

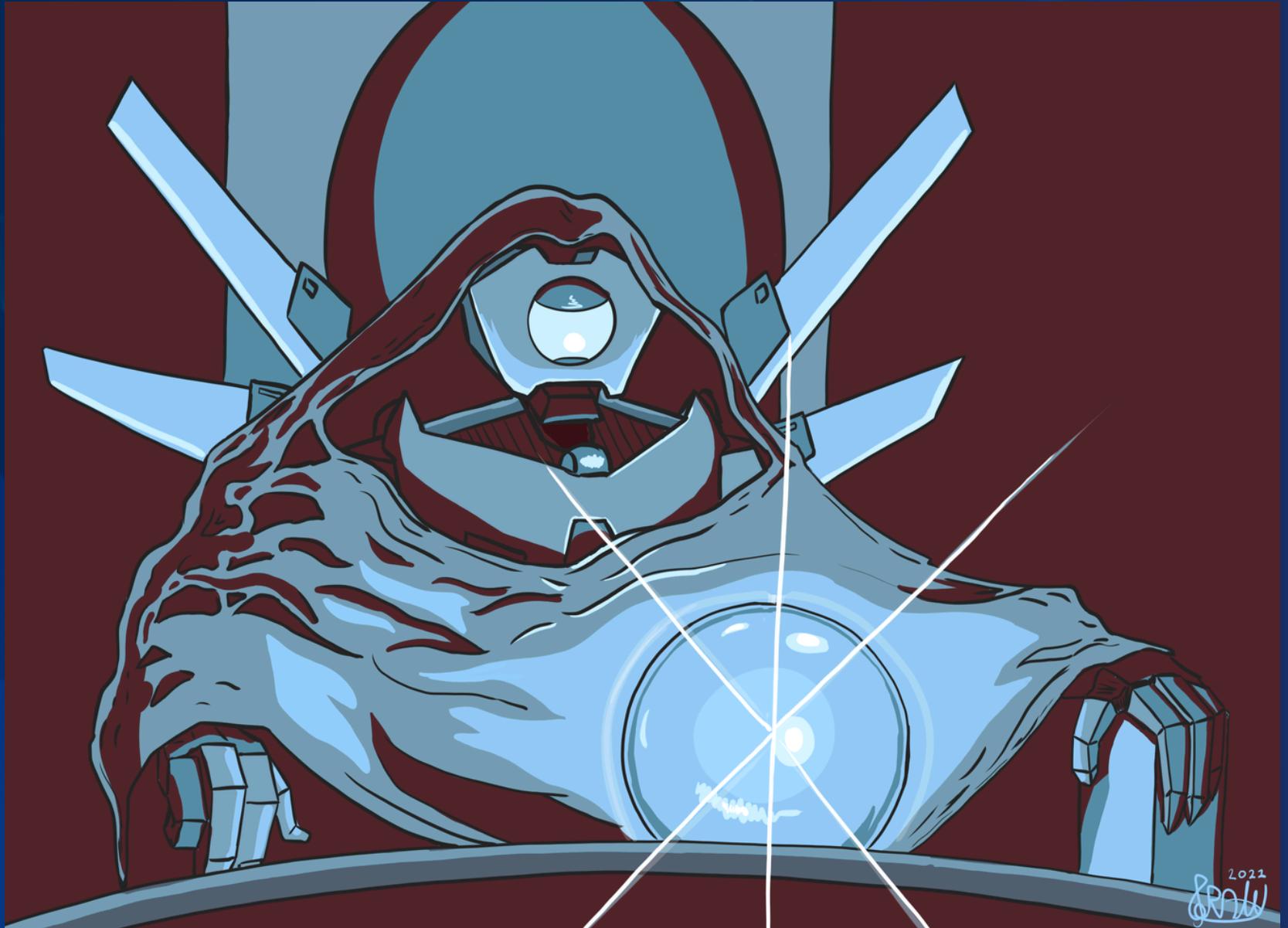


Alberto Carrillo

Redes Neuronales Recurrentes

02

¿Qué es una RNN?



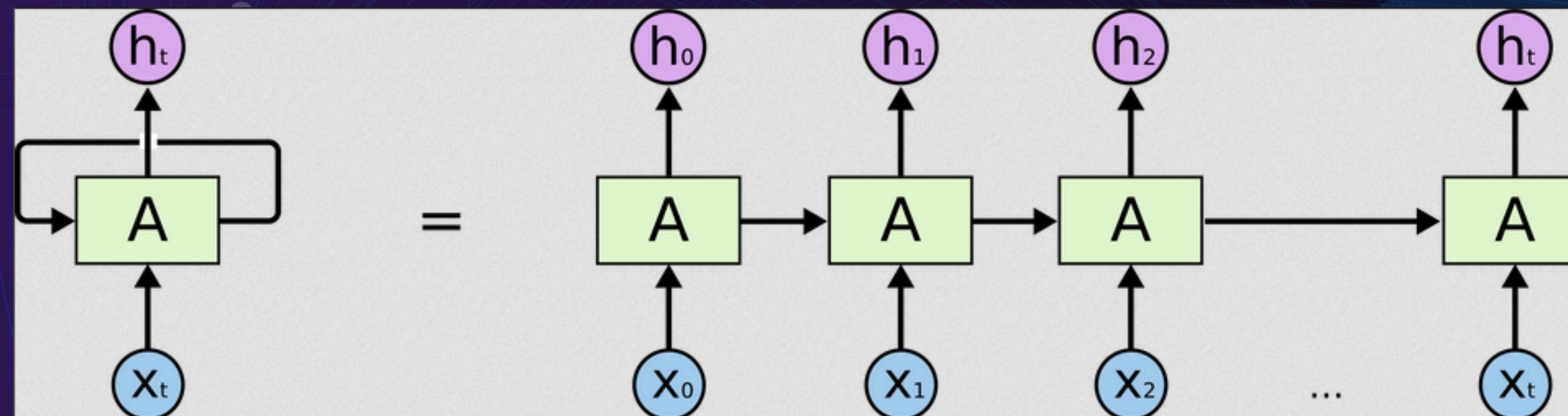
- *Para poder comprender o analizar ciertos sucesos, necesitamos entender sus pasos previos.*
- *Una red neuronal tradicional no puede considerar el contexto previo del evento que predice.*
- *Una RNN es capaz de persistir información utilizando bucles.*

Arquitectura

$$f_{\theta} = (x_t, h_t) \rightarrow (y_t, h_{t+1})$$

El concepto principal de una RNN básica (o “vanilla”) puede verse como una red neuronal tradicional con múltiples copias iguales, las cuales van “pasando” su mensaje a la red sucesora.

Para ello, definimos diferentes pasos temporales de la red neuronal, cada paso va a tomar un input (vector) que pasa a una capa oculta que combina el input actual, con el estado previo de la capa oculta.



Limitaciones



VANISHING GRADIENT

Los gradientes se multiplican repetidas veces entre pasos temporales.

El gradiente final puede reducirse hasta el punto de que el aprendizaje se “para”



EXPLODING GRADIENT

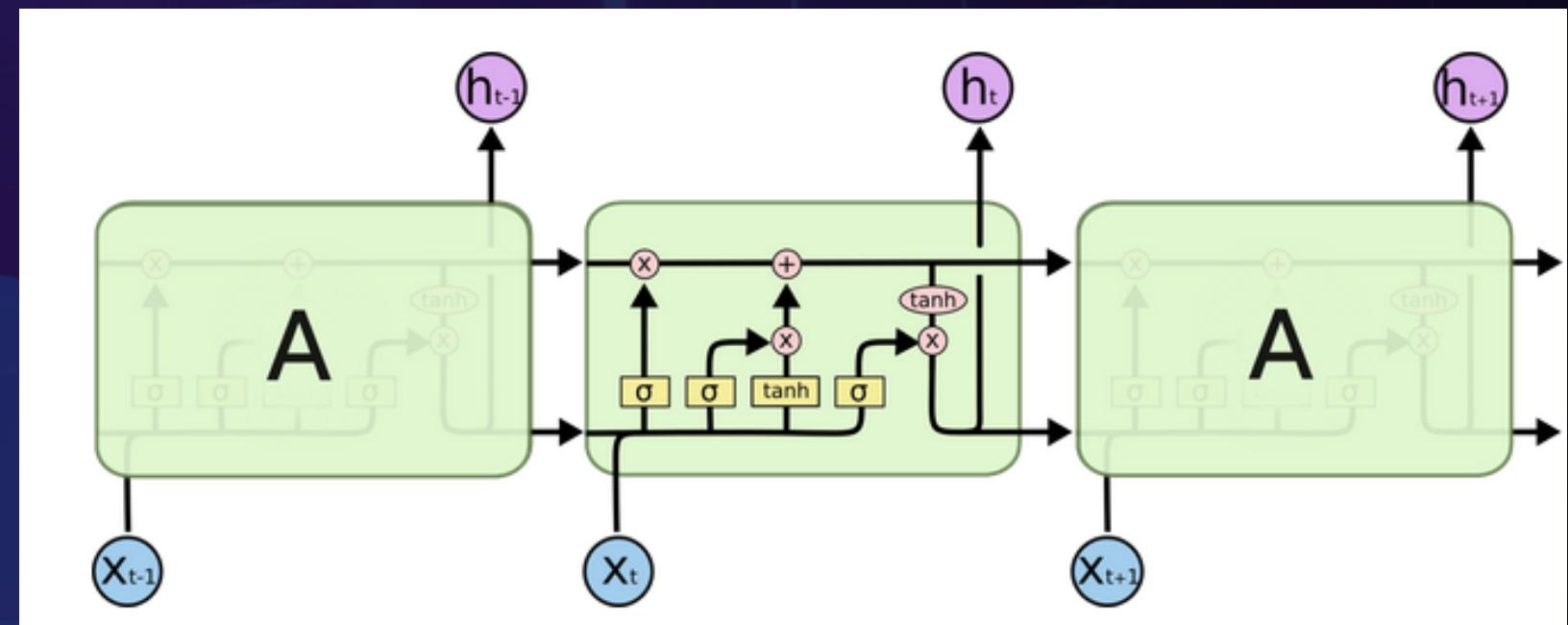
Cuando las derivadas del gradiente son > 1 los gradientes crecen exponencialmente.

Los pesos se vuelven inestables y excesivamente grandes.

05

LSTM Networks

- Capaces de entender dependencias a largo plazo
- Tienen el mismo diseño de bucles o pasos temporales.
- Disponen de una célula de memoria que se mantiene a lo largo de los pasos temporales.
- Esta célula se modifica con 4 redes neuronales o “puertas”.
 - Forget gate: Determina cuanta información eliminar del estado de memoria
 - Input gate: Determina información a almacenar
 - Candidate memory: Valores que pueden añadirse
 - Output gate: Determina la información a usar para el output.



Limitaciones

COMPUTACION

Muy costosas en recursos

OVERFITTING

Muchos parámetros hacen que sea susceptible a overfitting.

PARALELIZACIÓN

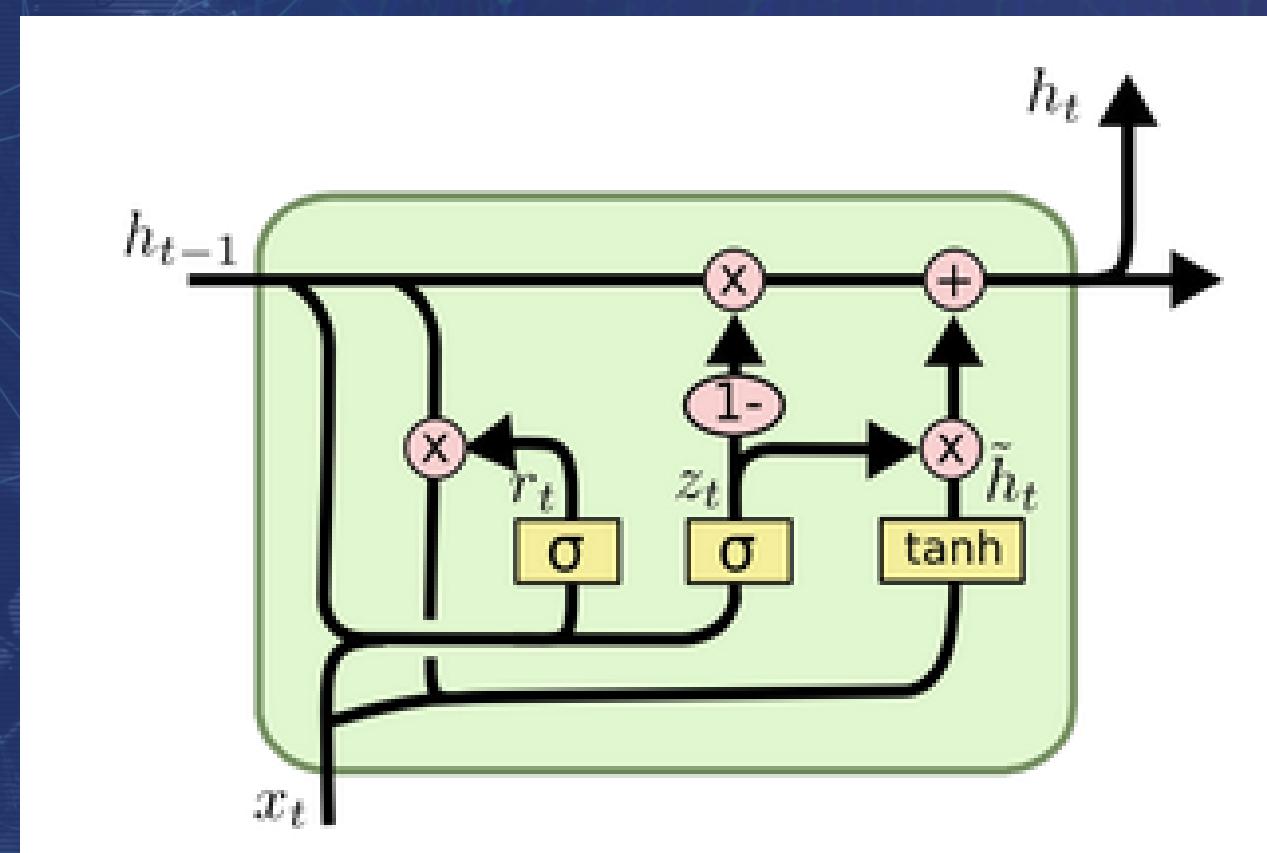
Limitadas para paralelización por su proceso secuencial.

HIPERPARÁMETROS

Muy sensibles a la selección de hiperparámetros.

GRUs

- Versión simplificada de la arquitectura LSTM
- Reducción de cuatro elementos a dos:
 - Reset gate: Cuanta información ovlidar del estado previo
 - Update gate: Cuanta información retener del estado previo, y cuanta información usar para actualizar el estado oculto.
- Menor consumo de recursos que LSTM



LSTM vs GRU

Característica	GRU	LSTM
Arquitectura	Más simple, 2 puertas (reset, update).	Más compleja, 3 puertas (input, forget, output) + estado de memoria separado.
Memoria	No tiene un estado de memoria separado.	Usa un estado de memoria separado y un estado oculto.
Tiempo de Entrenamiento	Más rápido debido a tener menos parámetros.	Más lento debido a más parámetros y cálculos.
Expresividad	Menor, con menos control granular sobre la memoria.	Mayor, con un control más detallado de la memoria.
Rendimiento	A menudo comparable o mejor en conjuntos de datos pequeños.	Mejor para tareas que requieren una gestión más compleja de la memoria.
Aplicaciones	Adecuado para tareas generales de modelado de secuencias.	Preferido para tareas que implican dependencias más intrincadas y secuencias más largas.

CUANDO USAR RNN

- *Análisis de series temporales*
 - *Tendencias de precios*
 - *Predicción meteorológica*
 - *Sensores IoT*
- *NLP*
 - *Análisis de sentimiento*
 - *Traducción*
 - *Generación de texto*
 - *Named Entity Recognition*

- *Reconocimiento de voz*
- *Procesamiento de audio*
 - *Restauración*
- *Análisis de video o secuencias*



Comparativa

Escenario	Modelo Recomendado	Razón
Secuencias cortas y tareas simples	Vanilla RNN	Es eficiente, simple y suficiente para dependencias a corto plazo.
Secuencias moderadas y apps en tiempo real	GRU	Balance entre rendimiento y eficiencia; ideal para baja latencia.
Secuencias largas y dependencias complejas	LSTM	Captura relaciones a largo plazo con control detallado de memoria.
Datos pequeños o limitados	GRU	Requiere menos parámetros, lo que facilita generalizar con menos datos.
Tareas complejas de lenguaje natural	LSTM	Gestiona oraciones largas y relaciones complejas (traducción, resumen).
Pronósticos de series temporales simples	Vanilla RNN	Adecuado para tareas básicas como tendencias a corto plazo.
Reconocimiento de voz en tiempo real	GRU	Más rápido que LSTM, pero suficientemente potente para dependencias moderadas.
Análisis de video o datos ruidosos	LSTM	Robusto ante datos complejos o ruido, ideal para tareas exigentes.

Recursos utilizados y recomendados

- <https://www.deeplearningbook.org/contents/rnn.html>
- https://en.wikipedia.org/wiki/Recurrent_neural_network
- <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* 🐐
- *Good ol' GPT*
- *Pascanu, R., Mikolov, T., & Bengio, Y. (2013). "On the difficulty of training recurrent neural networks"*

Thank You!

