED DE MEMORIA A LARGO Y CORTO PLAZO

LSTM TEORIA - PUERTAS

- El LSTM contiene puertas que pueden permitir o bloquear el paso de información.
- Las puertas consisten en una capa neural de red sigmoide junto con una operación de multiplicación puntual.
- La salida sigmoide va de 0 a 1:
- 0 = No permitir que ningún dato flu
- 1 = ¡permitir que todo fluya!



Referencia y Creditos de Foto: https://colah.github.io/posts/2015-08-Understanding-LSTMs/
Credito Foto: https://commons.wikimedia.org/wiki/File:LSTM.png

Referencia y Credito de Fotos: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

TEORIA LSTM - MATEMATICAS

$$f_t = \sigma(W_f . [h_{t-1}, x_t] + b_f) \qquad i_t = \sigma(W_i . [h_{t-1}, x_t] + b_i)$$

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

$$\tilde{C}_t = \tanh(W_c . [h_{t-1}, x_t] + b_c)$$

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

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