

Redes Neuronales Recurrentes

Introducción a las Redes de Elman



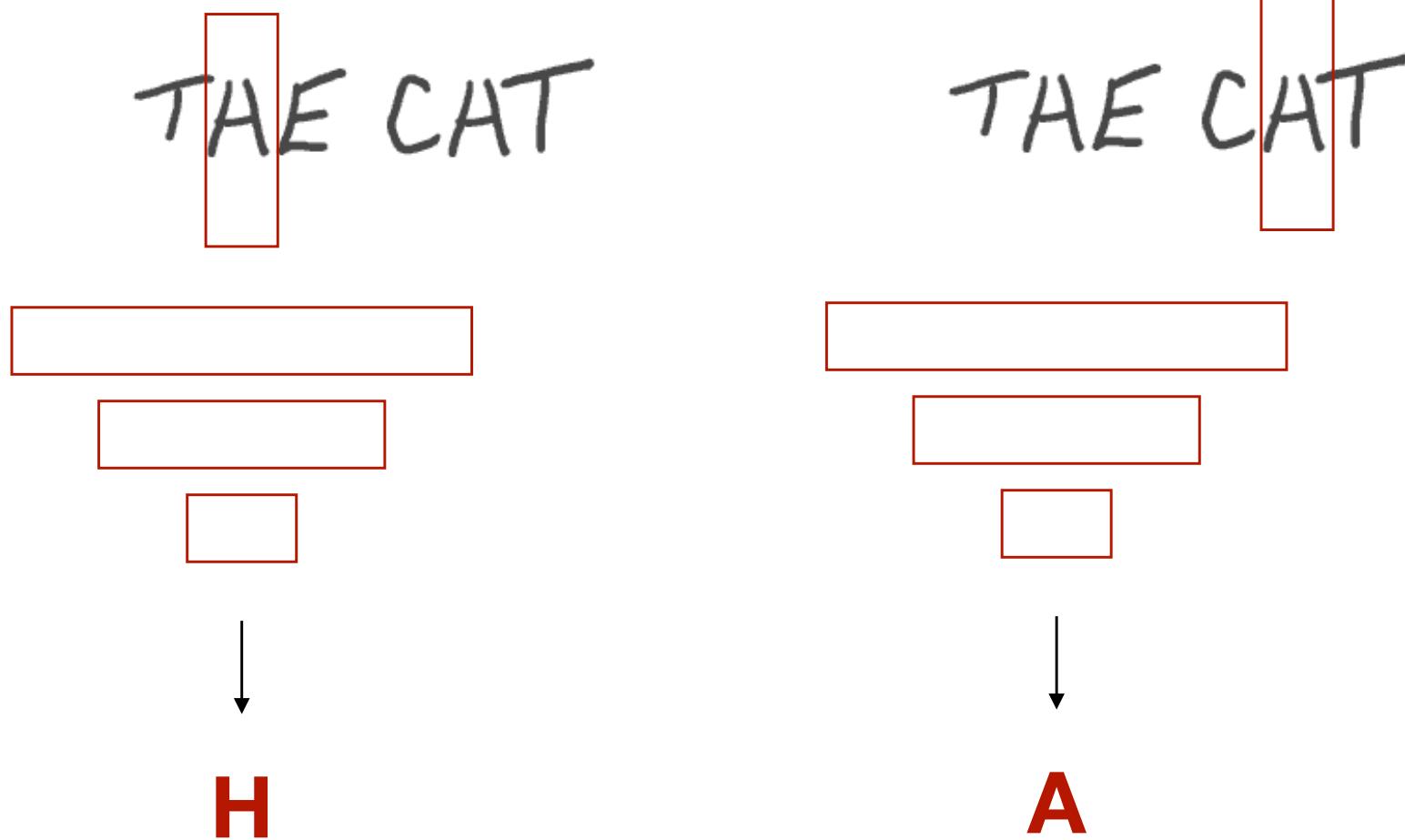
Prof. Ricardo Ñanculef - Departamento de Informática UTFSM

Memoria & Contexto

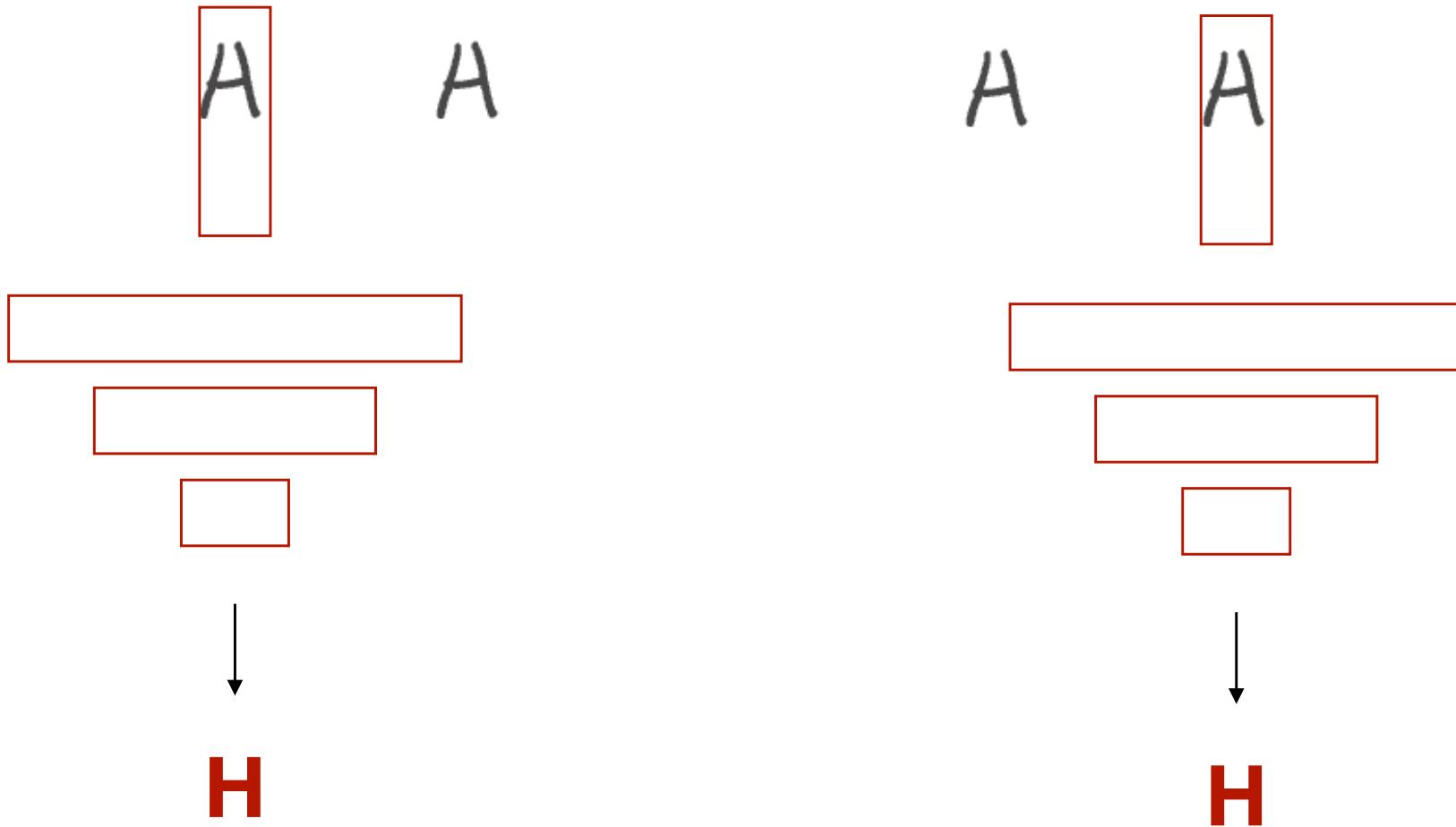
TAE CAT



Memoria & Contexto



Memoria & Contexto



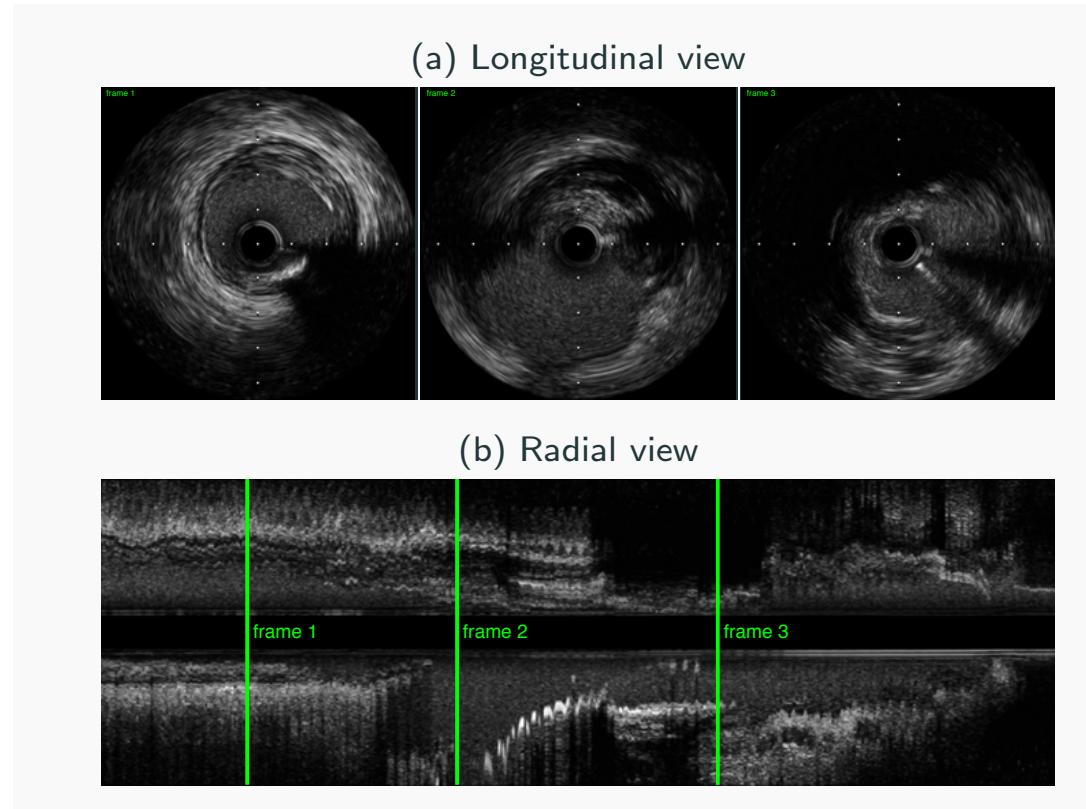
Memoria & Contexto

- Los modelos vistos hasta ahora implementan funciones entrada-salida **libres de contexto**: si el input es el mismo en dos momentos distintos del “tiempo”, el output será el mismo, independiente del “contexto” en que se encuentre.



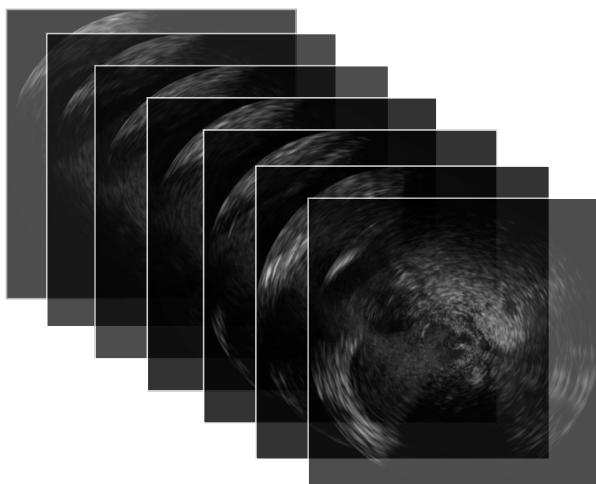
Memoria & Contexto

- Los modelos vistos hasta ahora implementan funciones entrada-salida **sin memoria de su decisiones anteriores o de sus propios estados anteriores.**



Memoria & Contexto

- Los modelos vistos hasta ahora implementan funciones entrada-salida **sin memoria de su decisiones anteriores o de sus propios estados anteriores.**

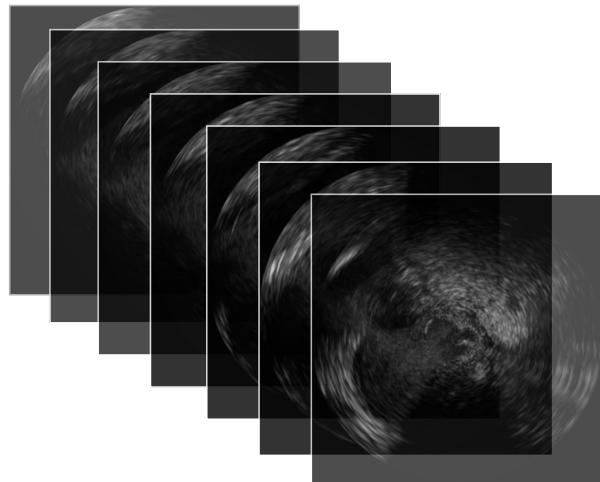


clasificación framewise



Memoria & Contexto

- Los modelos vistos hasta ahora implementan funciones entrada-salida
sin memoria de su decisiones anteriores o de sus propios estados anteriores.



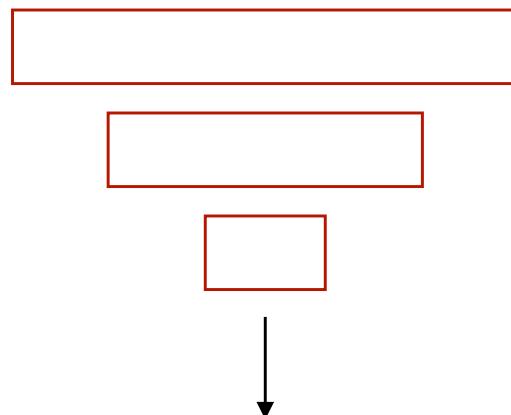
clasificación contextual



Orden

- La ausencia de memoria de su estado interno hace que esos modelos sean inapropiados para modelar tareas intrínsecamente secuenciales y aprender de datos temporales donde el orden es fundamental.

He threw the cat on the sofa



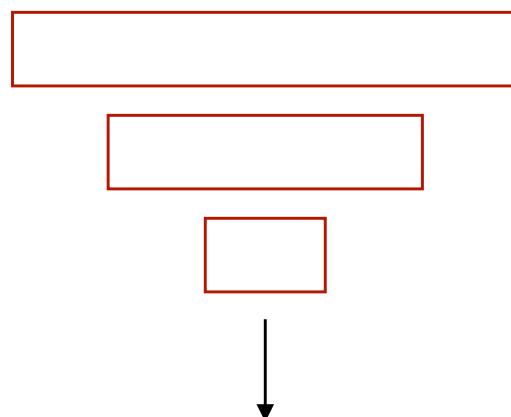
not violent



Orden

- La ausencia de memoria de su estado interno hace que esos modelos sean inapropiados para modelar tareas intrínsecamente secuenciales y aprender de datos temporales donde el orden es fundamental.

He threw the sofa on the cat



violent



Ejemplo: Sentiment Analysis.

- Queremos aprender a asignar una polaridad (e.g. sentimiento positivo/negativo) a una frase de entrada.
- Para poder obtener una entrada de largo fijo el modelo tradicional de representación de texto rompe/ignora la estructura temporal de las palabras, contando la frecuencia global de aparición en la frase.

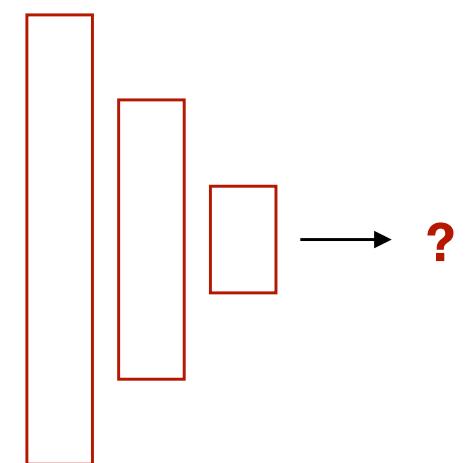
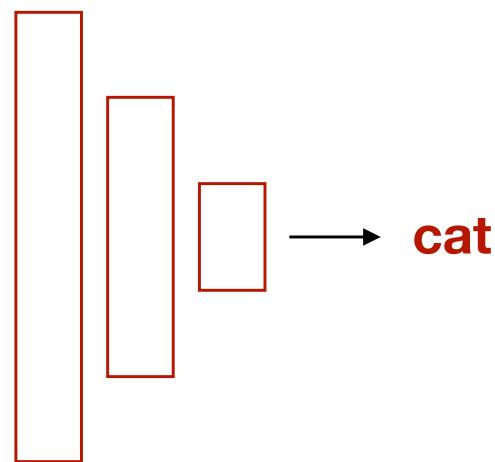
He threw the cat on the sofa +

He threw the sofa on the cat -



Complejidad Fija

- Los modelos vistos hasta el momento procesan inputs de tamaño fijo aplicando un grafo de computación cuya “complejidad” es fija



Input de Tamaño Fijo

- Los modelos vistos hasta el momento procesan inputs de tamaño fijo (vectores o volúmenes).

#1	The	quick	brown	fox	jumps	over	the	lazy	dog
#2	The	quick	brown	fox	jumps	over	the	lazy	dog
#3	The	quick	brown	fox	jumps	over	the	lazy	dog
#4	The	quick	brown	fox	jumps	over	the	lazy	dog



Ejemplo: Phrase Completion

- Queremos entrenar un modelo para auto-completar frases a medida que una persona escribe.
- En este ejemplo es aún más claro que la “decisión” de la red debiese depender de lo que la persona haya escrito antes.

**Wayne Dohey is a criminal that has been abusing children from years.
This priest should be sent to the church.**

**Wayne Dohey is a criminal that has been abusing children from years.
This priest should be sent to heaven.**

**Wayne Dohey is a criminal that has been abusing children from years.
This priest should be sent to jail.**



Ejemplo: Phrase Completion

- Enfoque clásico: usar ventanas de contexto de un determinado largo (fijo) de manera que la “decisión” de la red dependa de las ultimas k palabras.
 - Si no queremos que el modelo “explote” en número parámetros sólo podemos aprender dependencias cortas.

```
#5 The quick brown fox jumps over the lazy dog
#6 The quick brown fox jumps over the lazy dog
#7 The quick brown fox jumps over the lazy dog
```



Output de Tamaño Fijo

- Los modelos vistos hasta el momento producen outputs de tamaño fijo, (escalares, vectores o volúmenes).



A woman is throwing a frisbee in a park.

Xu, Kelvin, et al. "Show, attend and tell: Neural image caption generation with visual attention." *International Conference on Machine Learning*. 2015.



Ejemplo: Captioning

- Supongamos que queremos aprender a generar un texto que describa una determinada imagen de entrada.



A woman is throwing a frisbee in a park.

- Disponemos de una vocabulario de salida V y muchos ejemplos.

Ejemplo: Captioning

- **Modelo 1:** Pre-indexar todas las frases posibles, asignándoles un número o categoría.
- Número de neuronas de salida crecería exponencialmente en el vocabulario y consecuentemente el número de parámetros (problemas computacionales y estadísticos).



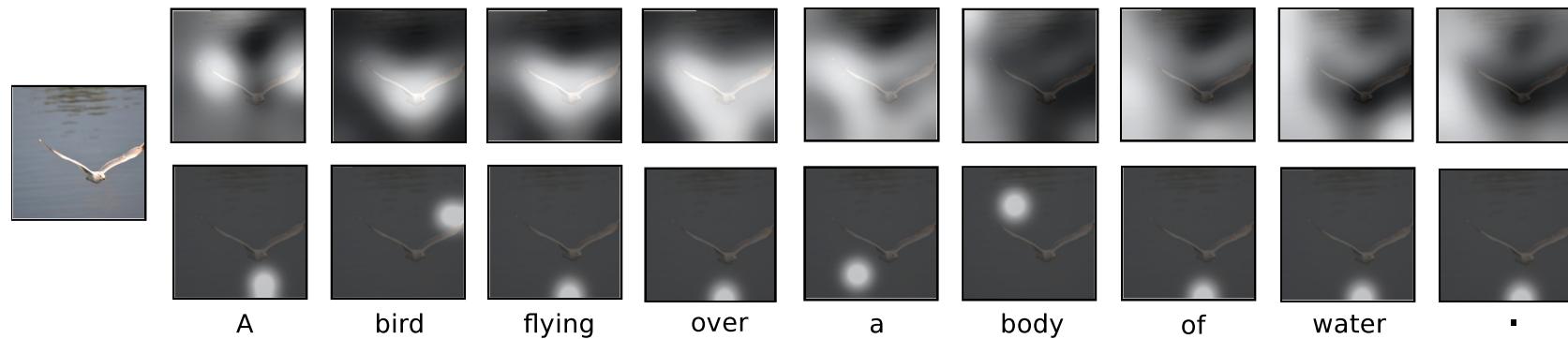
Ejemplo: Captioning

- **Modelo 2:** Output es una “imagen” (arreglo bi-dimensional) de tamaño cuadrático en el tamaño del vocabulario.
- Número de neuronas y parámetros sigue siendo muy grande (problemas computacionales y estadísticos). No estamos modelando la relación entre una palabra y la siguiente.

Mark	1	0	0	0	0	0	0	0
plays	0	1	0	0	0	0	0	1
Basketball	0	0	1	0	0	0	0	0
PAD	0	0	0	0	0	0	0	0
PAD	0	0	0	0	0	0	0	0
PAD	0	0	0	0	0	0	0	0
PAD	0	0	0	0	0	0	0	0

Ejemplo: Captioning

- El problema de fondo es que la salida de la red debe ser de largo variable. Idealmente la red debiese producir una palabra a la vez, tomando en cuenta aquello que haya producido hasta el momento.



Redes Recurrentes

- Redes especializadas en el procesamiento de secuencias.
- Redes capaces de “recordar” lo que han procesado anteriormente y “decidir una respuesta” **tanto en función de lo que entra** de un determinado momento **como de sus estados anteriores**.
- Capaces de hacer todo esto de modo parsimonioso, i.e. sin “explotar” en número de parámetros.
- Capaces de procesar entradas de largo variable y producir salidas de largo variable.



Modelo Básico (Elman)

COGNITIVE SCIENCE 14, 179–211 (1990)

Finding Structure in Time

JEFFREY L. ELMAN

University of California, San Diego

Time underlies many interesting human behaviors. Thus, the question of how to represent time in connectionist models is very important. One approach is to represent time implicitly by its effects on processing rather than explicitly (as in a spatial representation). The current report develops a proposal along these lines first described by Jordan (1986) which involves the use of recurrent links in order to provide networks with a dynamic memory. In this approach, hidden unit pat-

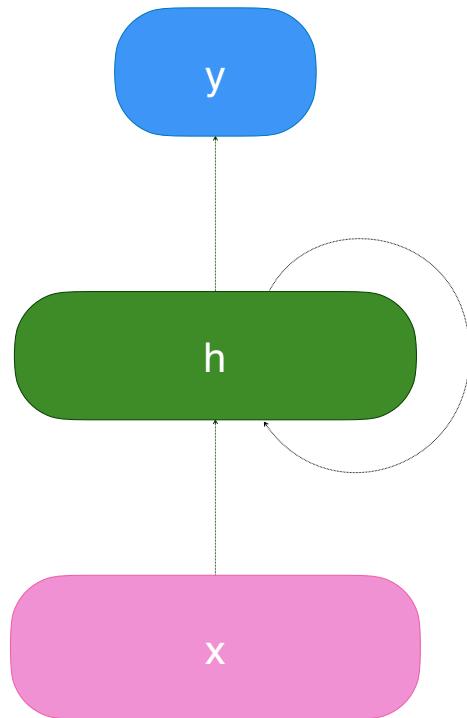
THE PROBLEM WITH TIME

One obvious way of dealing with patterns that have a temporal extent is to represent time explicitly by associating the serial order of the pattern with

Jeffrey Locke Elman (January 22, 1948 – June 28, 2018)



Modelo Básico (Elman)



- Como antes, la salida de red depende de su estado “interno” (capa de neuronas ocultas, 1 o más).
- El estado de la red (h) es una función del input (x), pero también del estado de la red en el “tiempo” anterior.
- De esta forma, el output es una función del input, pero también una función de todo lo que la red haya visto antes.
- Introducimos así la idea de “tiempo”. La red ya no es una función “estática” del input de entrada (x) sino que es un sistema dinámico.

Modelo Básico (Elman)

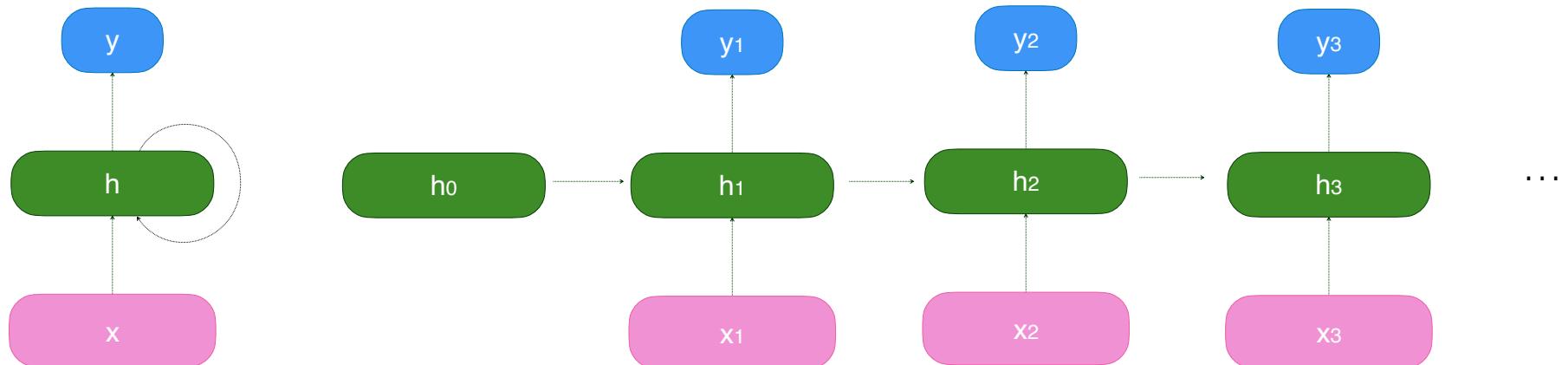
- Más formalmente, la red toma como input **secuencias $x_1x_2x_3 \dots$** de entrada y produce como output **secuencias $y_1y_2y_3$** , procesando un elemento a la vez. Partiendo de un estado inicial h_0 , las salidas quedan definidas mediante ecuaciones de la forma:

$$y_t = g_o(h_t)$$
$$h_t = g_h(x_t, h_{t-1})$$

- Cuando decimos “tiempo” nos referimos entonces al conjunto que indexa las secuencias de entrada y salida $t=1,2,3\dots$

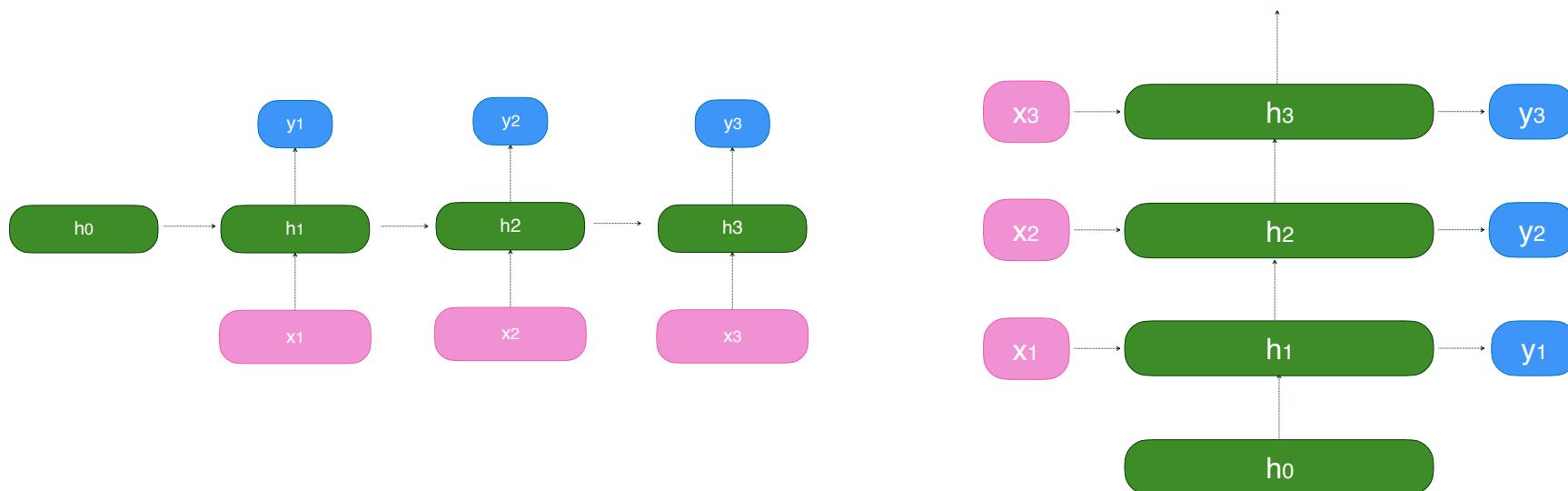
Desenrollado de la Red

- Un modo alternativo de representar la secuencia de cómputo ejecutada por la red consiste en “desenrollar” el grafo anterior (unfolding).



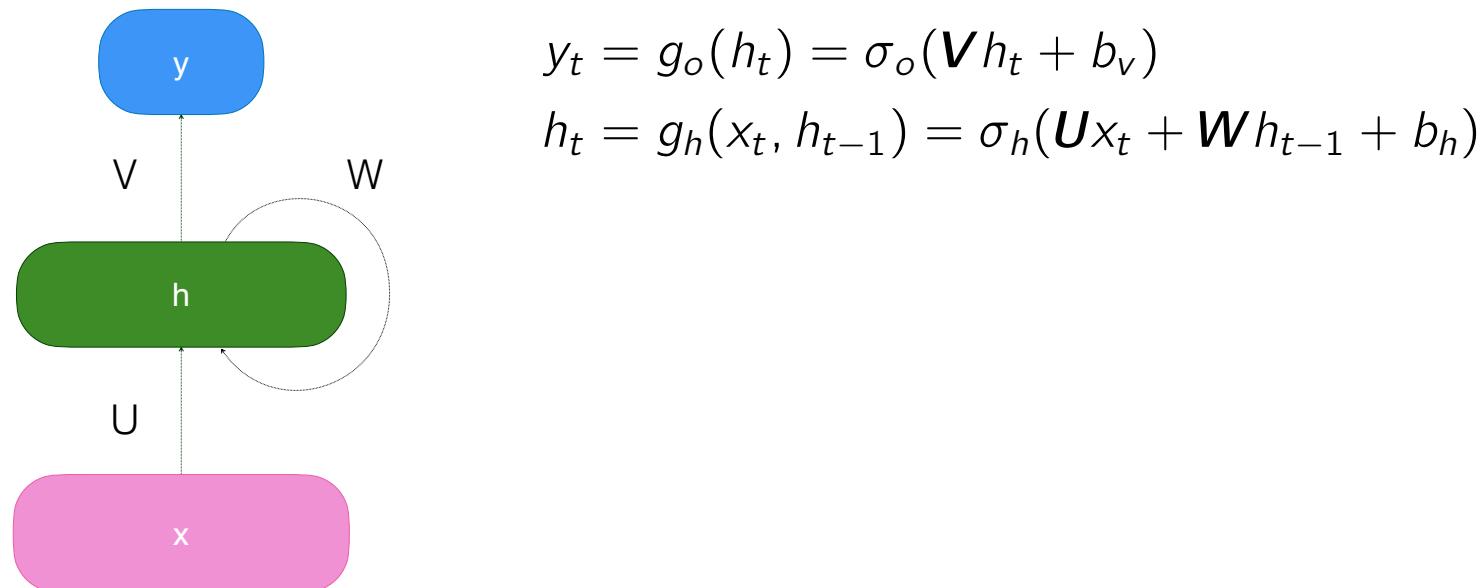
Desenrollado de la Red

- Esta representación deja en evidencia que una red recurrente es una red muy profunda, donde cada “tiempo” de la secuencia a procesar define una capa.



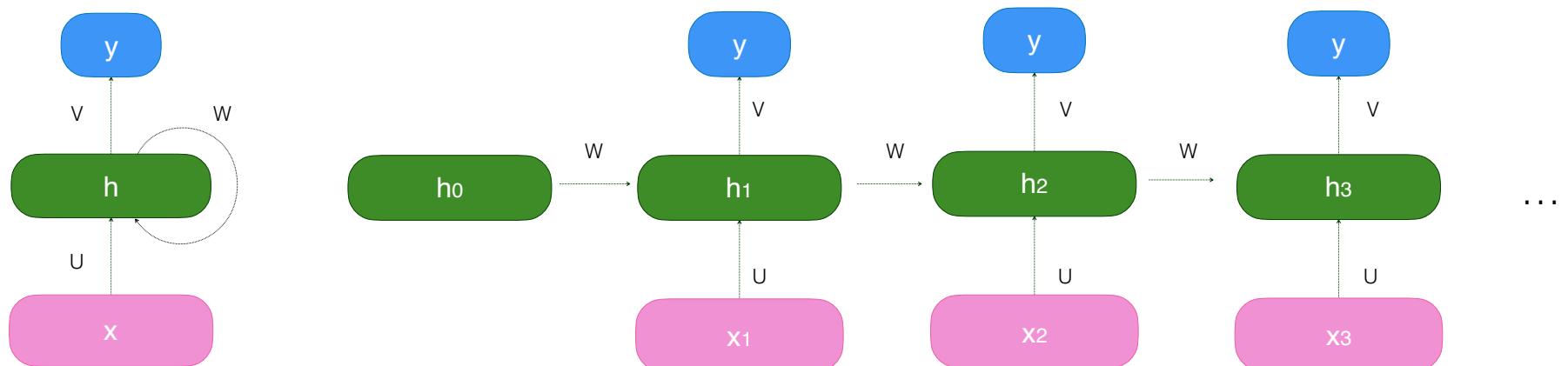
Modelo Básico (Elman)

- Para implementar las transformaciones g_o y g_h podemos reciclar las “capas” que hemos usado hasta el momento.
- En particular, si los elementos de la secuencia son vectores $x_t \in \mathbb{R}^d$ podemos usar capas tradicionales (densas) de la forma: transformación lineal + no-linealidad elemento a elemento.



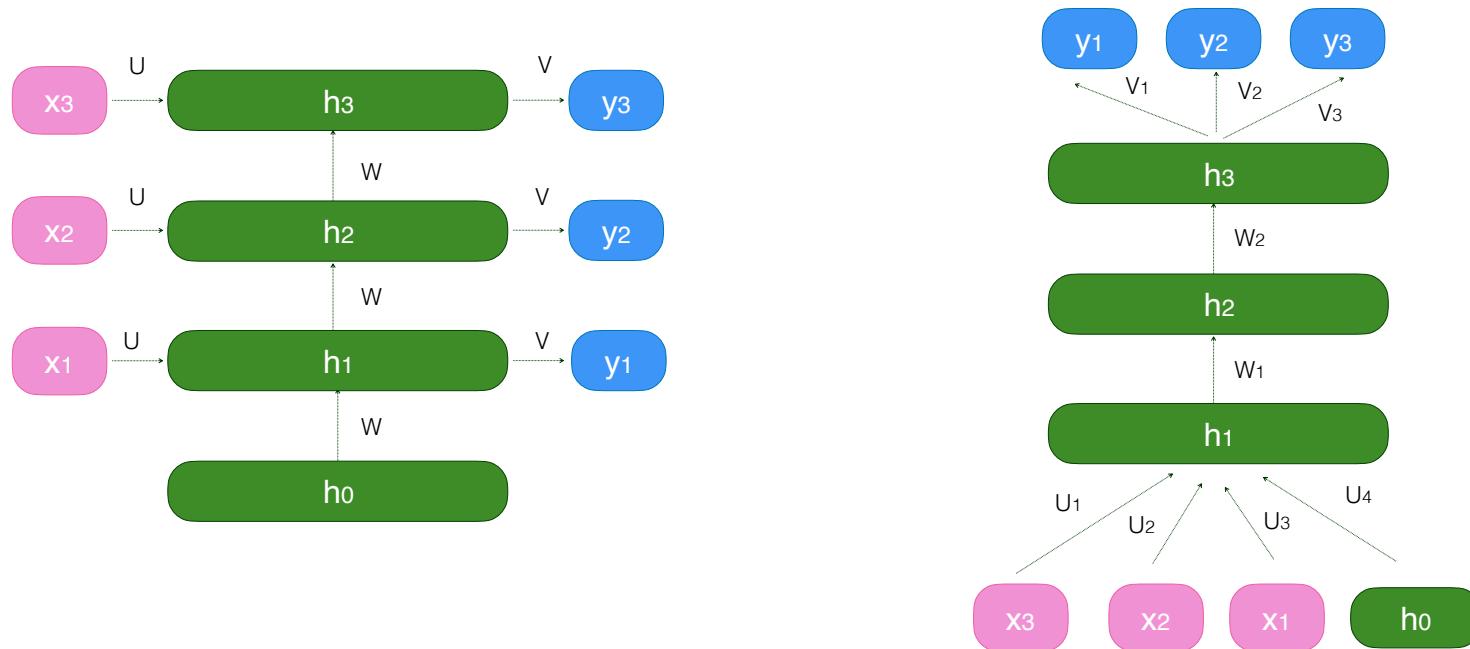
Compartición de Parámetros

- Si desenrollamos la red, es muy importante notar que los parámetros de las distintas capas que aparecen son **parámetros compartidos**.



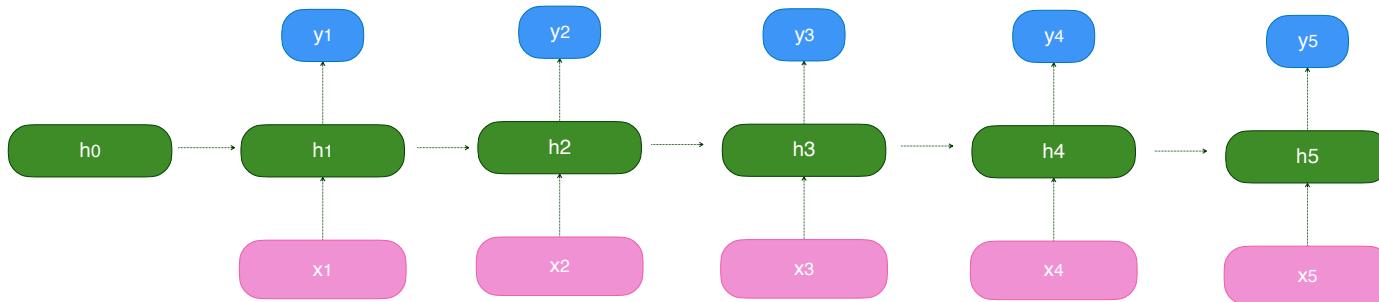
Compartición de Parámetros

- Esto contrasta con el número significativamente mayor de parámetros que tendría una arquitectura tradicional de profundidad equivalente y capaz de procesar la misma secuencia.



Input de Largo Variable

- Más interesantemente aún, con esta definición no necesitamos establecer a-priori el largo de las secuencias a procesar*. Si llega una secuencia más larga la red se sigue desenrollando ...



* En la práctica, por motivos computacionales, se requiere de todos modos fijar un largo máximo de secuencia T , pero eso no significa que las secuencias deban ser todas iguales. Además, sigue siendo correcto afirmar que la estructura recurrente se traduce en un ahorro notable de parámetros con respecto a una red tradicional que define a-priori el largo de las secuencias.

Entonces ...

- En una red neuronal recurrente los estados internos y la salida del modelo dependen de los datos que se hayan recibido en ese momento, pero también de los estados o salidas anteriores. Las dependencias entre tiempos se denominan recurrencias.
- El modelo más sencillo es el modelo de Elman, en que las recurrencias se restringen a una capa oculta consigno misma en el tiempo anterior.
- Estas redes se adaptan bien al aprendizaje de secuencias: problemas donde el input, el output que queremos aproximar o ambos están formados por elementos dispuestos en un determinado orden que es relevante considerar.
- Las redes recurrentes permiten naturalmente procesar inputs de largo variable, producir outputs de largo variable y adaptar la complejidad del modelo a la complejidad de la tarea de manera elegante y parsimoniosa al costo de una secuencialidad inherente.

