INTRODUCCIÓN A NLP CON APRENDIZAJE PROFUNDO
Parte 2

Pablo Martínez Olmos, **pamartin@ing.uc3m.es**

Procesamiento de Lenguaje Natural

uc3m

Redes neuronales para secuencias

Procesamiento de Lenguaje Natural

uc3m

Motivación: predecir la siguiente palabra de un texto

"This morning I took my cat for a walk."

given these words

predict the next word

Motivación: predecir la siguiente palabra de un texto

Idea #1: use a fixed window

"This morning I took my cat for a walk."

given these two words — predict the next word

One-hot feature encoding: tells us what each word is

[1 0 0 0 0 0 1 0 0 0]

for — a

prediction

Motivación: predecir la siguiente palabra de un texto

Idea #1: use a fixed window

"This morning I took my cat for a walk."

given these two words — predict the next word

One-hot feature encoding: tells us what each word is

[1 0 0 0 0 0 1 0 0 0]

for — a

prediction

Problem #1: can't model long-term dependencies

"**France** is where I grew up, but I now live in Boston. I speak fluent ____."

Motivación: predecir la siguiente palabra de un texto

"This morning I took my cat for a walk."

given these words

predict the next word

Motivación: predecir la siguiente palabra de un texto

Idea #2: use entire sequence as set of counts

"This morning I took my cat for a"

"bag of words"

[0 1 0 0 1 0 0 ... 0 0 1 1 0 0 0 1]

prediction

Motivación: predecir la siguiente palabra de un texto

Idea #2: use entire sequence as set of counts

"This morning I took my cat for a"

"bag of words"

Problem #2: counts don't preserve order

The food was good, not bad at all.

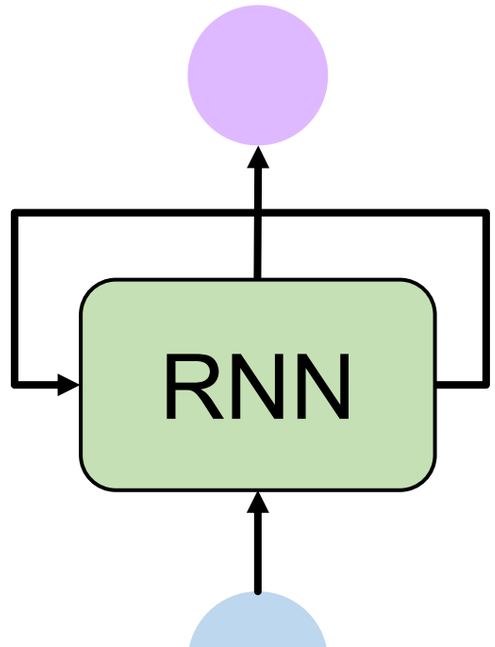
vs.

The food was bad, not good at all.

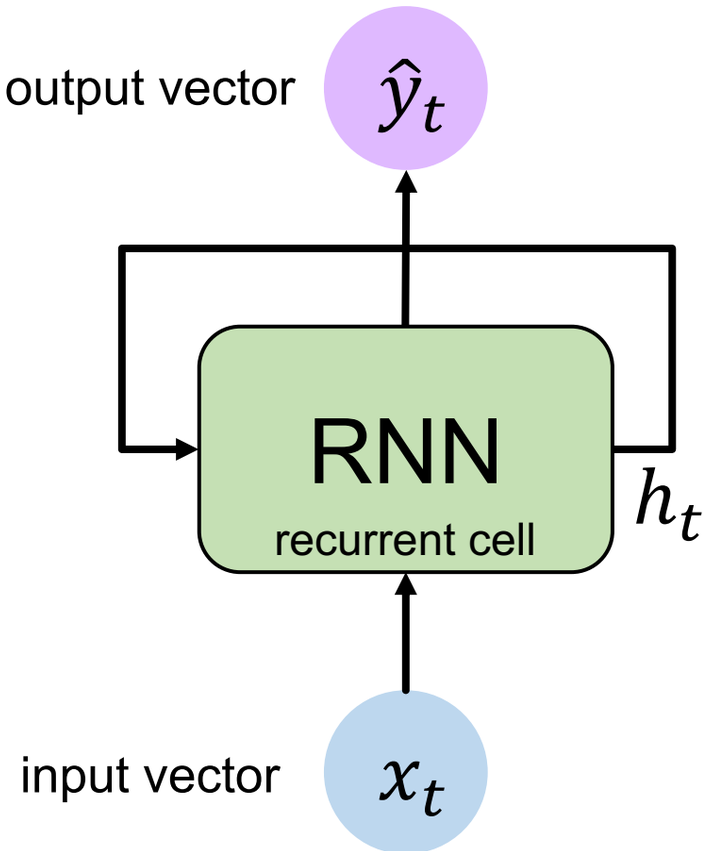
Redes neuronales recurrentes (RNNs)

Para modelar secuencias, debemos:

1. Tratar secuencias de **longitud variable**
2. Capturar **dependencias** temporales
3. Mantener información del **orden**



Una RNN básica



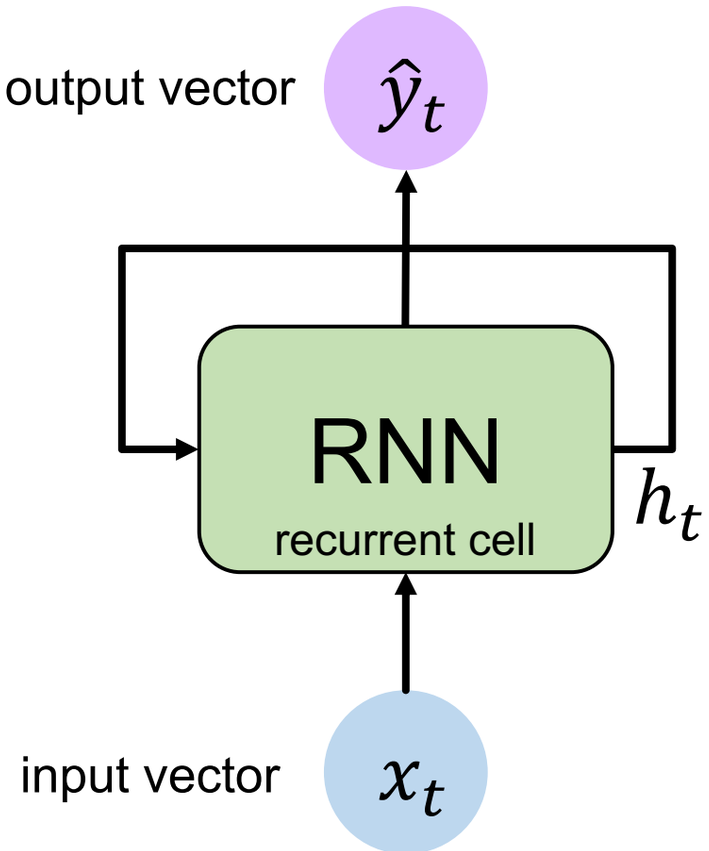
Misma **recurrencia** a lo largo del tiempo:

$$h_t = f_W\left(h_{t-1}, x_t\right)$$

La misma función y **parámetros se reutilizan** a lo largo del tiempo.

El **número de parámetros no crece con la longitud** de la secuencia.

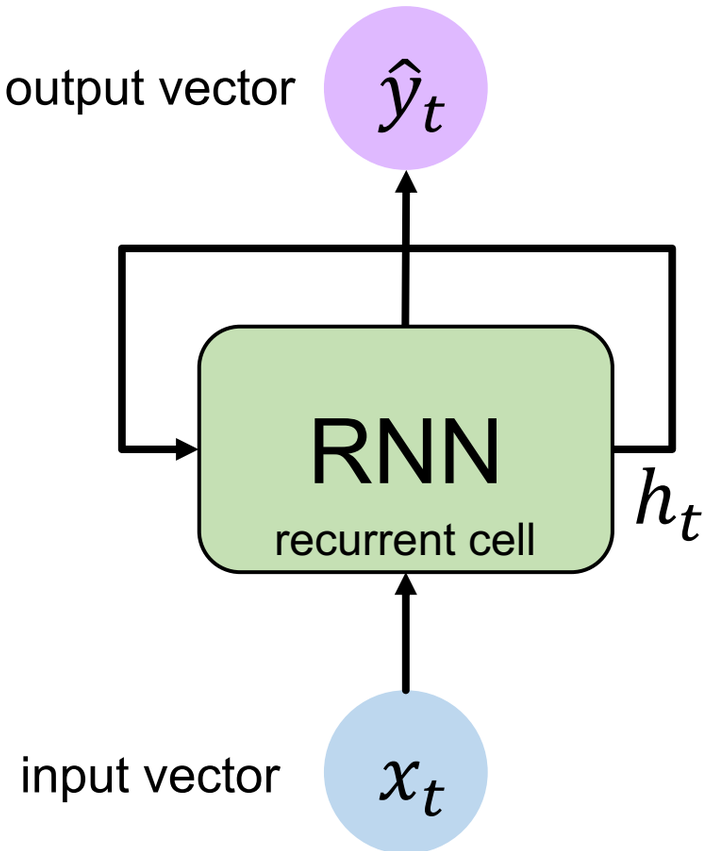
Una RNN básica



$$h_t = \tanh\left(\boldsymbol{W_{hh}} h_{t-1} + \boldsymbol{W_{xh}} x_t\right)$$

$$\hat{y}_t = \mathbf{W}_{hy} h_t$$

Una RNN básica

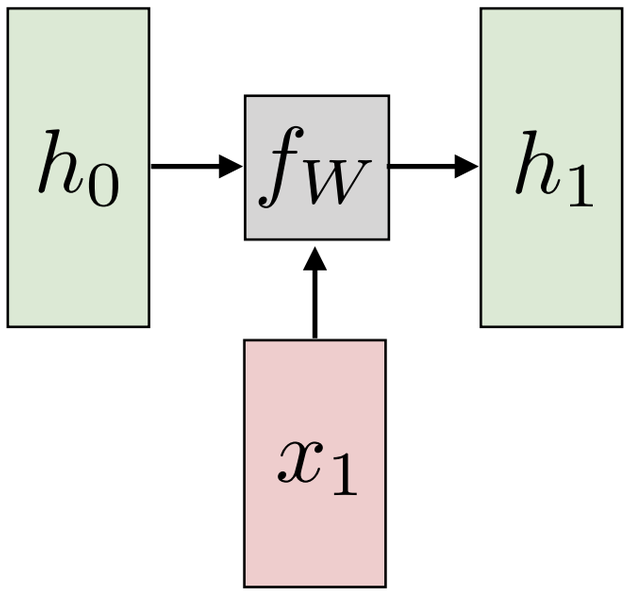


$$h_t = \tanh\left(\boldsymbol{W_{hh}} h_{t-1} + \boldsymbol{W_{xh}} x_t\right)$$

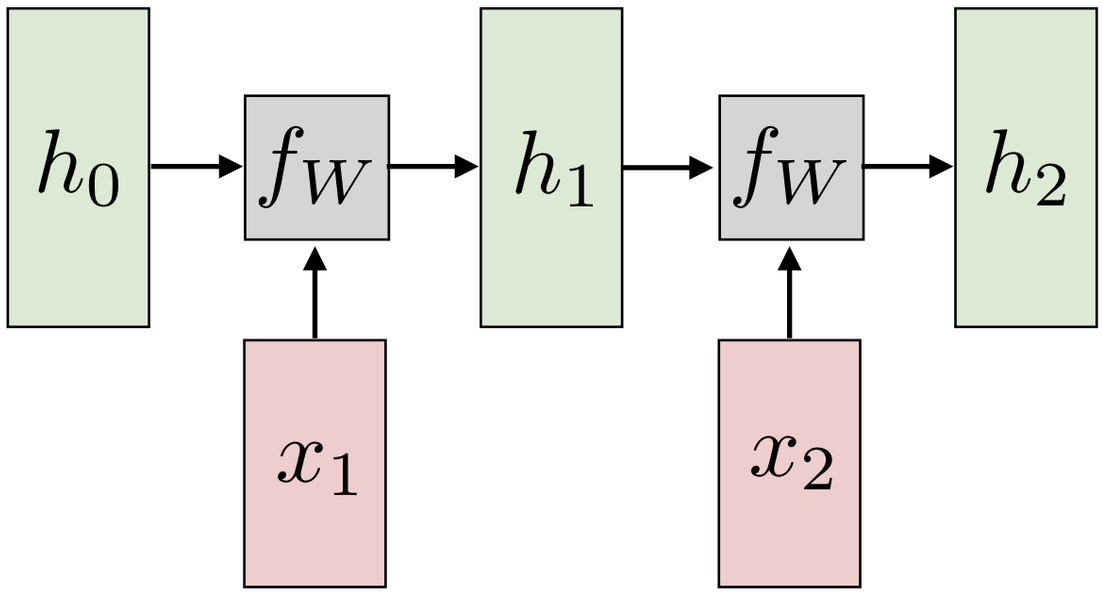
$$\hat{y}_t = \mathbf{W}_{hy} h_t$$

Combinar con una MLP para predicción final

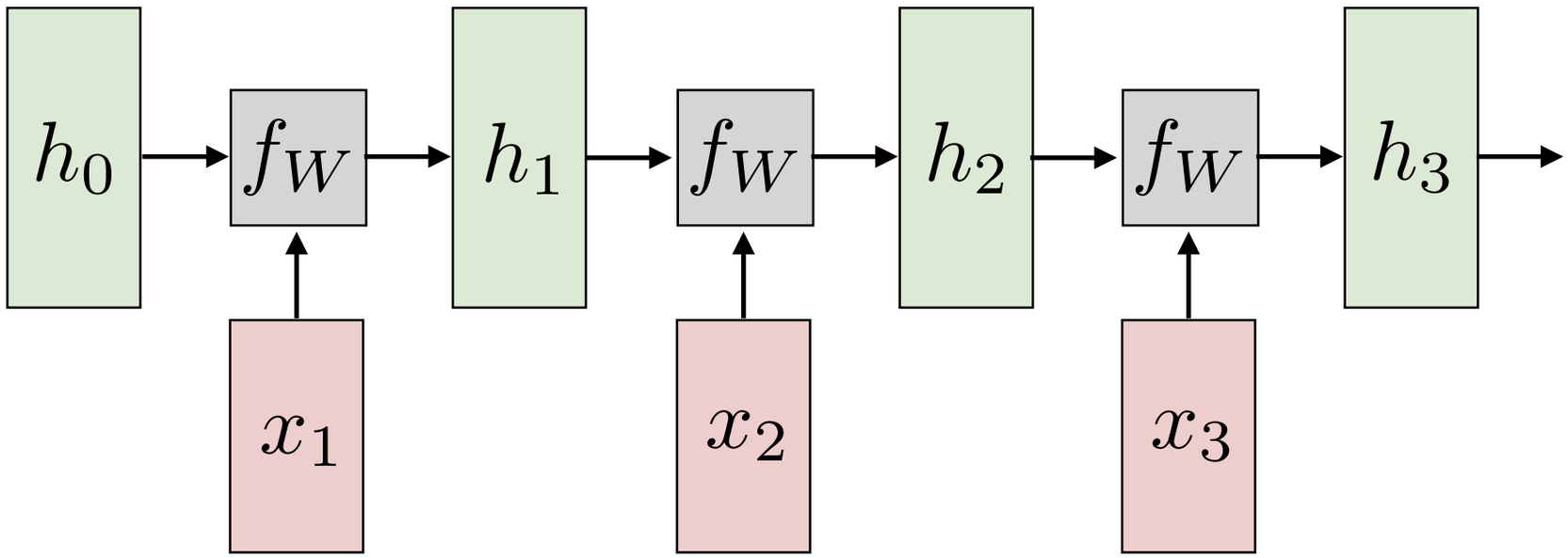
Una RNN básica



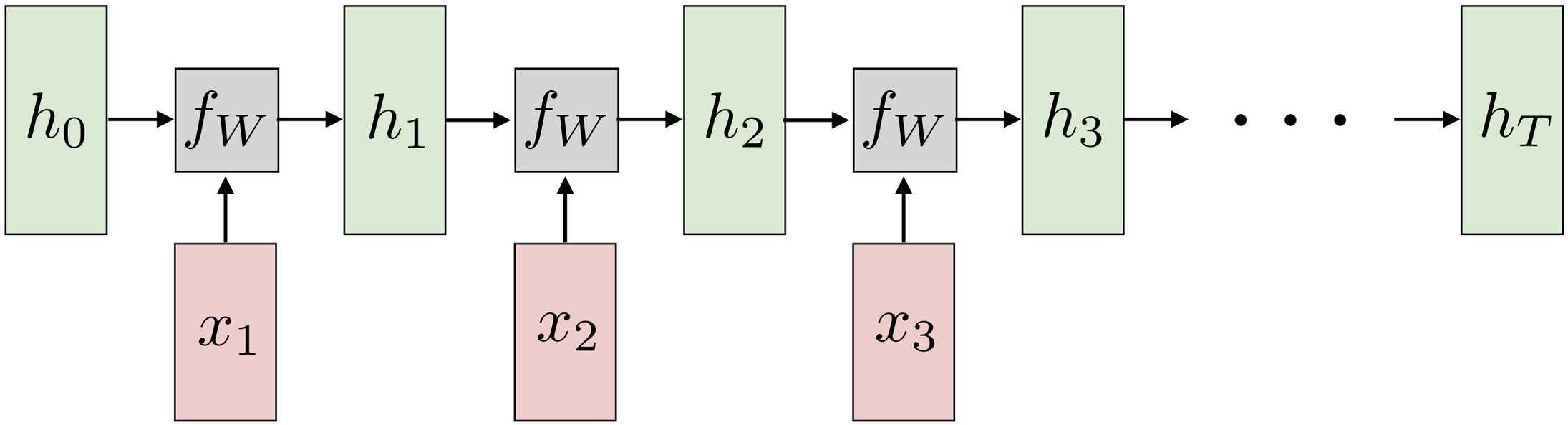
Una RNN básica



Una RNN básica

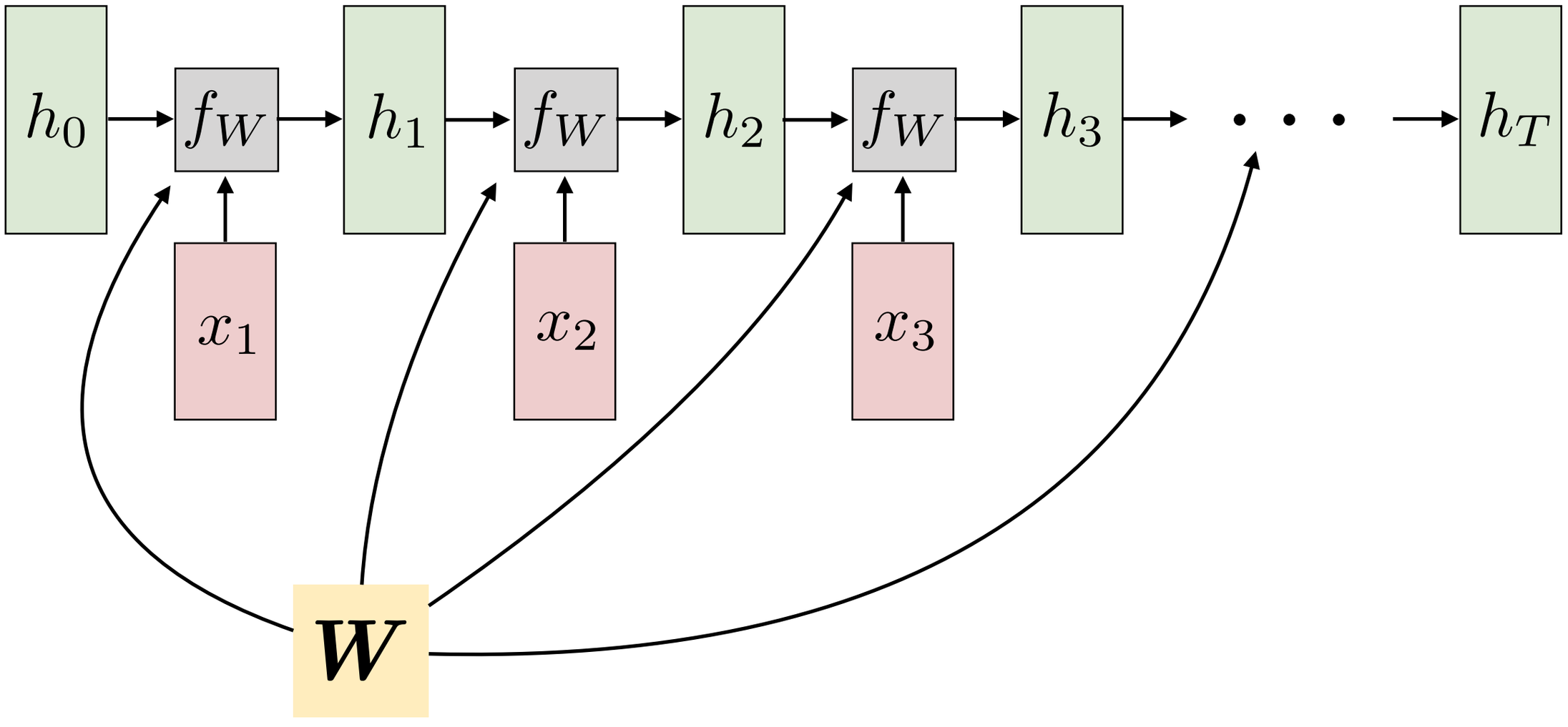


Una RNN básica



Una RNN básica

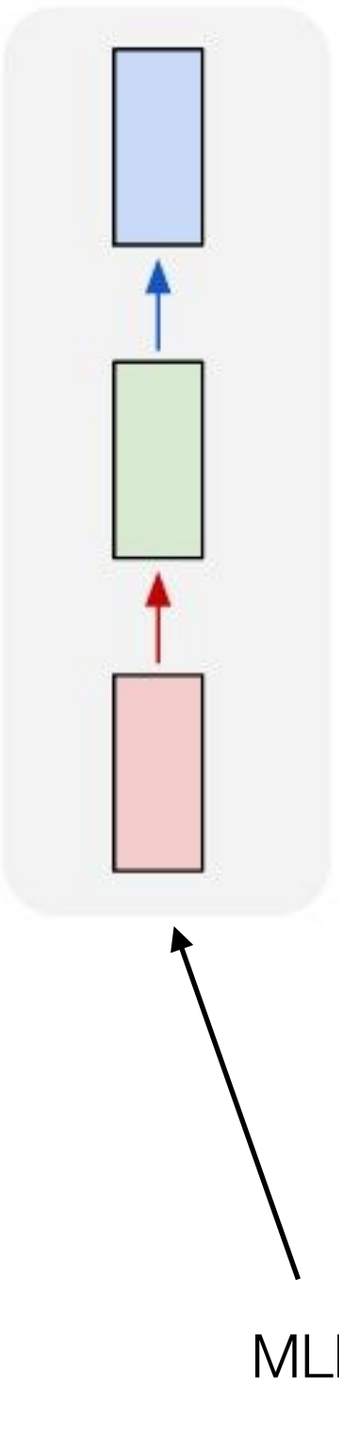
Reutilización de los mismos parámetros para procesar toda la secuencia



Variedad de estructuras utilizando RNNs

Variedad de estructuras utilizando RNNs

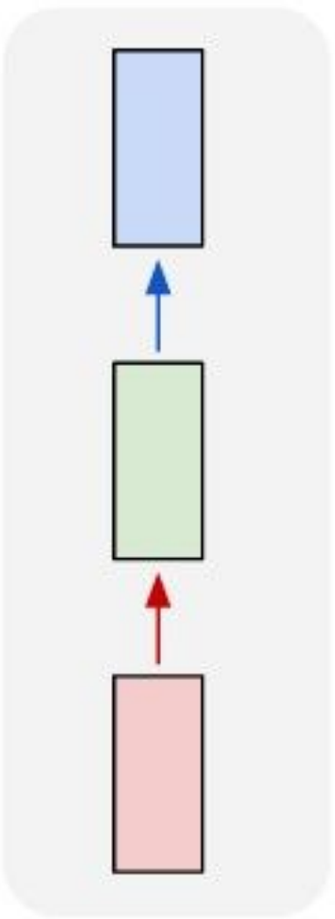
one to one



MLP

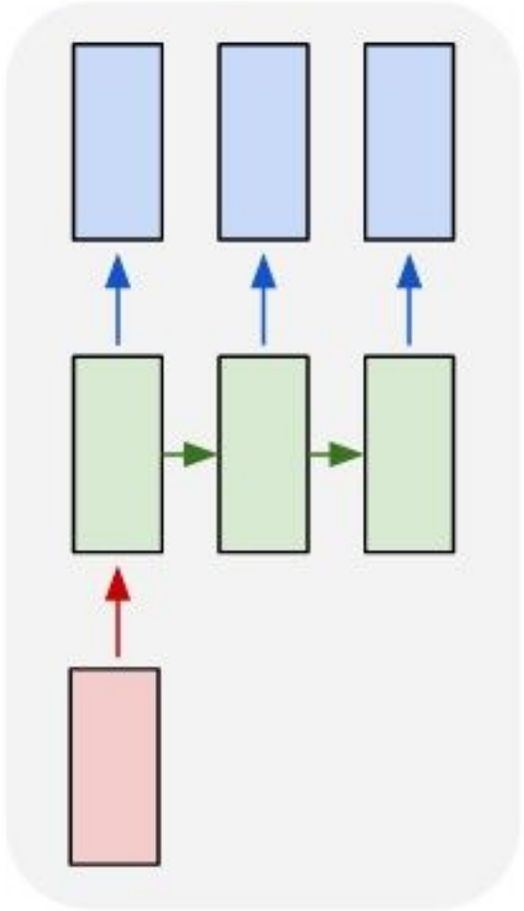
Variedad de estructuras utilizando RNNs

one to one



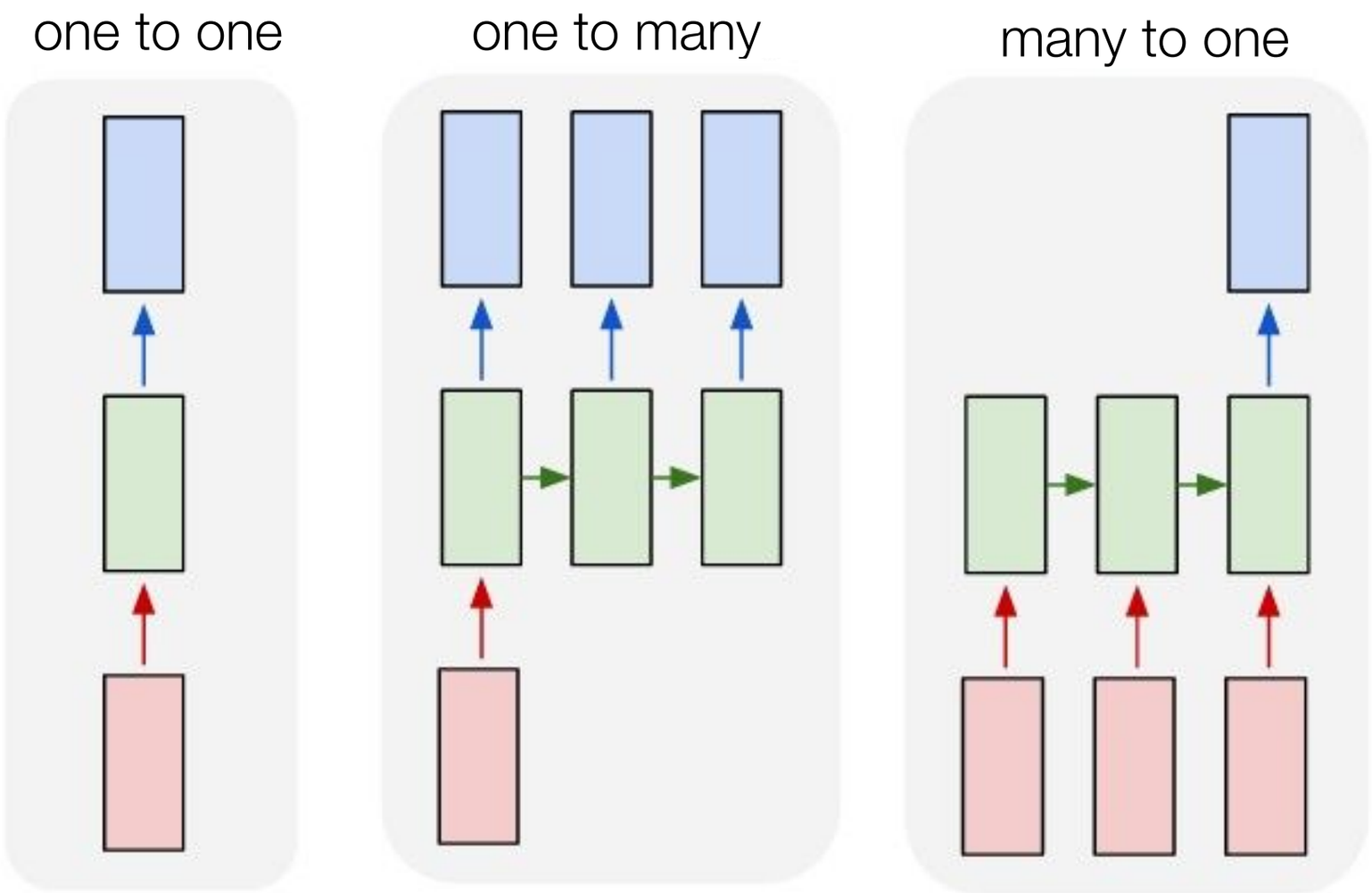
MLP

one to many



e.g. **Image Captioning**
image -> sequence of words

Variedad de estructuras utilizando RNNs

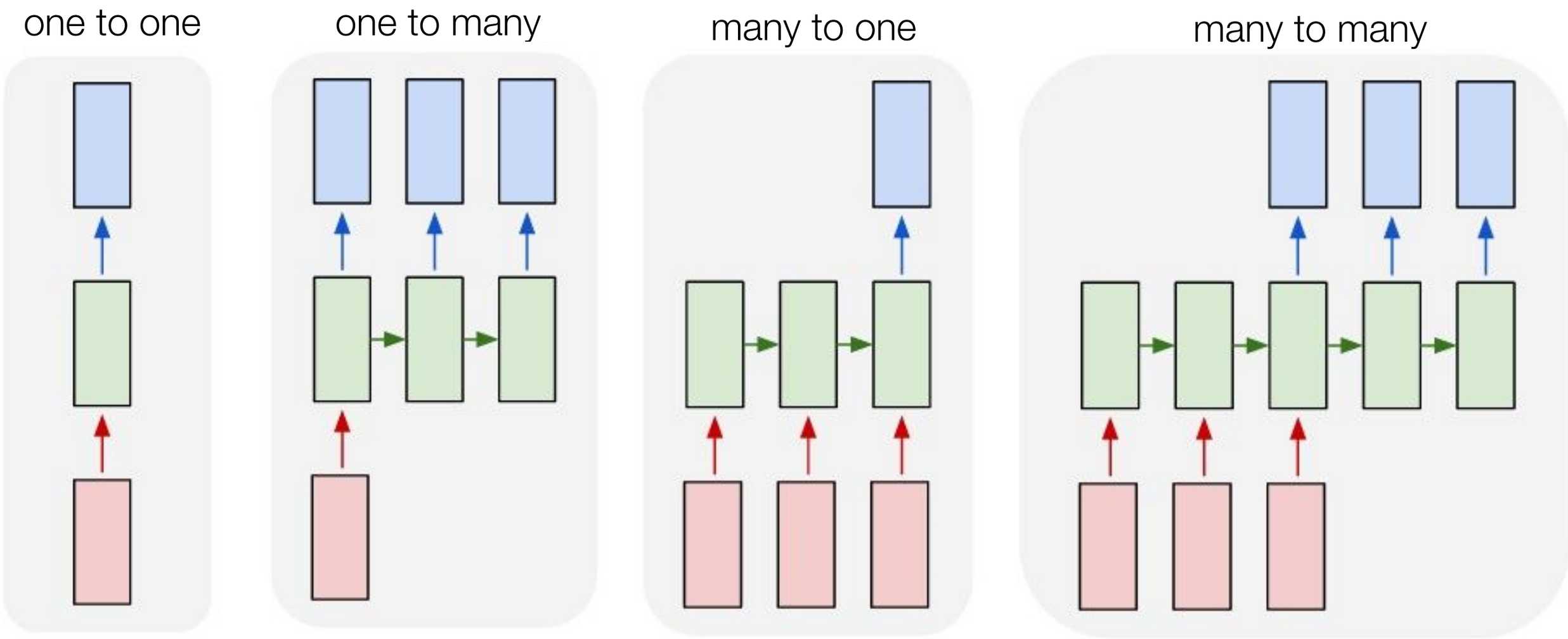


MLP

e.g. **Image Captioning**
image -> sequence of words

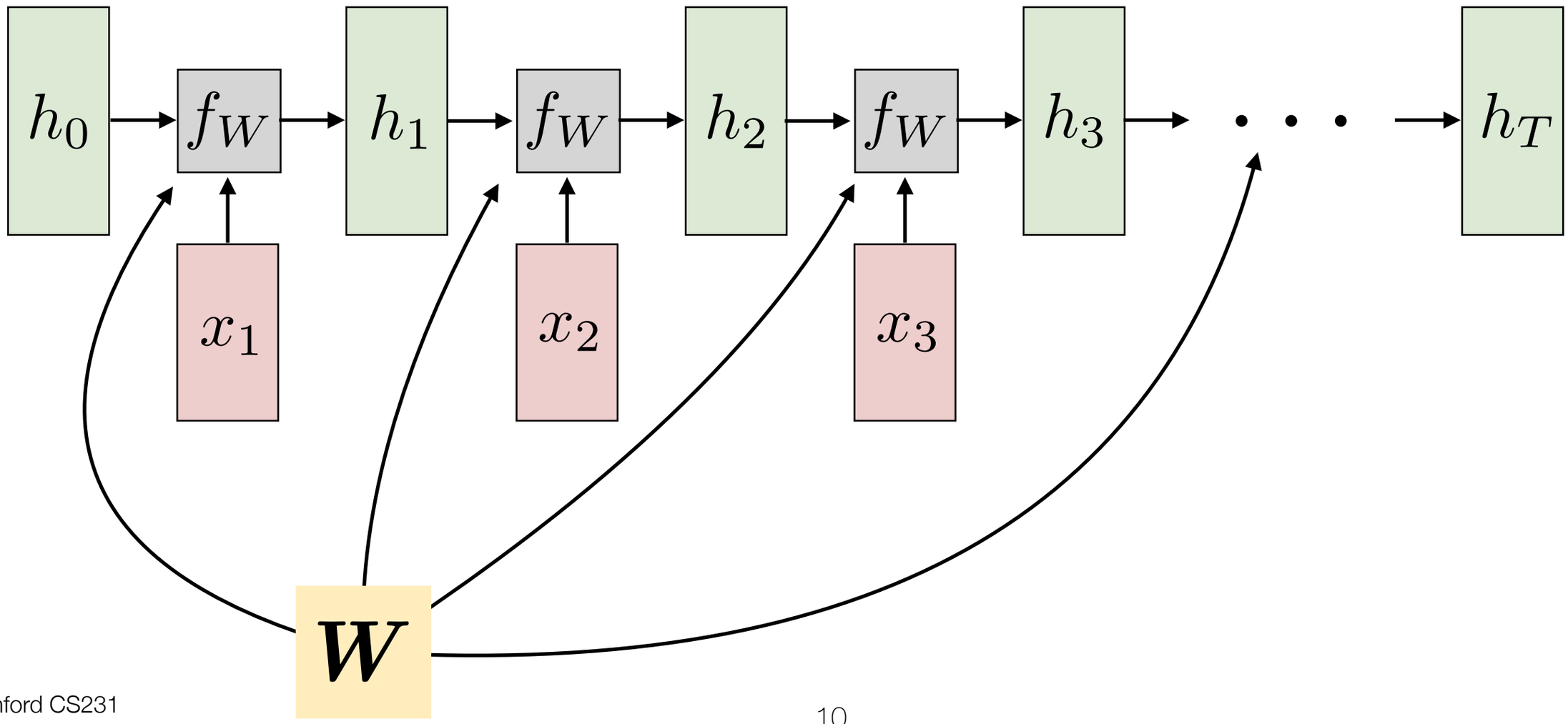
e.g. **Sentiment Classification**
sequence of words -> sentiment

Variedad de estructuras utilizando RNNs

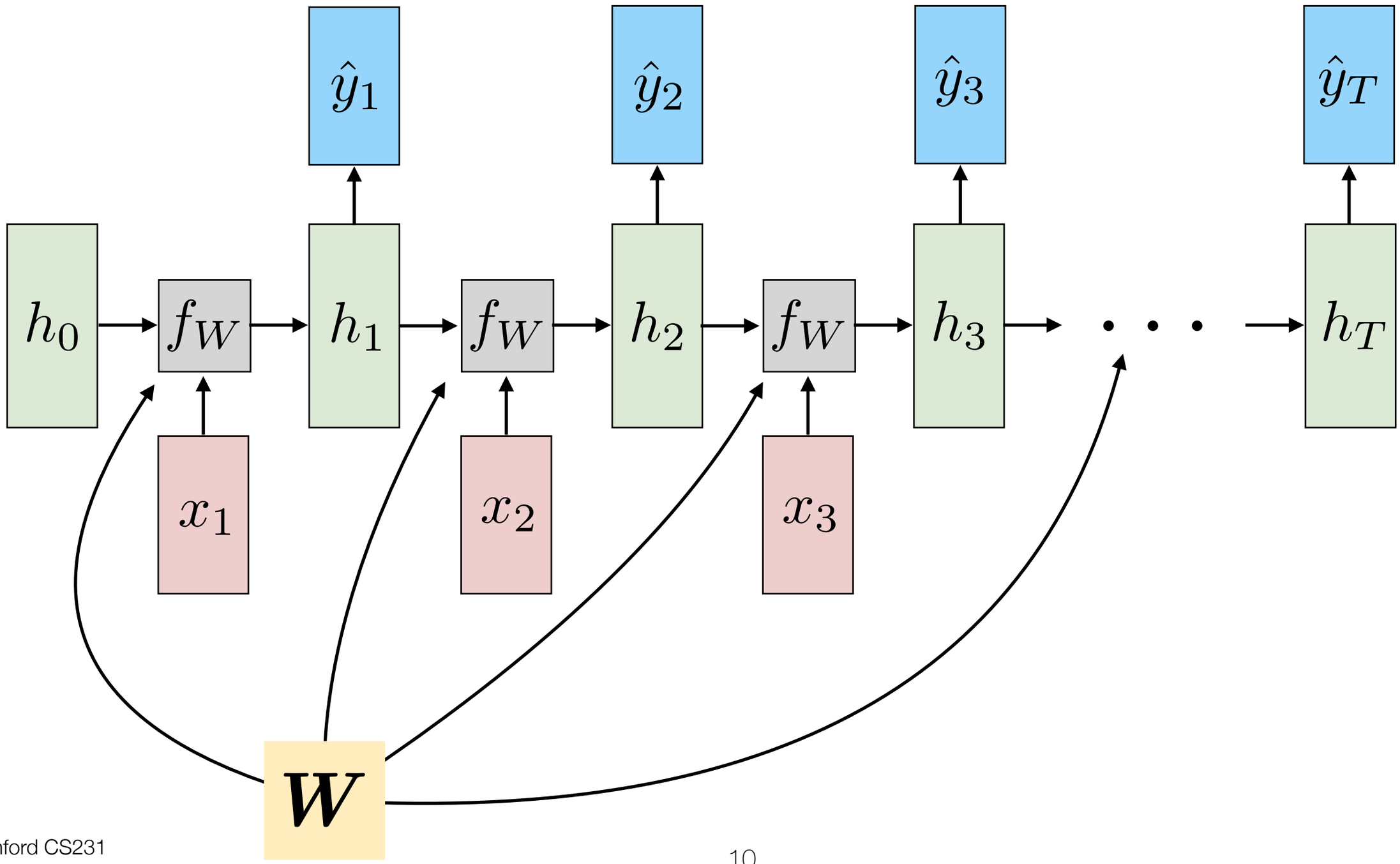


Múltiples predicciones a lo largo de la secuencia

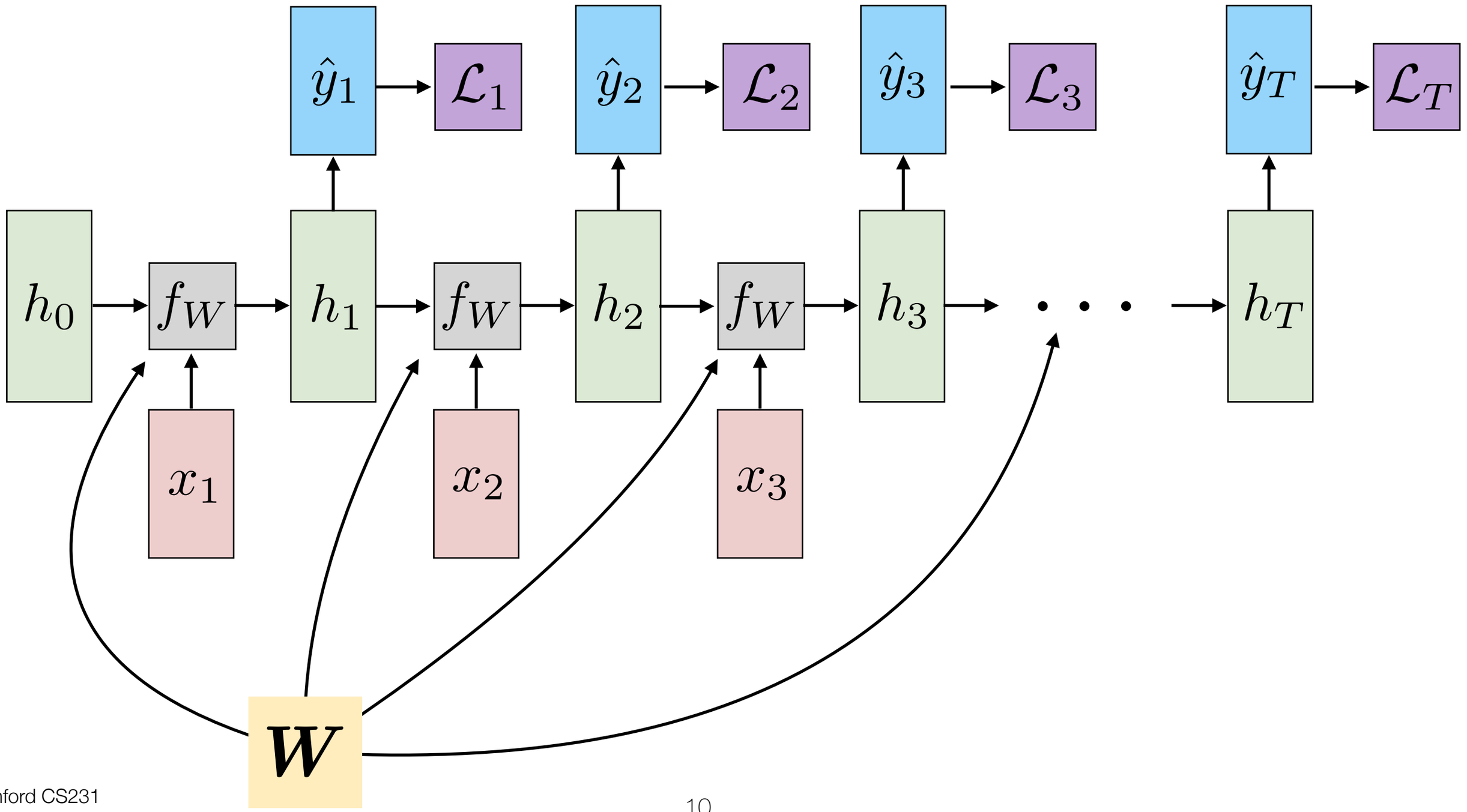
Múltiples predicciones a lo largo de la secuencia



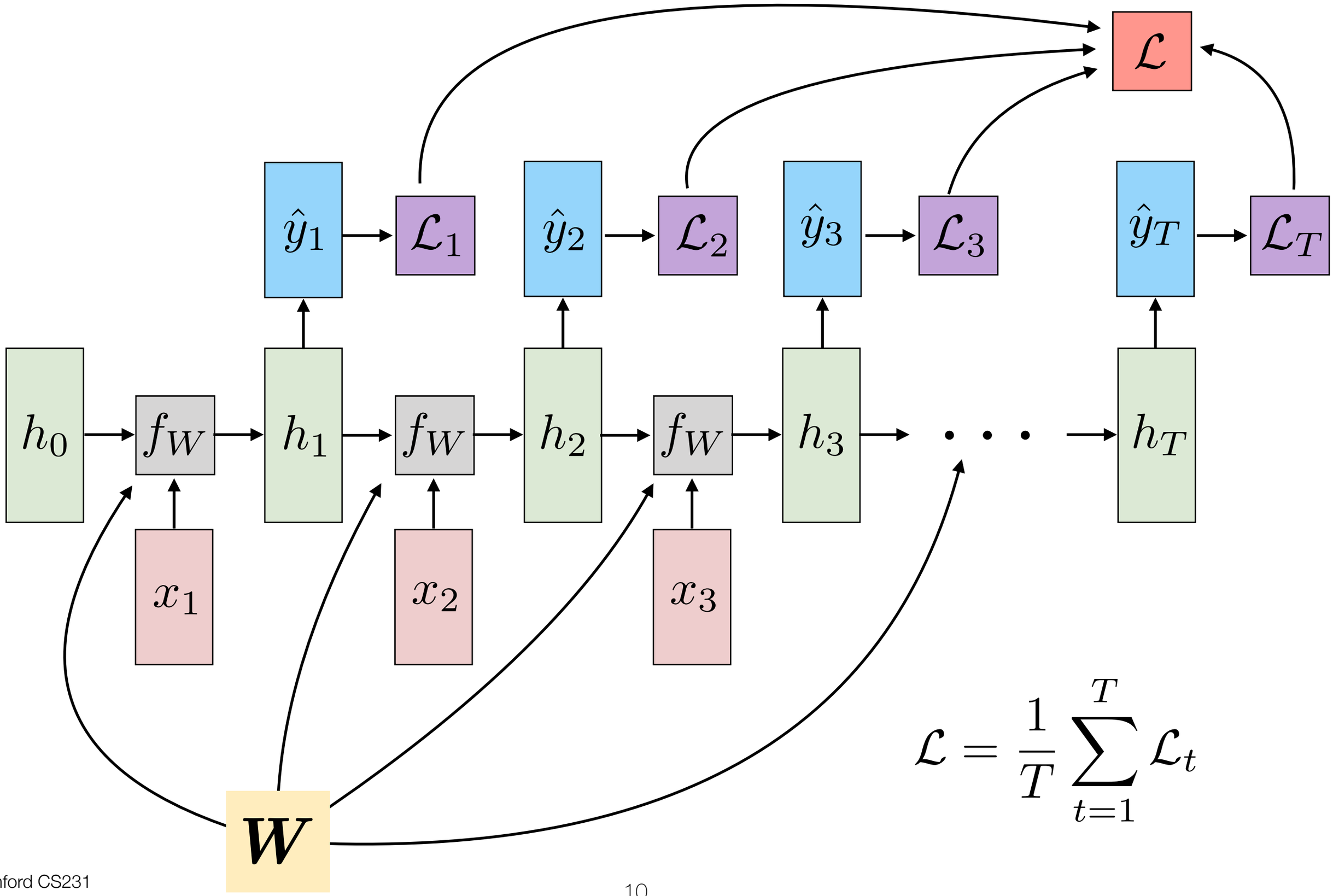
Múltiples predicciones a lo largo de la secuencia



Múltiples predicciones a lo largo de la secuencia



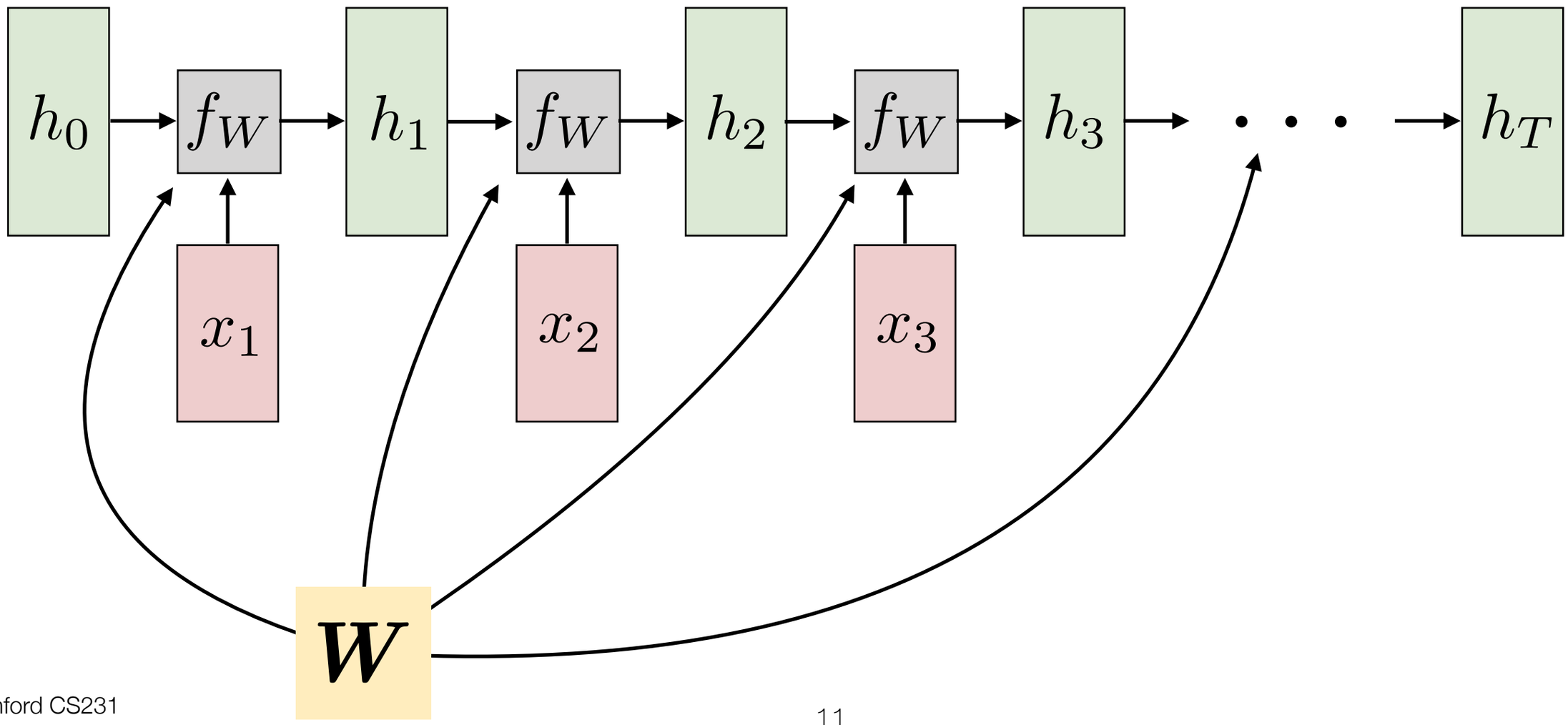
Múltiples predicciones a lo largo de la secuencia



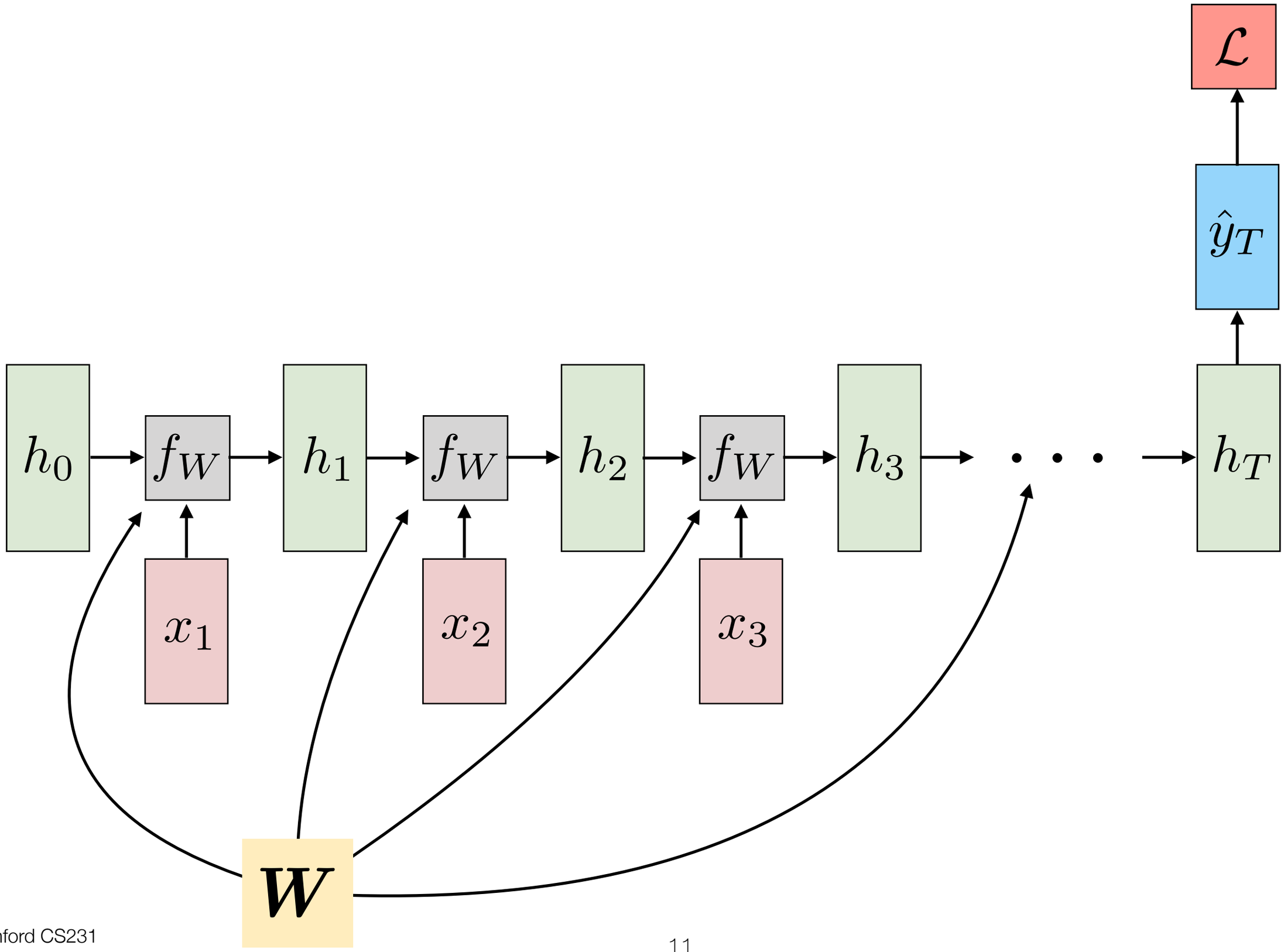
$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_t$$

Única predicción al final de la secuencia

Única predicción al final de la secuencia

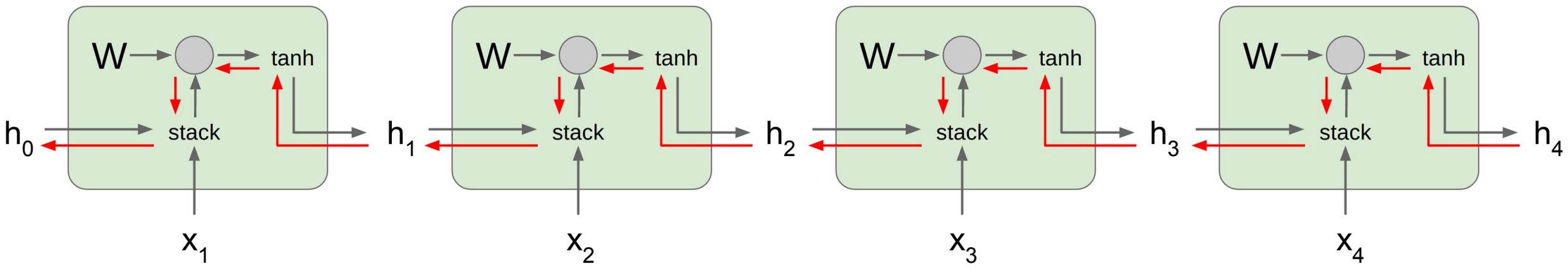


Única predicción al final de la secuencia



Limitaciones RNN básica

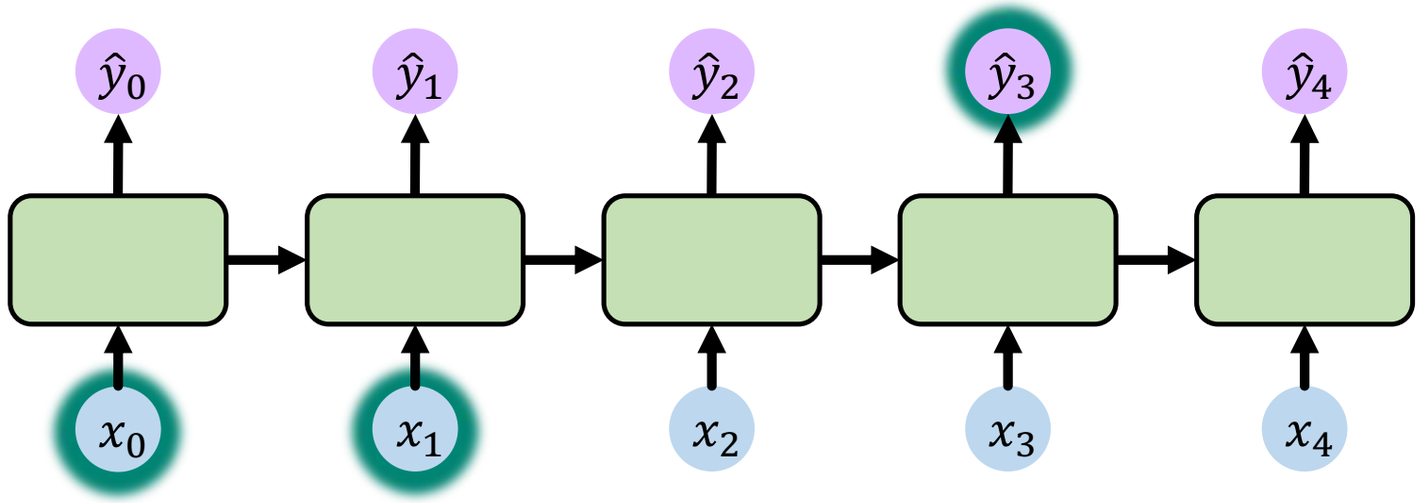
Un único modo de operación en el que cada nueva entrada modifica el estado



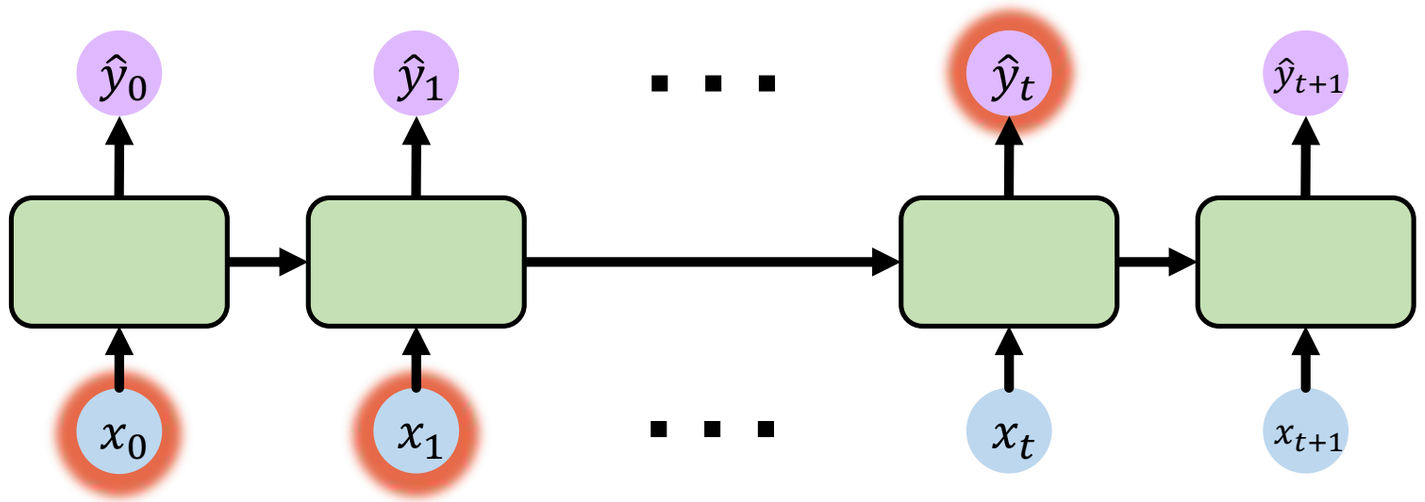
La red carece de mecanismos para recordar eventos pasados o ignorar observaciones presentes

Limitaciones RNN básica

"The clouds are in the ___"



"I grew up in France, … and I I speak fluent___ "

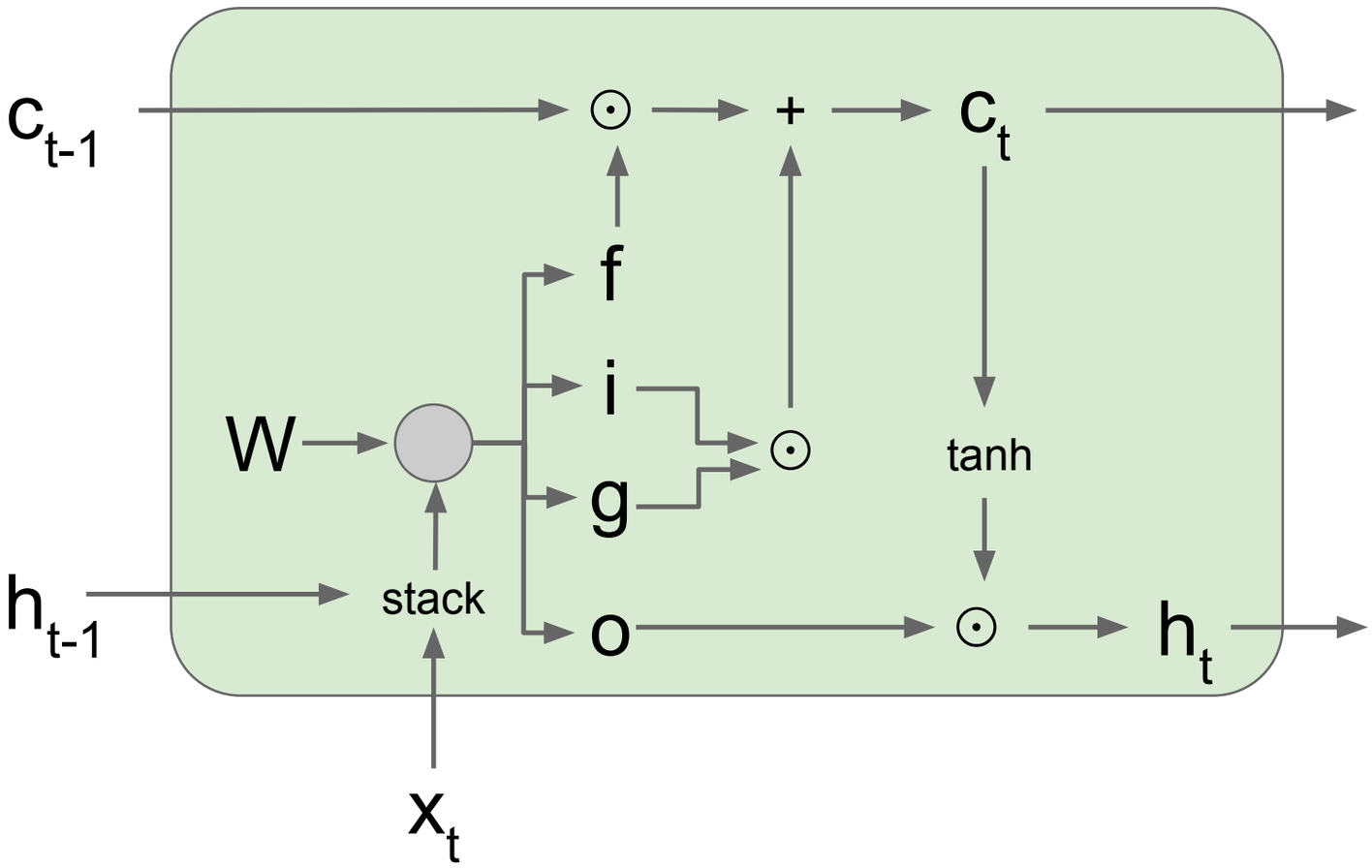


@Stanford CS231
http://cs231n.stanford.edu

Long Short Term Memory (LSTMs) networks

Hochreiter and Schmidhuber, Neural Computation 1997

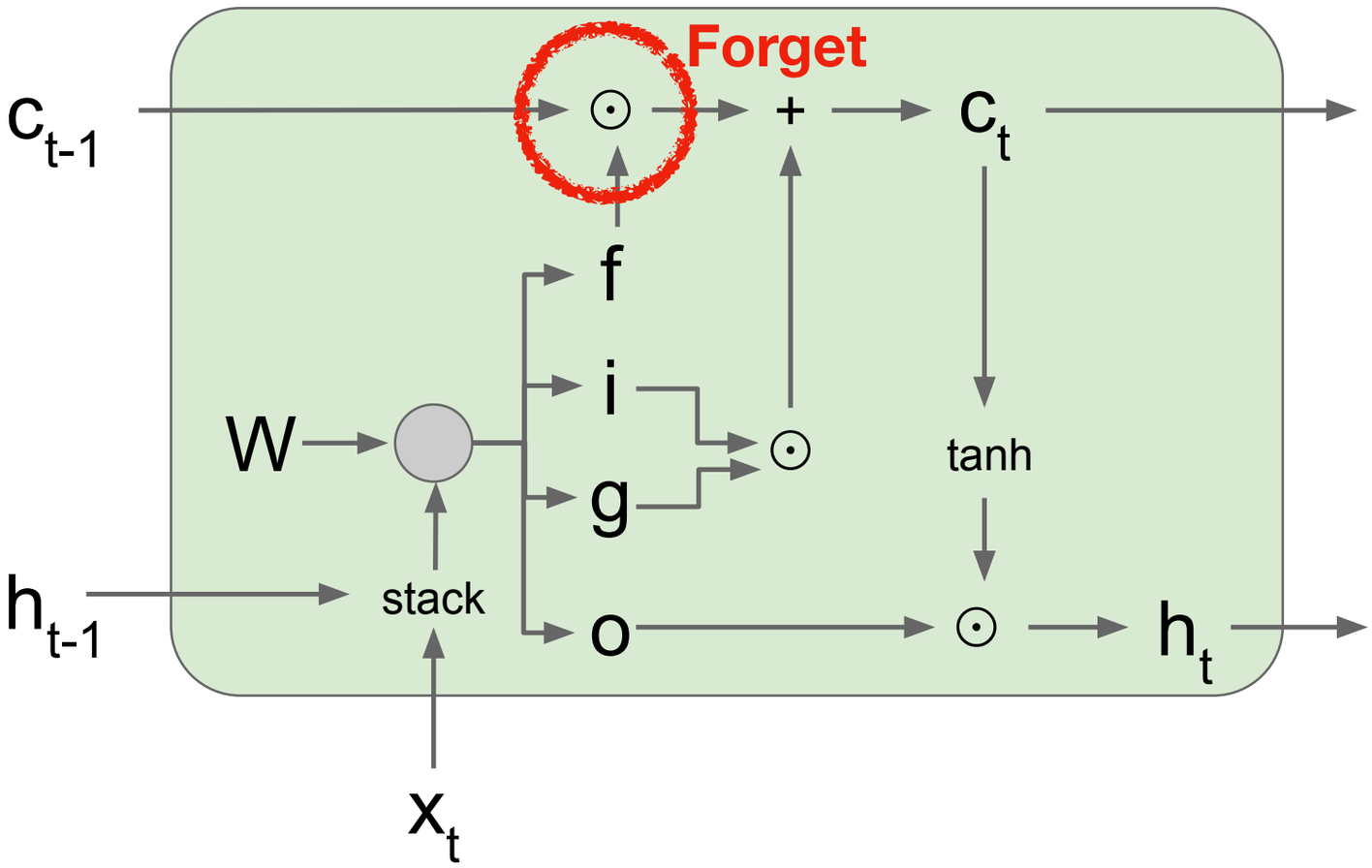
LSTMs: mecanismo de puertas



$$
\begin{aligned}
f &= \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + b_f\right) \\
i &= \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right) \\
o &= \sigma\left(W_{ox}x_t + W_{oh}h_{t-1} + b_o\right) \\
g &= \tanh\left(W_{cx}x_t + W_{ch}h_{t-1} + b_c\right) \\
c_t &= f \circ c_{t-1} + i \circ \tilde{c}_t \\
h_t &= o_t \circ \tanh\left(c_t\right)
\end{aligned}
$$

Las puertas controlan la propagación de la información a lo largo del tiempo

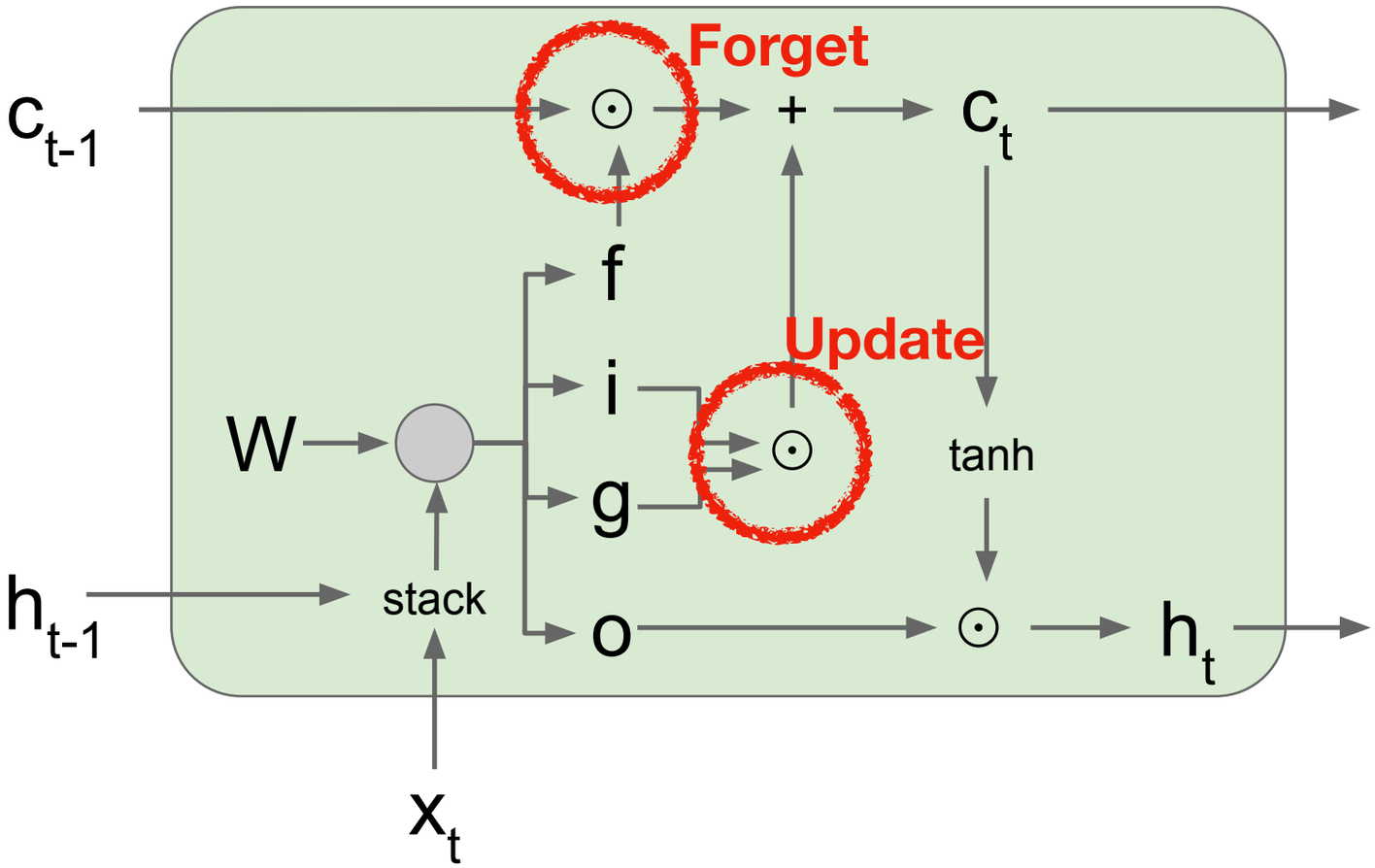
LSTMs: mecanismo de puertas



$$f = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + b_f\right)$$
$$i = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right)$$
$$o = \sigma\left(W_{ox}x_t + W_{oh}h_{t-1} + b_o\right)$$
$$g = \tanh\left(W_{cx}x_t + W_{ch}h_{t-1} + b_c\right)$$
$$c_t = f \circ c_{t-1} + i \circ \tilde{c}_t$$
$$h_t = o_t \circ \tanh\left(c_t\right)$$

Las puertas controlan la propagación de la información a lo largo del tiempo

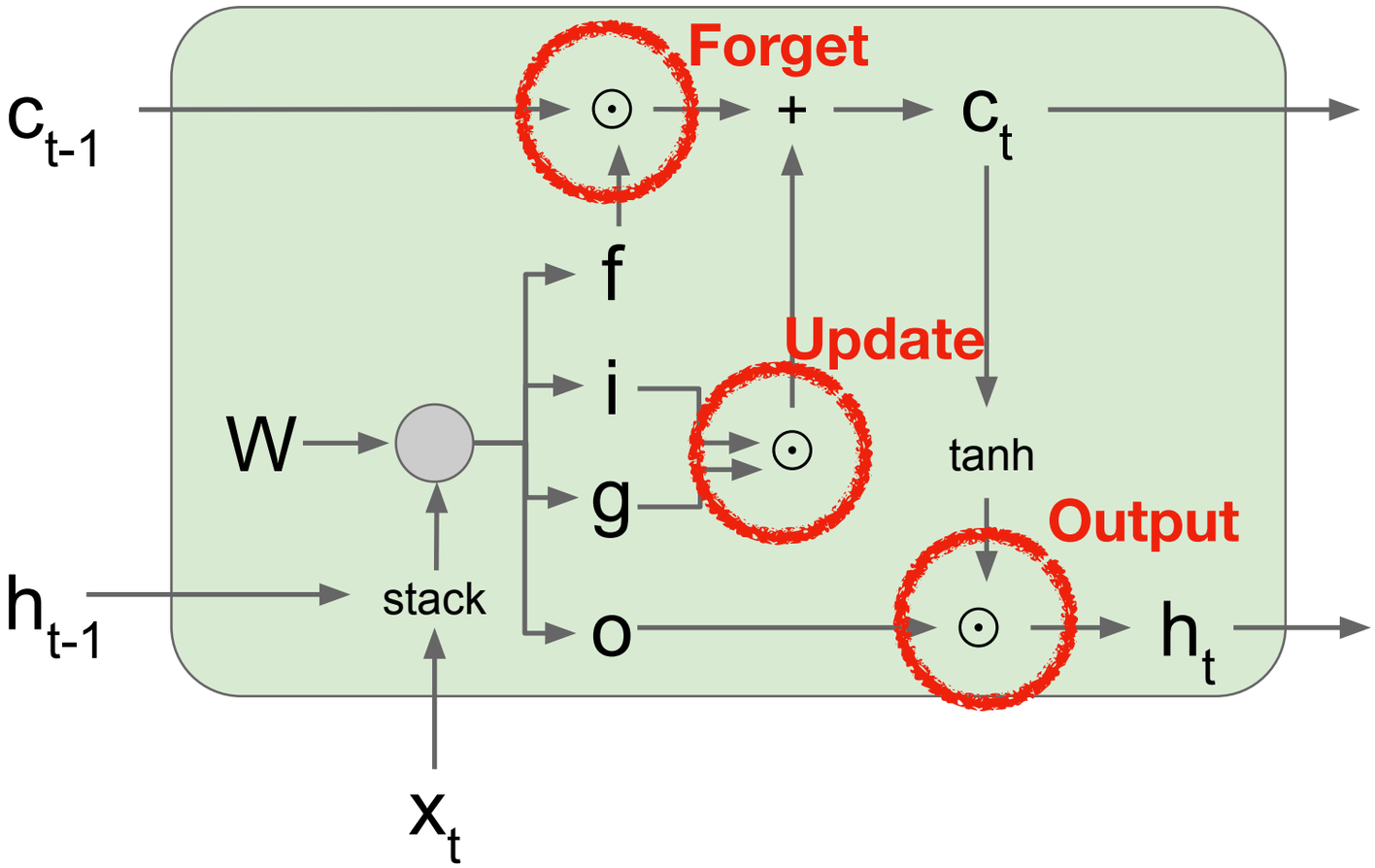
LSTMs: mecanismo de puertas



$$f = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + b_f\right)$$
$$i = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right)$$
$$o = \sigma\left(W_{ox}x_t + W_{oh}h_{t-1} + b_o\right)$$
$$g = \tanh\left(W_{cx}x_t + W_{ch}h_{t-1} + b_c\right)$$
$$c_t = f \circ c_{t-1} + i \circ \tilde{c}_t$$
$$h_t = o_t \circ \tanh\left(c_t\right)$$

Las puertas controlan la propagación de la información a lo largo del tiempo

LSTMs: mecanismo de puertas



$$
\begin{aligned}
f &= \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + b_f\right) \\
i &= \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right) \\
o &= \sigma\left(W_{ox}x_t + W_{oh}h_{t-1} + b_o\right) \\
g &= \tanh\left(W_{cx}x_t + W_{ch}h_{t-1} + b_c\right) \\
c_t &= f \circ c_{t-1} + i \circ \tilde{c}_t \\
h_t &= o_t \circ \tanh\left(c_t\right)
\end{aligned}
$$

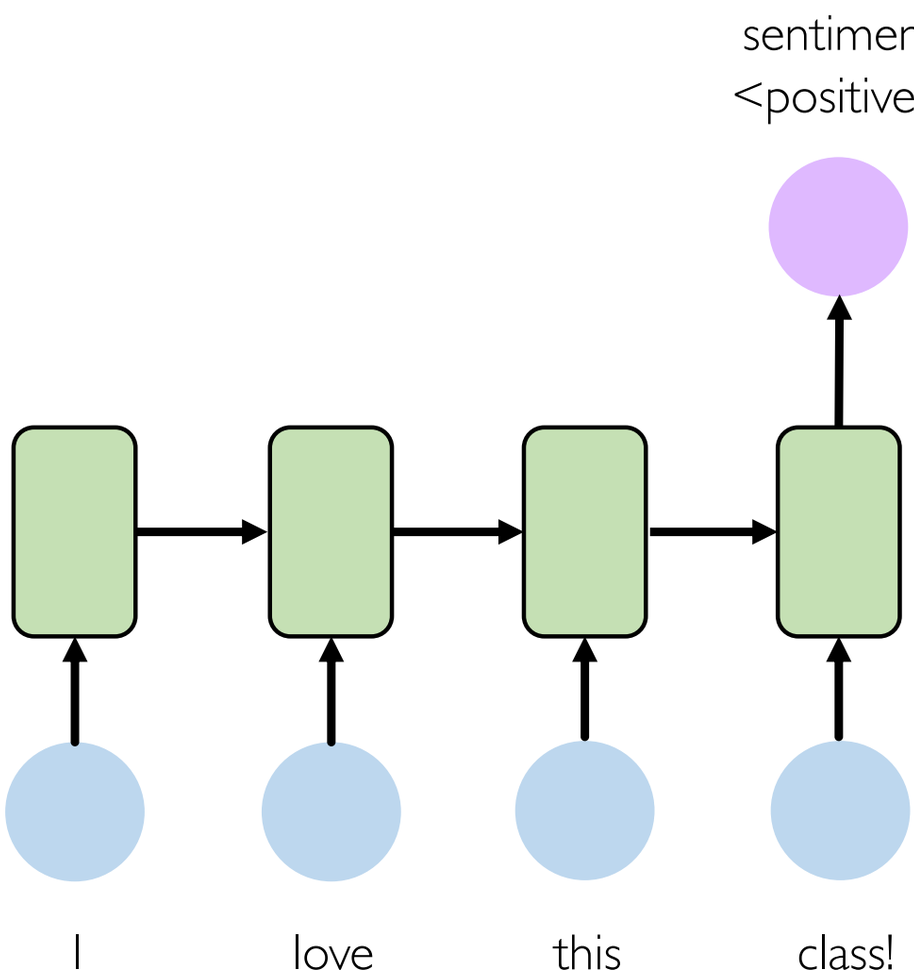
Las puertas controlan la propagación de la información a lo largo del tiempo

RNNs en NLP

Procesamiento de Lenguaje Natural

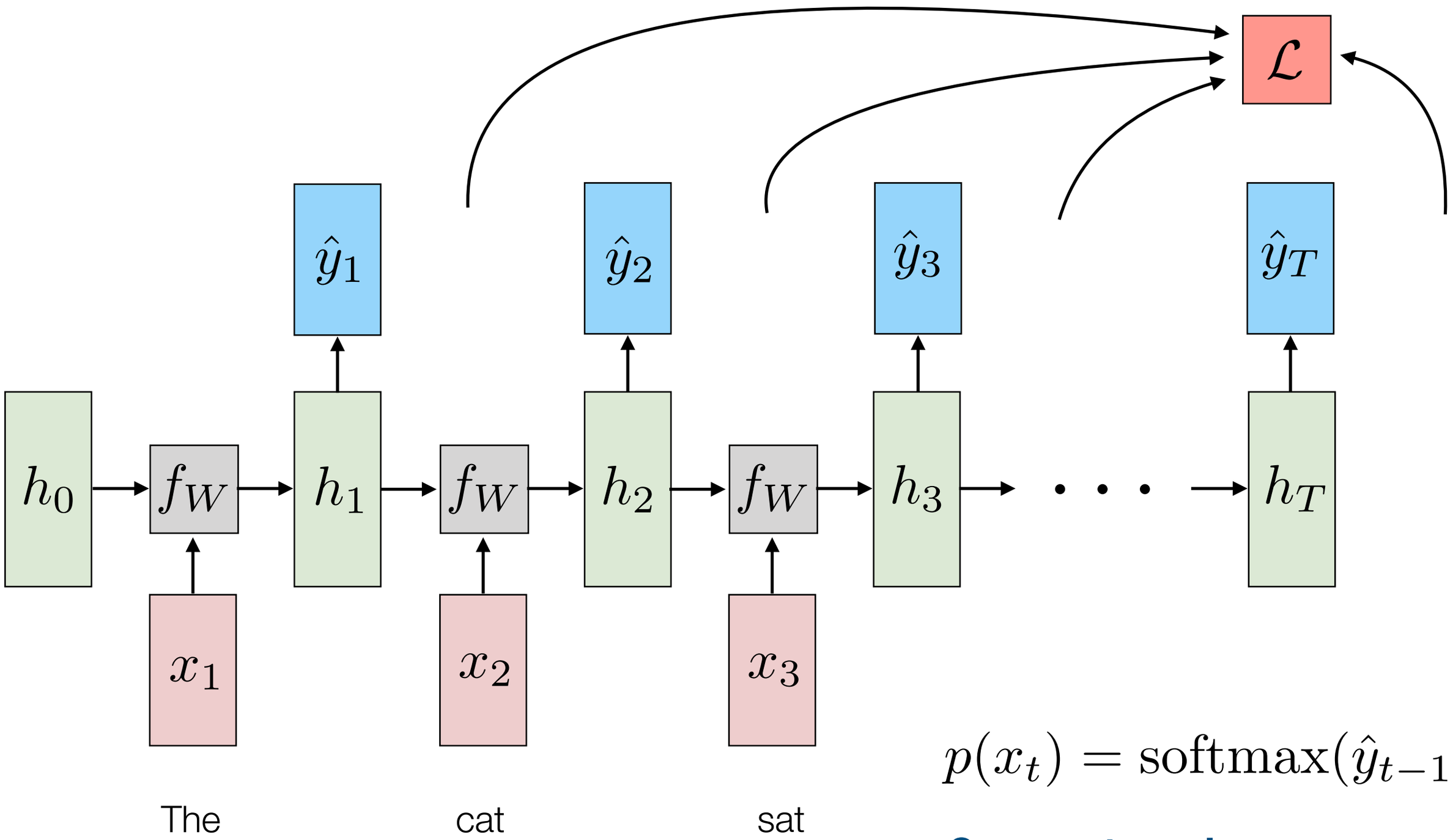
uc3m

Sentiment Analysis



Tweet sentiment classification

Generación de texto con RNNs (Entrenamiento)

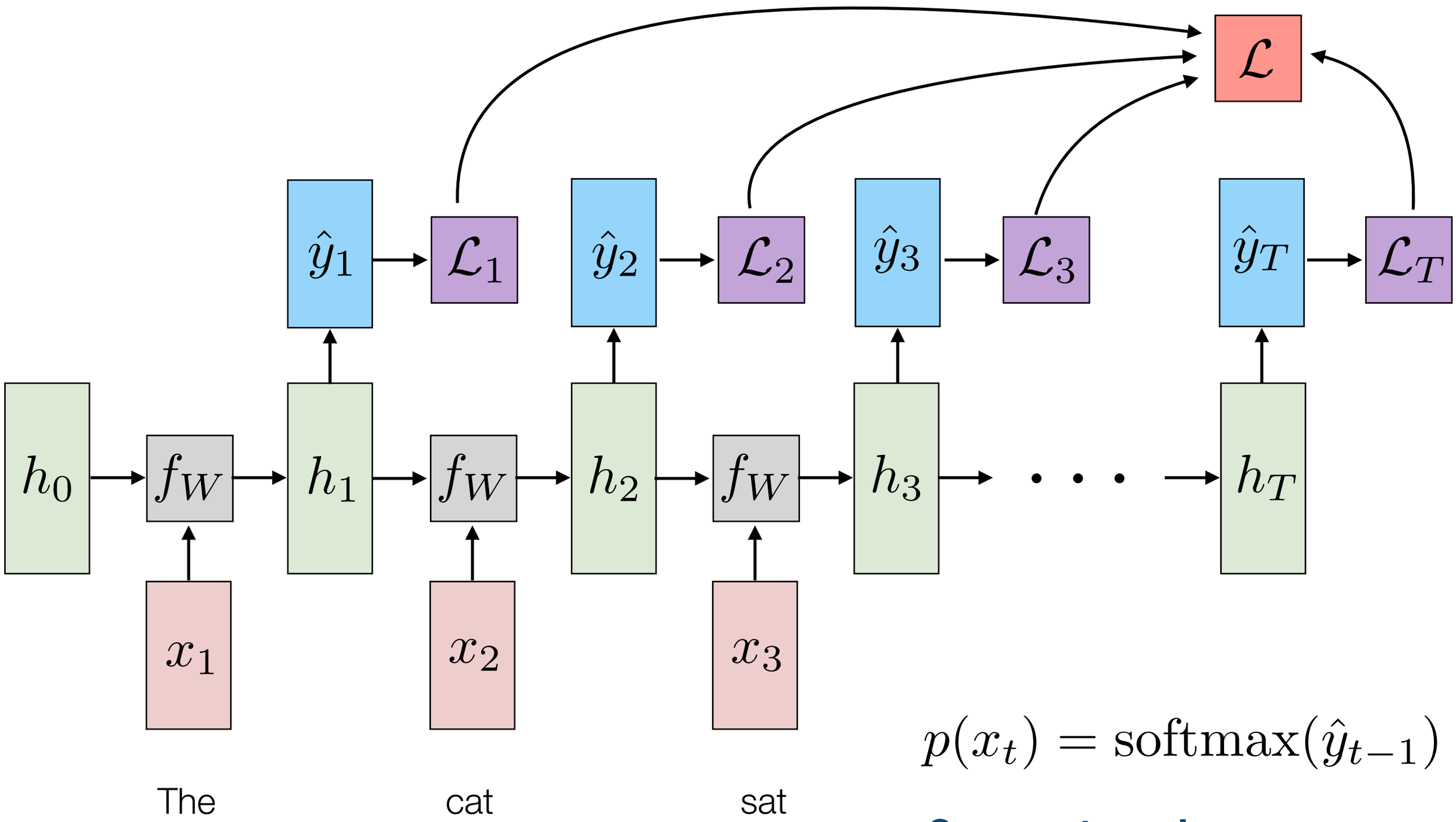


$$p(x_t) = \text{softmax}(\hat{y}_{t-1})$$

Cross entropy loss

Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. InProceedings of INTERSPEECH

Generación de texto con RNNs (Entrenamiento)



$$p(x_t) = \text{softmax}(\hat{y}_{t-1})$$

Cross entropy loss

Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. InProceedings of INTERSPEECH

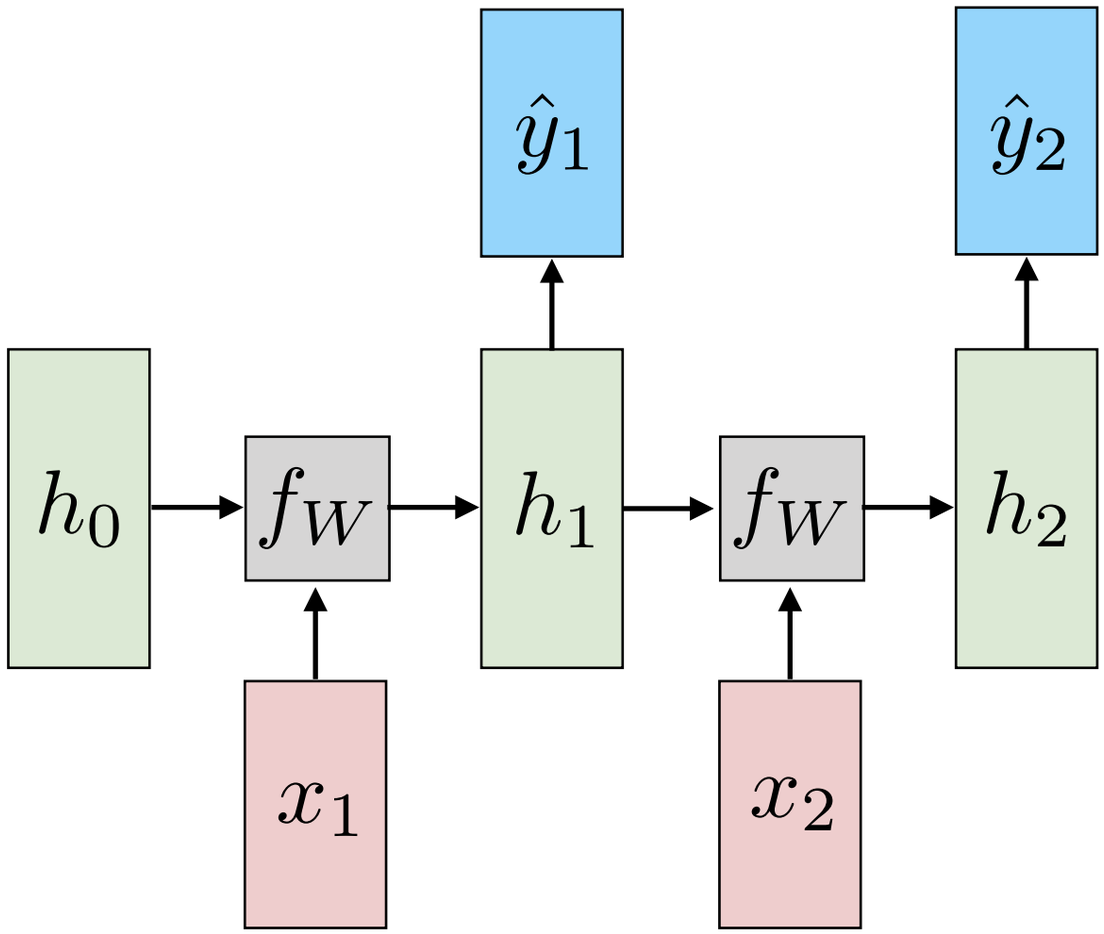
Generación de texto con RNNs (Test)

Muestreo secuencial …

Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. InProceedings of INTERSPEECH

Generación de texto con RNNs (Test)

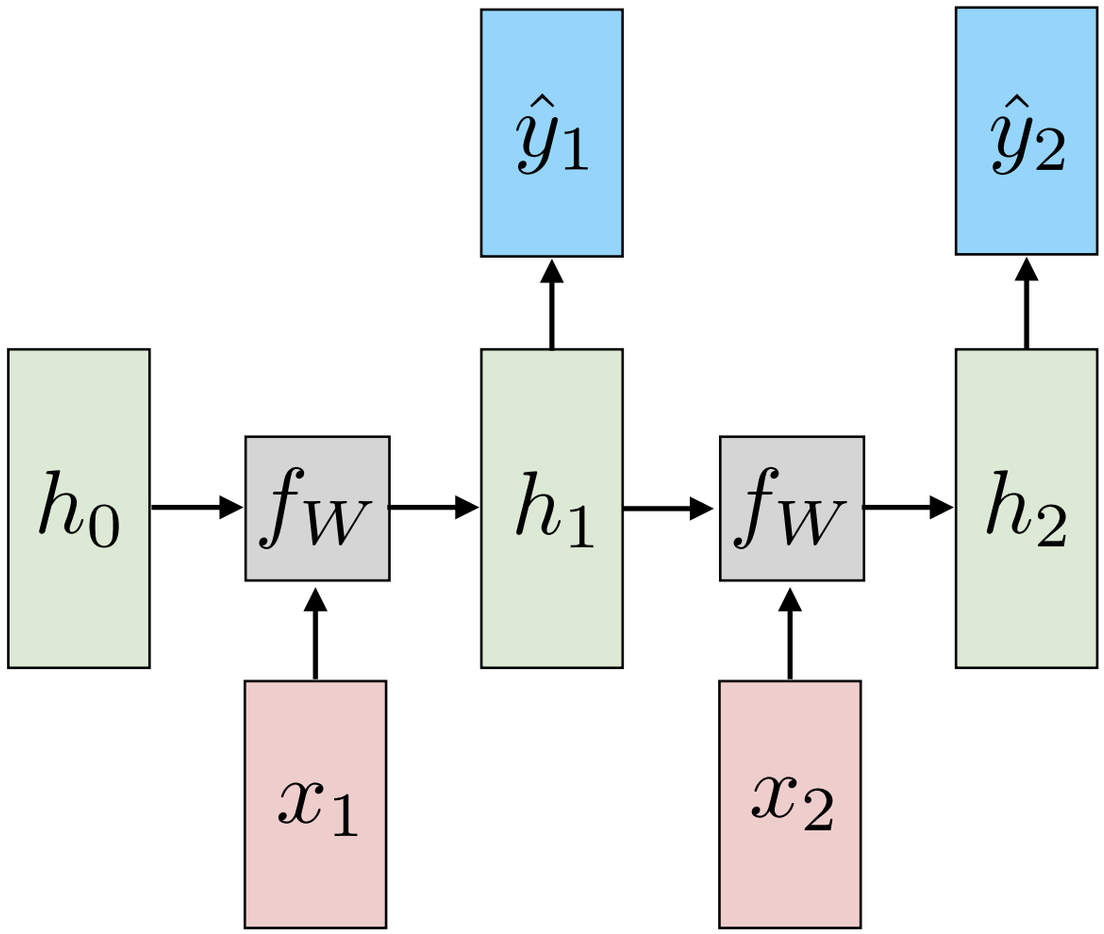
Muestreo secuencial …



Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. InProceedings of INTERSPEECH

Generación de texto con RNNs (Test)

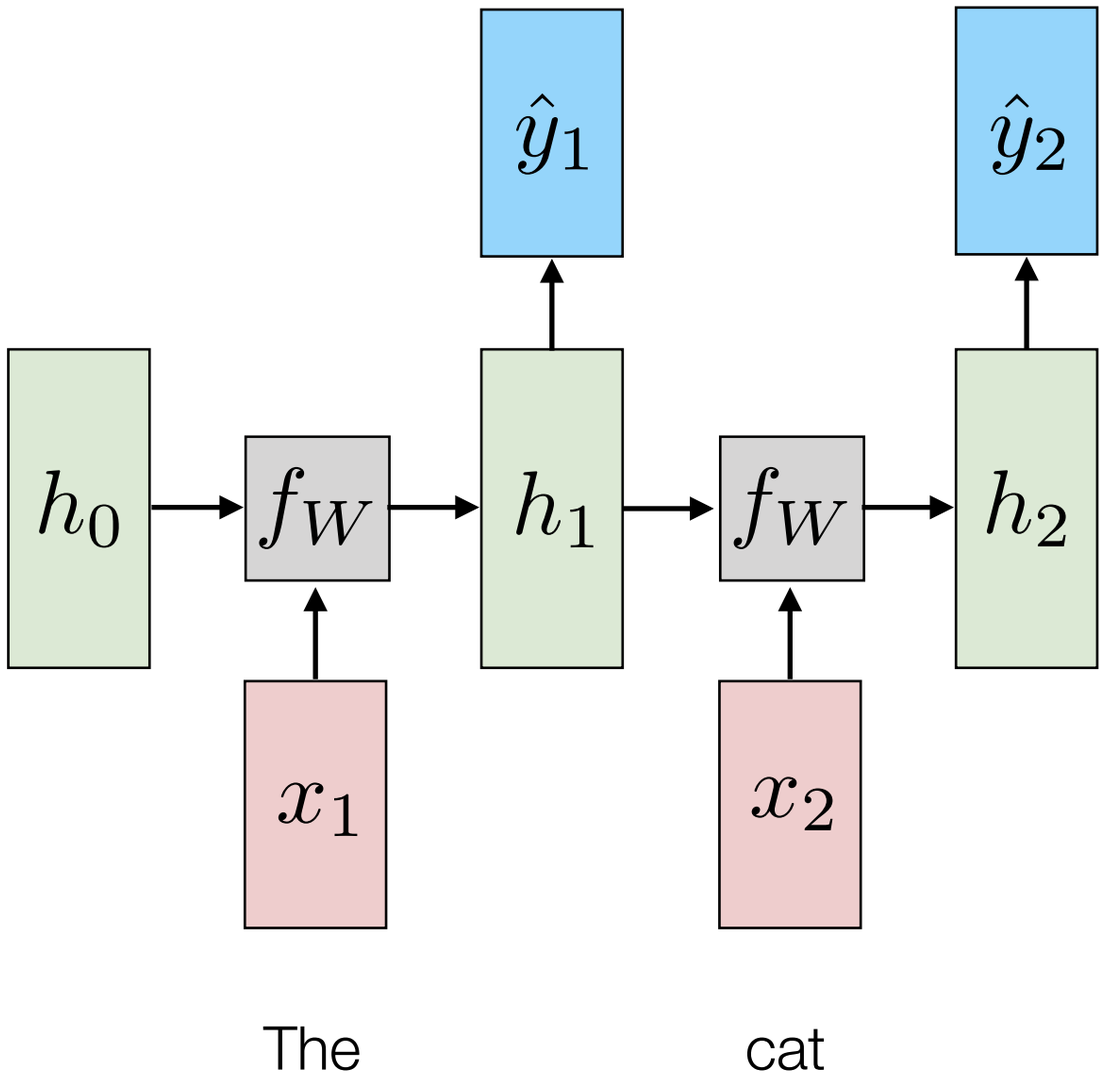
Muestreo secuencial …



Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. InProceedings of INTERSPEECH

Generación de texto con RNNs (Test)

Muestreo secuencial …

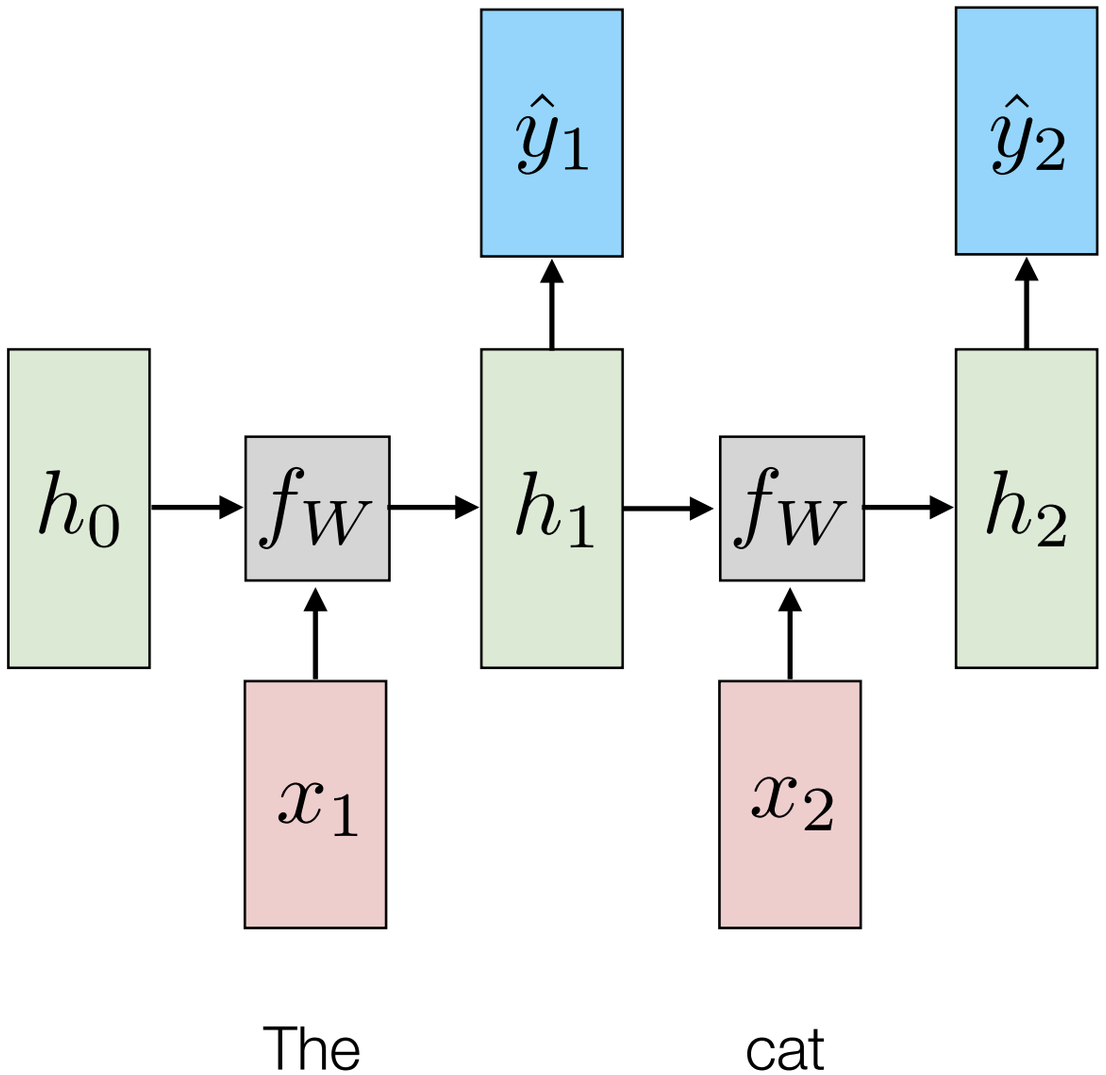


Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. InProceedings of INTERSPEECH

Generación de texto con RNNs (Test)

Muestreo secuencial …

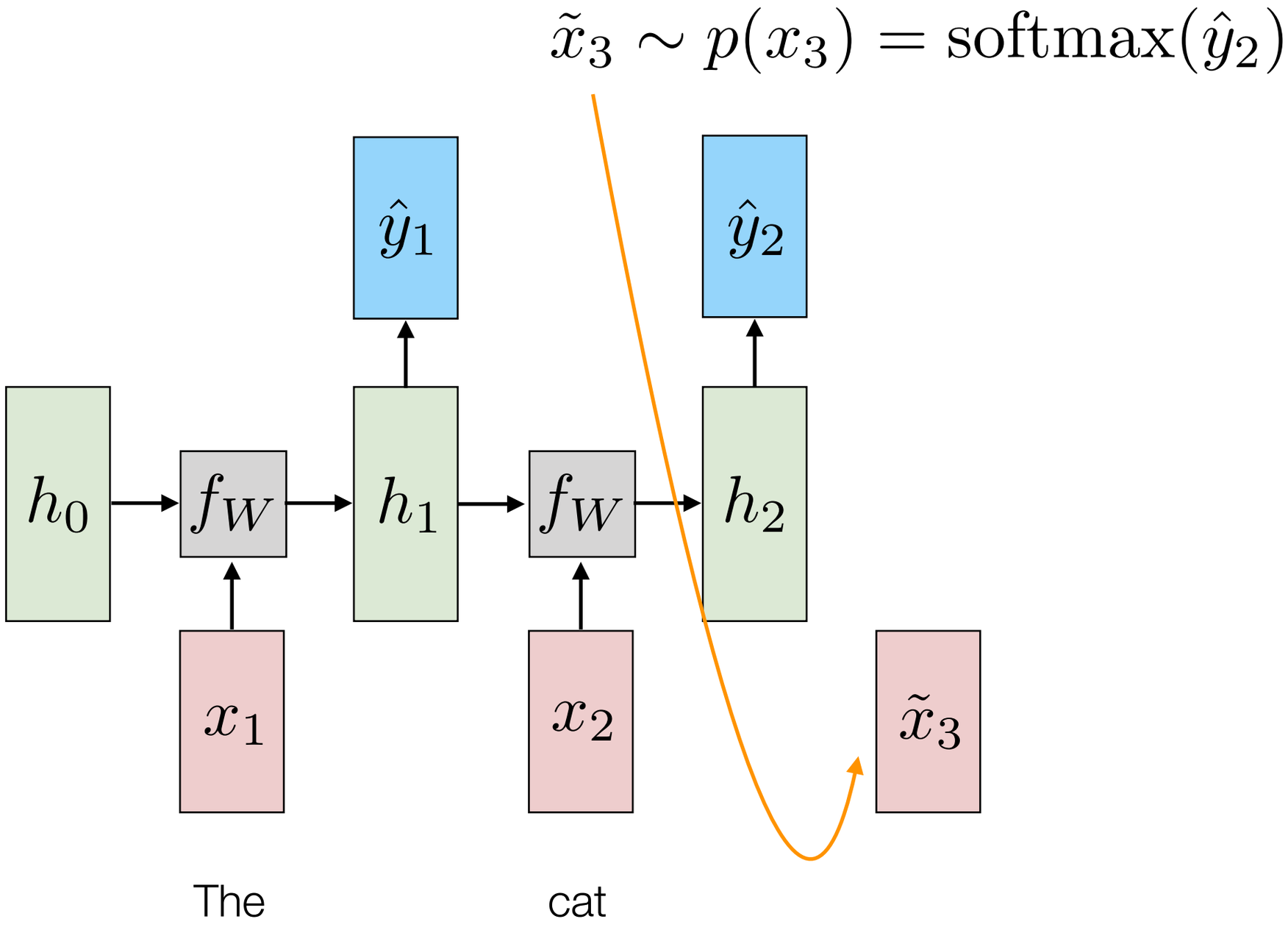
$$\tilde{x}_3 \sim p(x_3) = \text{softmax}(\hat{y}_2)$$



Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. InProceedings of INTERSPEECH

Generación de texto con RNNs (Test)

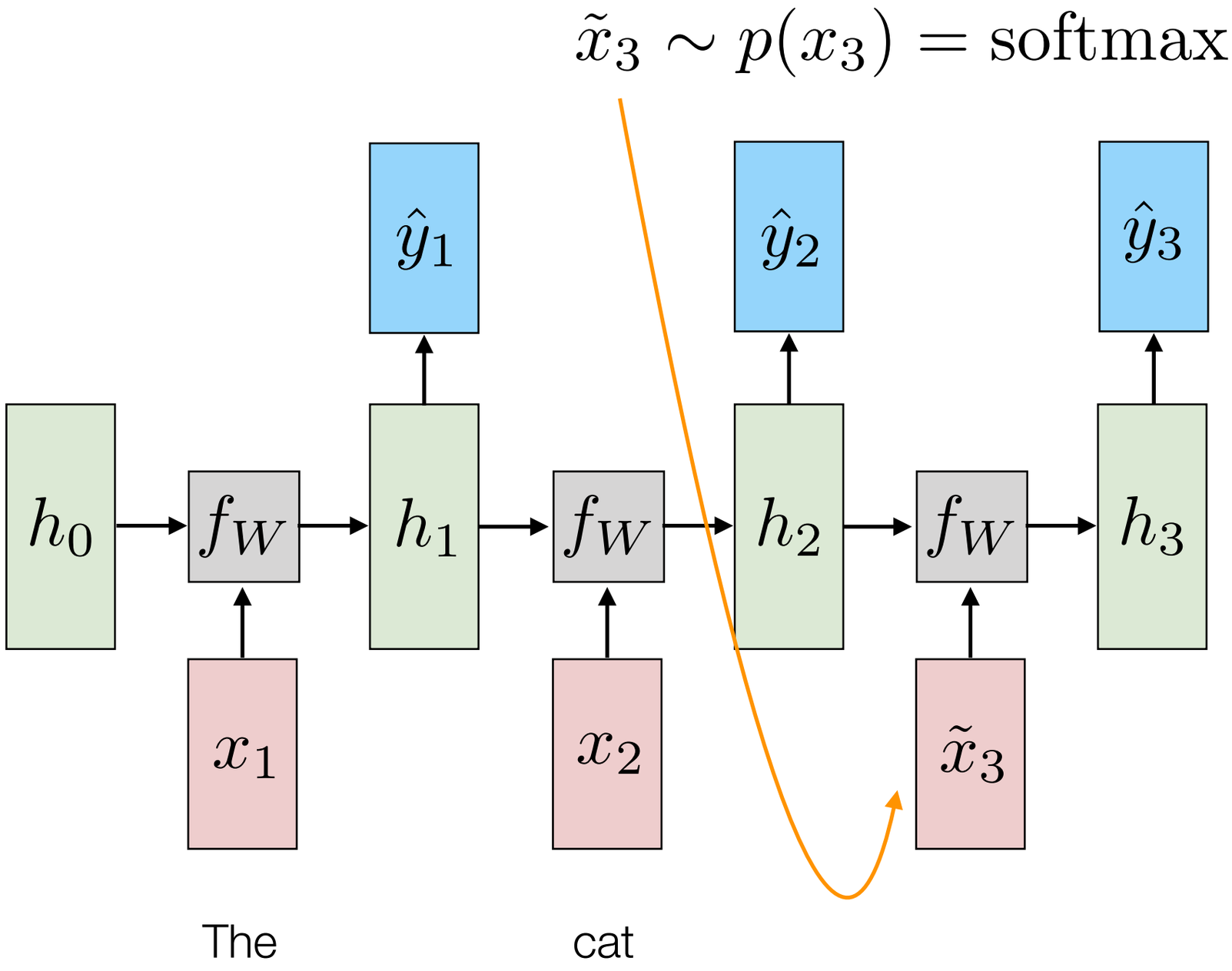
Muestreo secuencial …



Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. InProceedings of INTERSPEECH

Generación de texto con RNNs (Test)

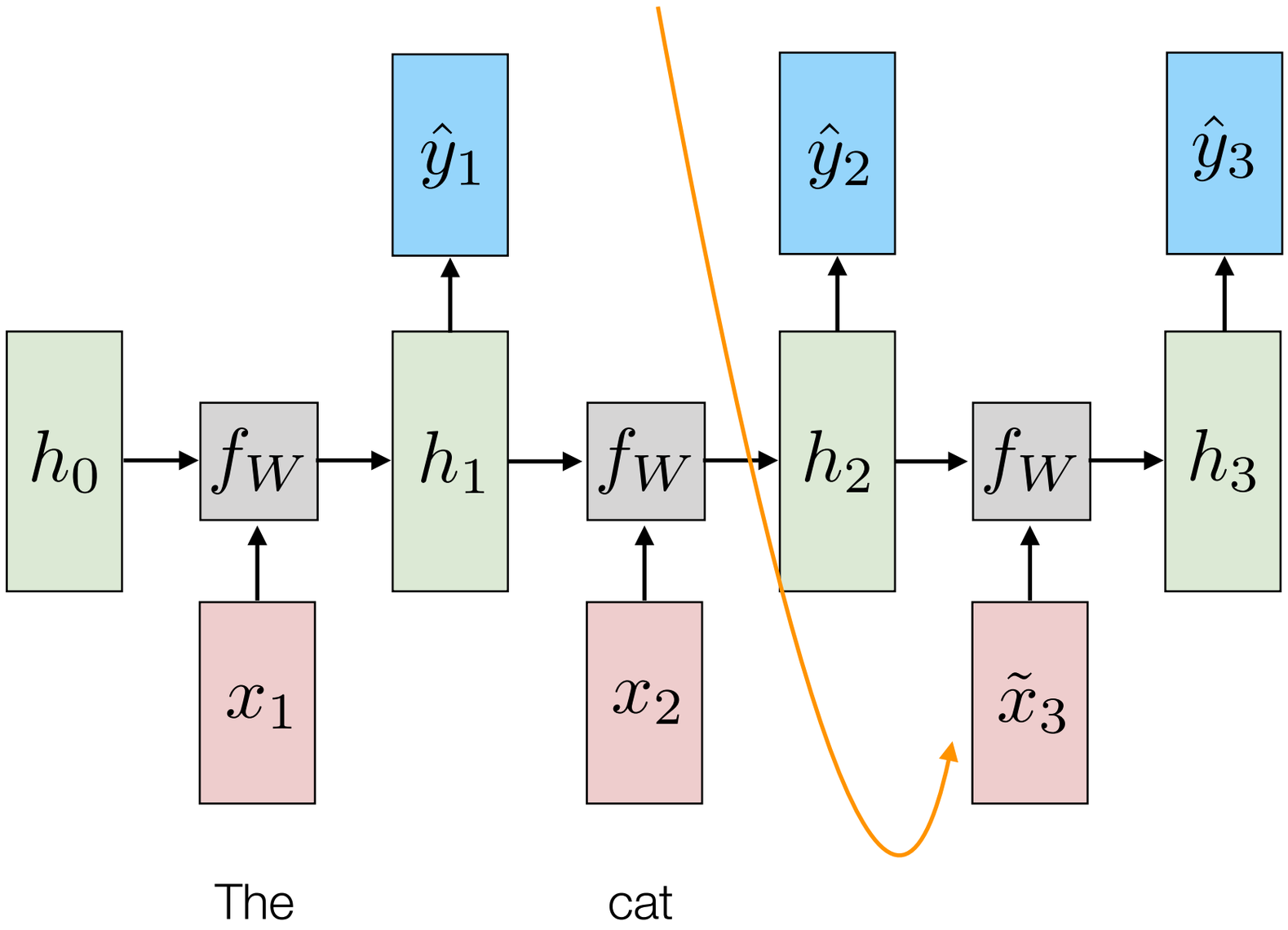
Muestreo secuencial …



Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. InProceedings of INTERSPEECH

Generación de texto con RNNs (Test)

Muestreo secuencial …

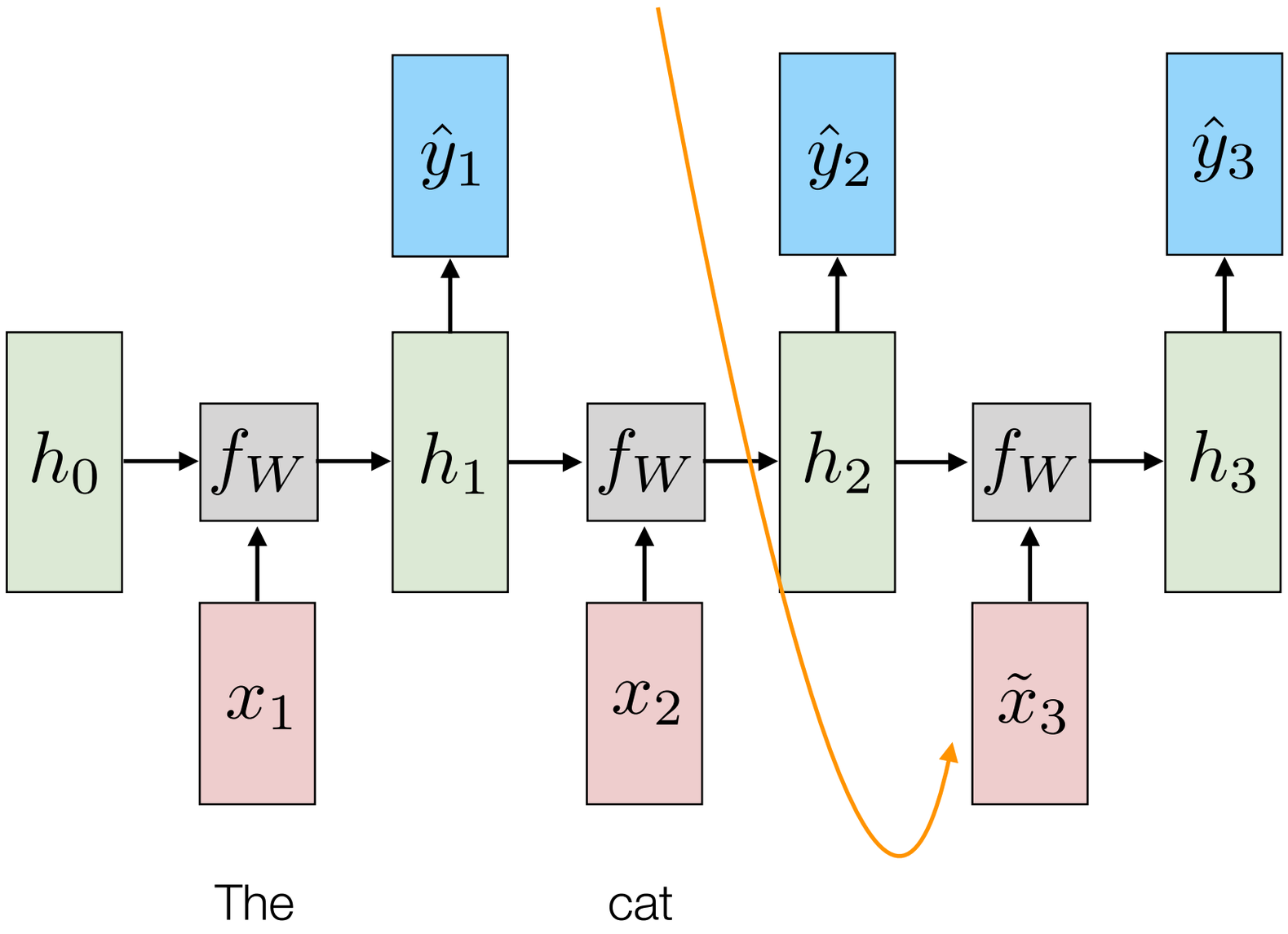


Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. InProceedings of INTERSPEECH

Generación de texto con RNNs (Test)

Muestreo secuencial …

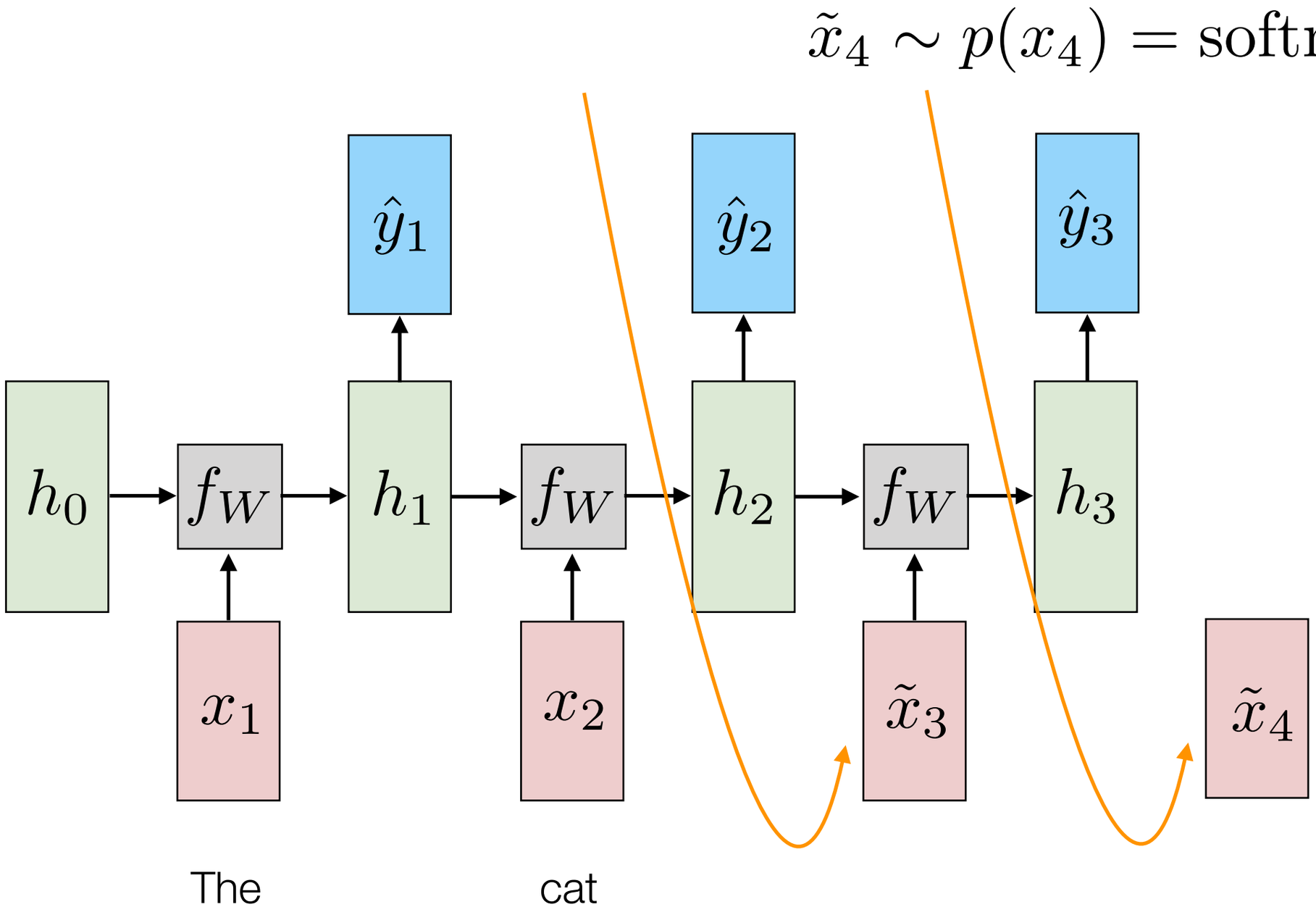
$$\tilde{x}_4 \sim p(x_4) = \text{softmax}(\hat{y}_3)$$



Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. InProceedings of INTERSPEECH

Generación de texto con RNNs (Test)

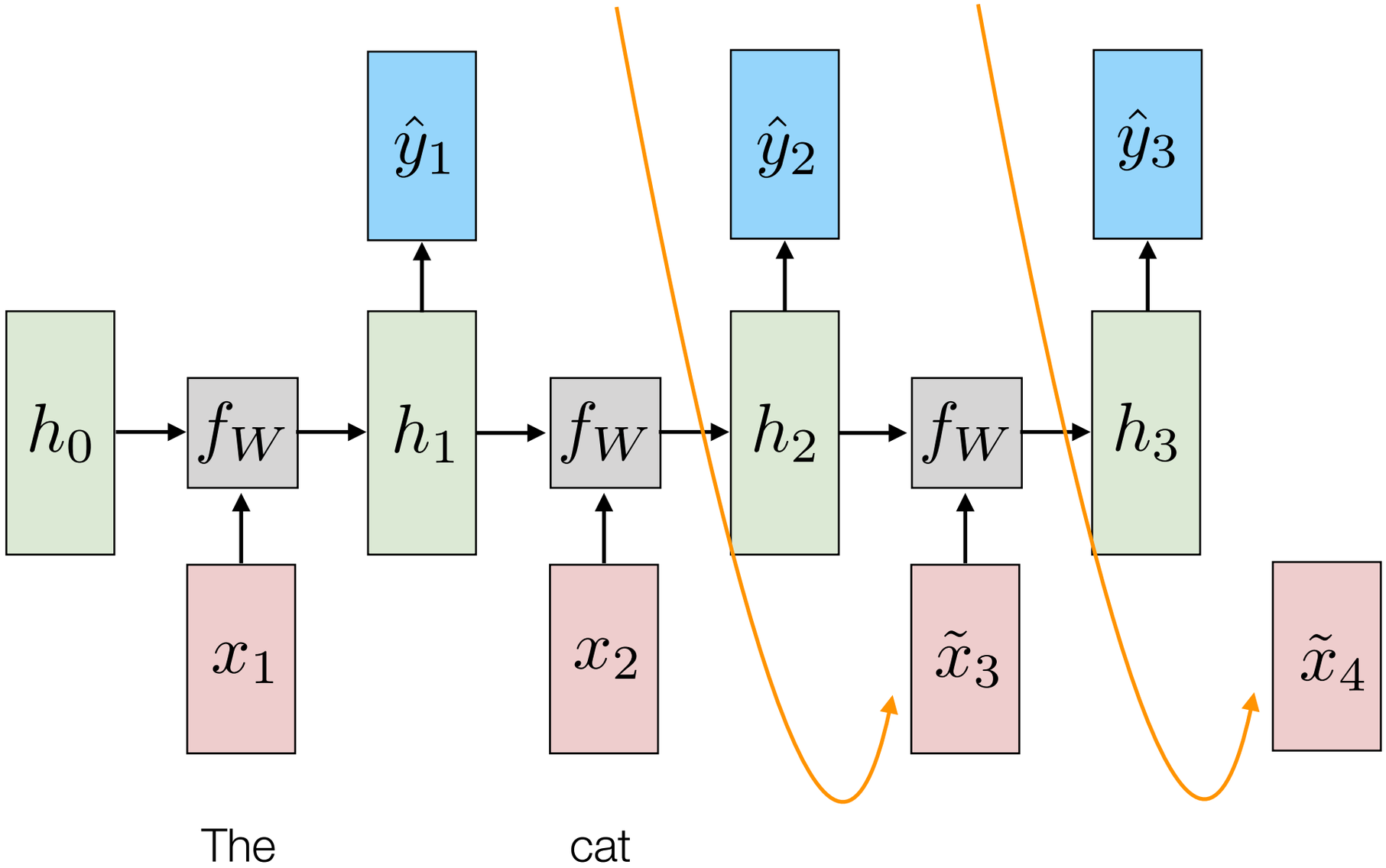
Muestreo secuencial …



Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. InProceedings of INTERSPEECH

Generación de texto con RNNs (Test)

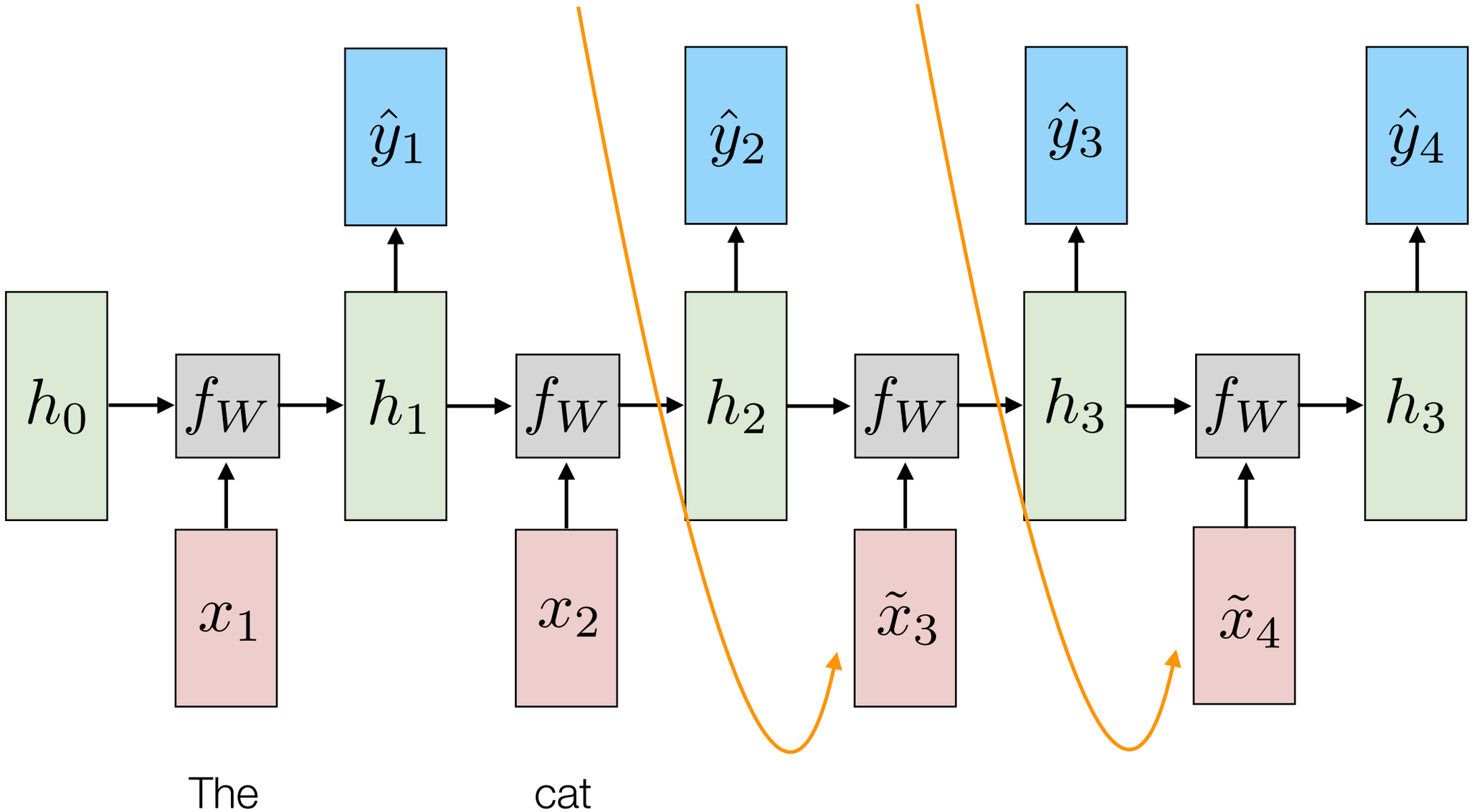
Muestreo secuencial …



Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. InProceedings of INTERSPEECH

Generación de texto con RNNs (Test)

Muestreo secuencial …

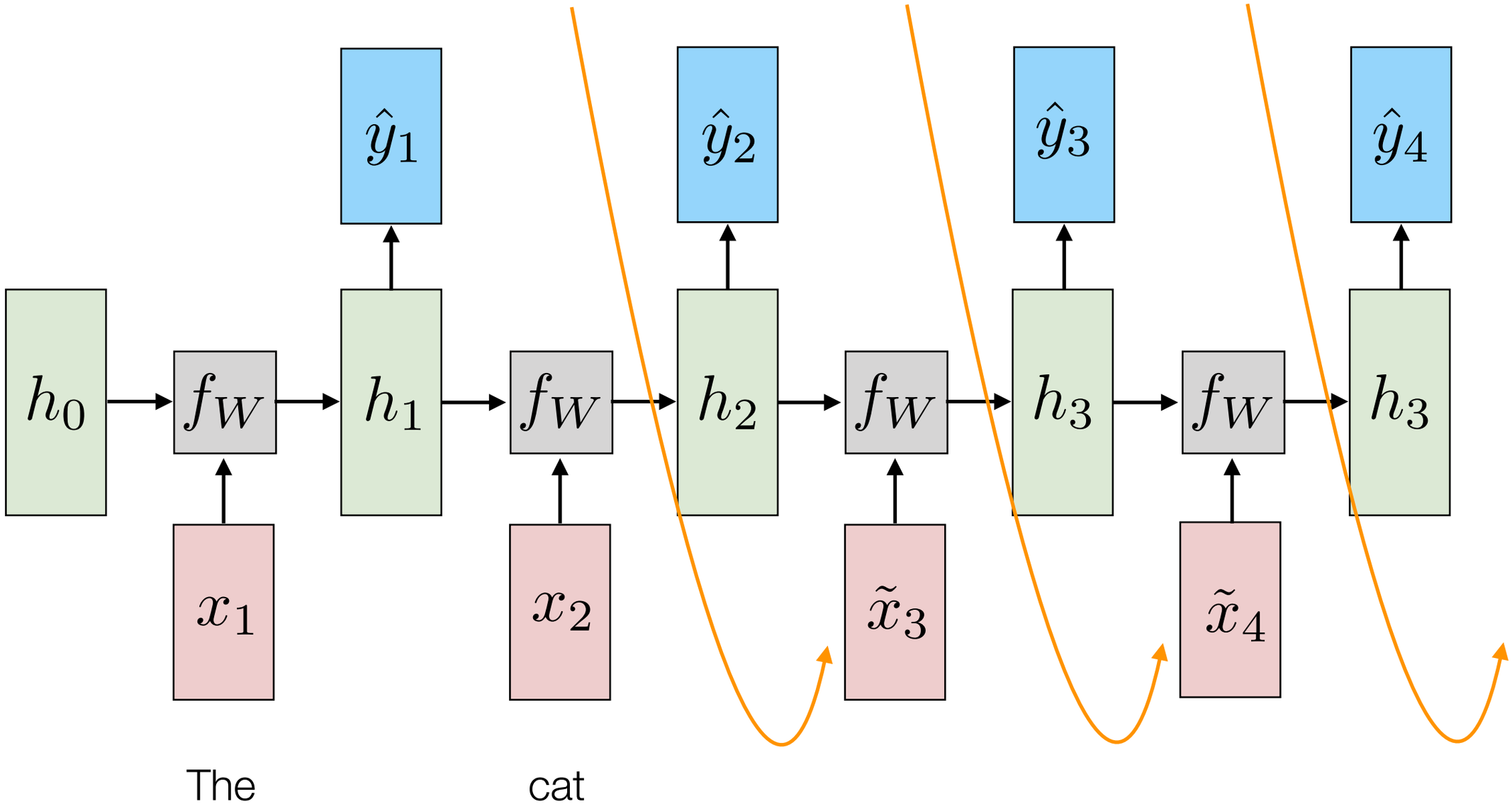


Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. InProceedings of INTERSPEECH

Generación de texto con RNNs (Test)

Muestreo secuencial …

$$\tilde{x}_5 \sim p(x_5) = \text{softmax}(\hat{y}_4)$$

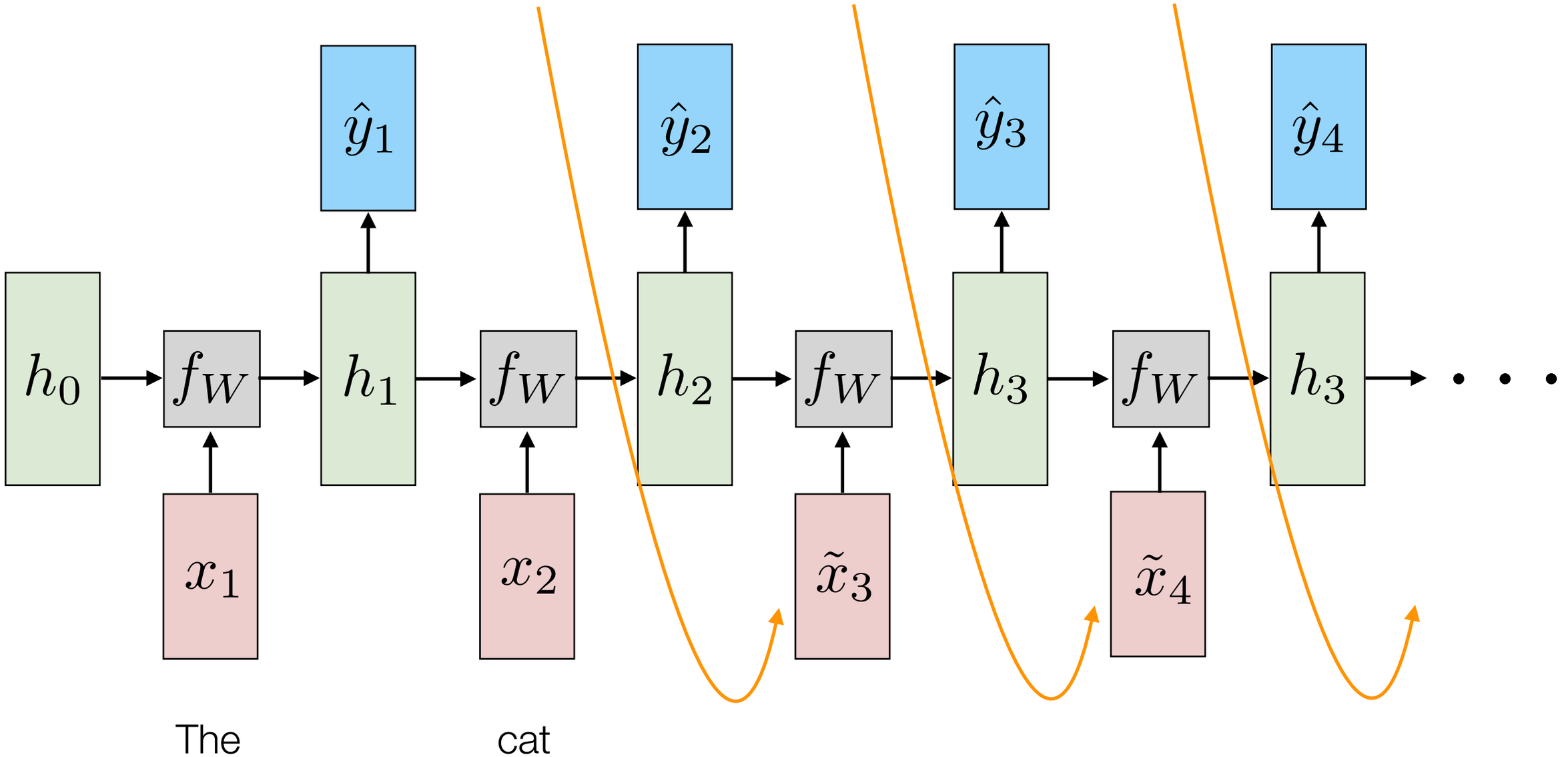


Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. InProceedings of INTERSPEECH

Generación de texto con RNNs (Test)

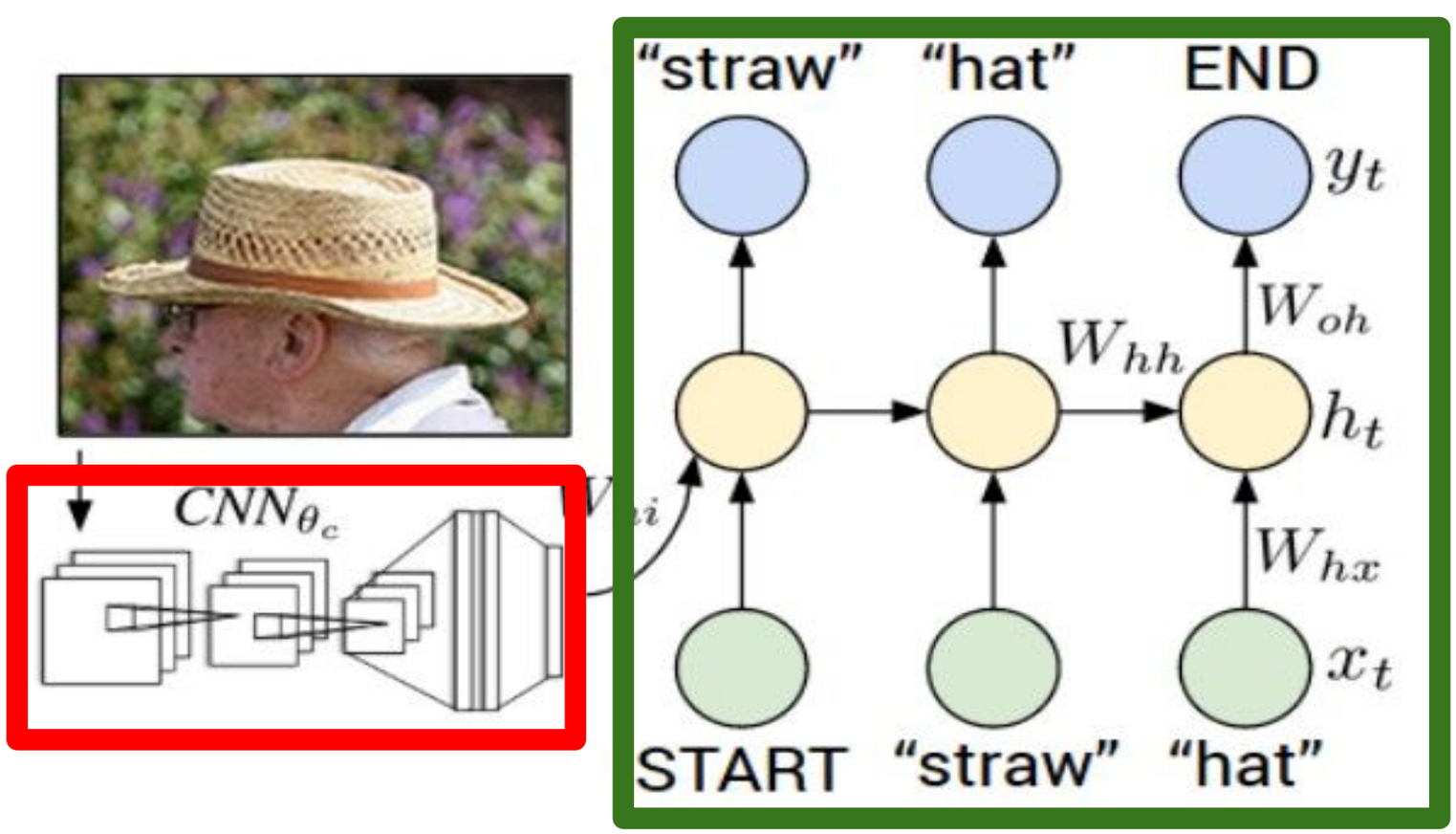
Muestreo secuencial …

$$\tilde{x}_5 \sim p(x_5) = \text{softmax}(\hat{y}_4)$$



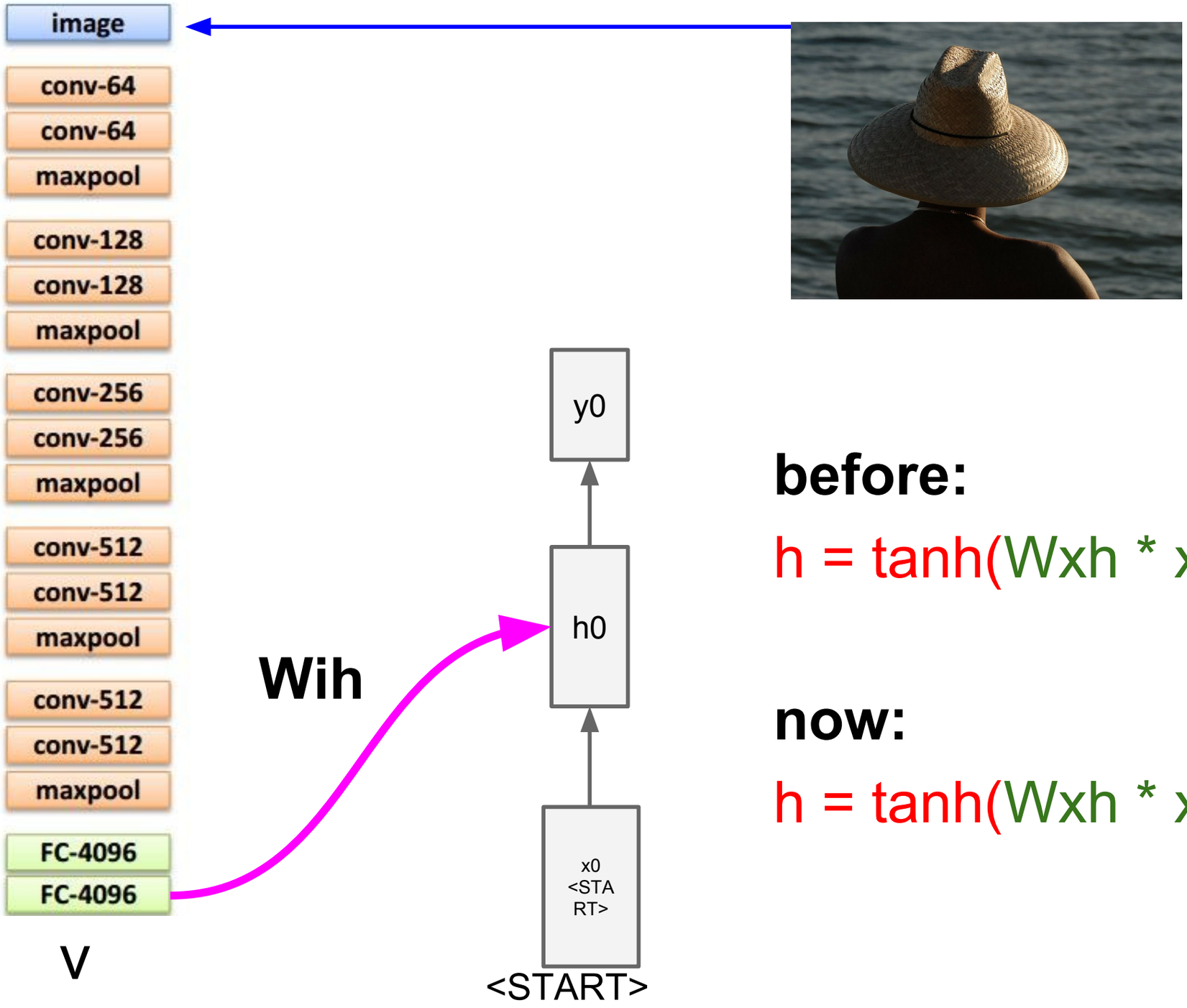
Tomas Mikolov, Martin Karafiat, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent Neural Network Based Language Model. InProceedings of INTERSPEECH

Descripción automática de imágenes



Explain Images with Multimodal Recurrent Neural Networks, Mao et al.
Deep Visual-Semantic Alignments for Generating Image Descriptions, Karpathy and Fei-Fei
Show and Tell: A Neural Image Caption Generator, Vinyals et al.
Long-term Recurrent Convolutional Networks for Visual Recognition and Description, Donahue et al.
Learning a Recurrent Visual Representation for Image Caption Generation, Chen and Zitnick

Descripción automática de imágenes



before:

h = tanh(Wxh * x + Whh * h)

now:

h = tanh(Wxh * x + Whh * h + **Wih * v**)

Descripción automática de imágenes

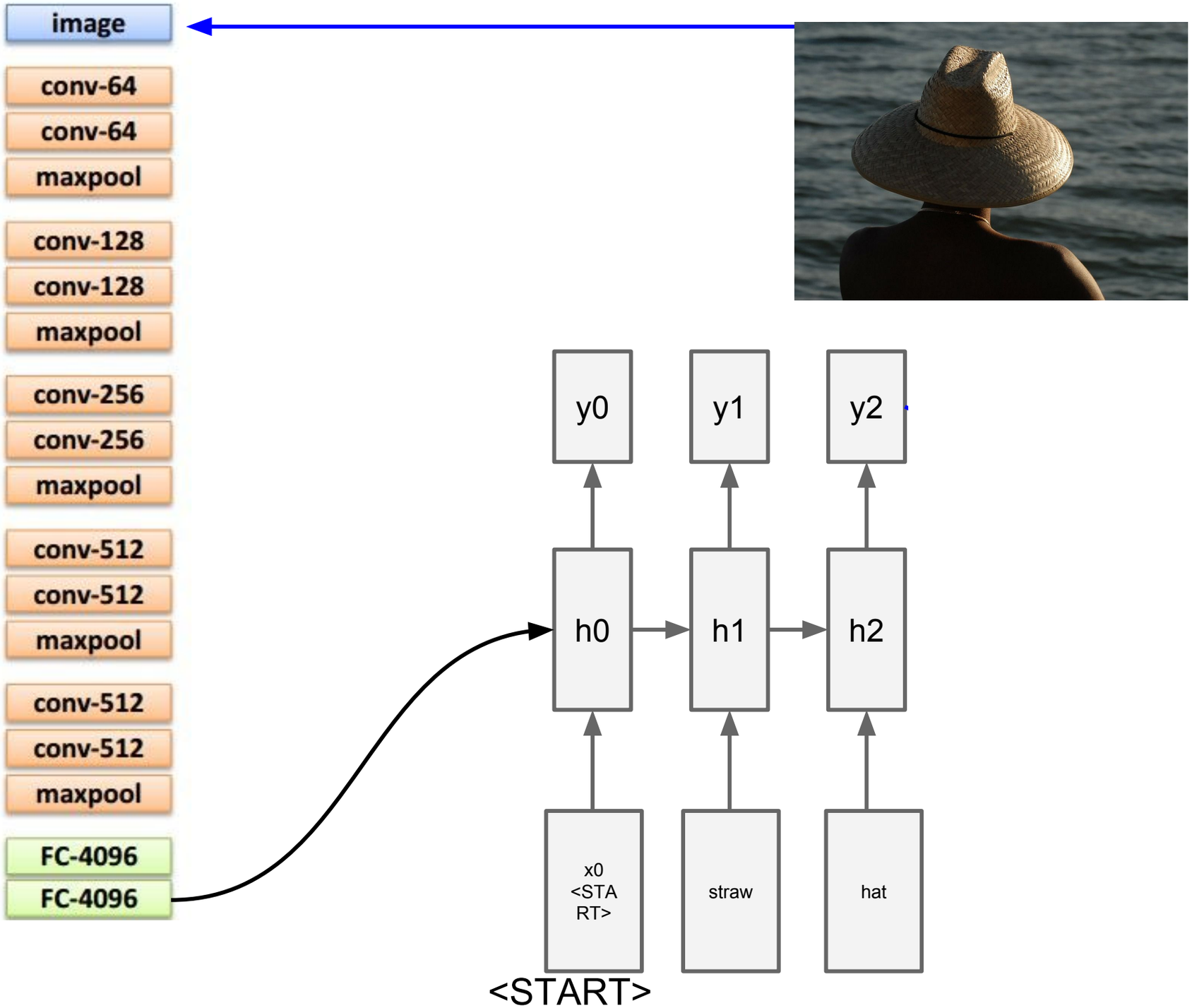


Image Captioning: Example Results

Captions generated using neuraltalk2
All images are CC0 Public domain:
cat suitcase, cat tree, dog, bear,
surfers, tennis, giraffe, motorcycle

A cat sitting on a suitcase on the floor

A cat is sitting on a tree branch

A dog is running in the grass with a frisbee

A white teddy bear sitting in the grass

Two people walking on the beach with surfboards

A tennis player in action on the court

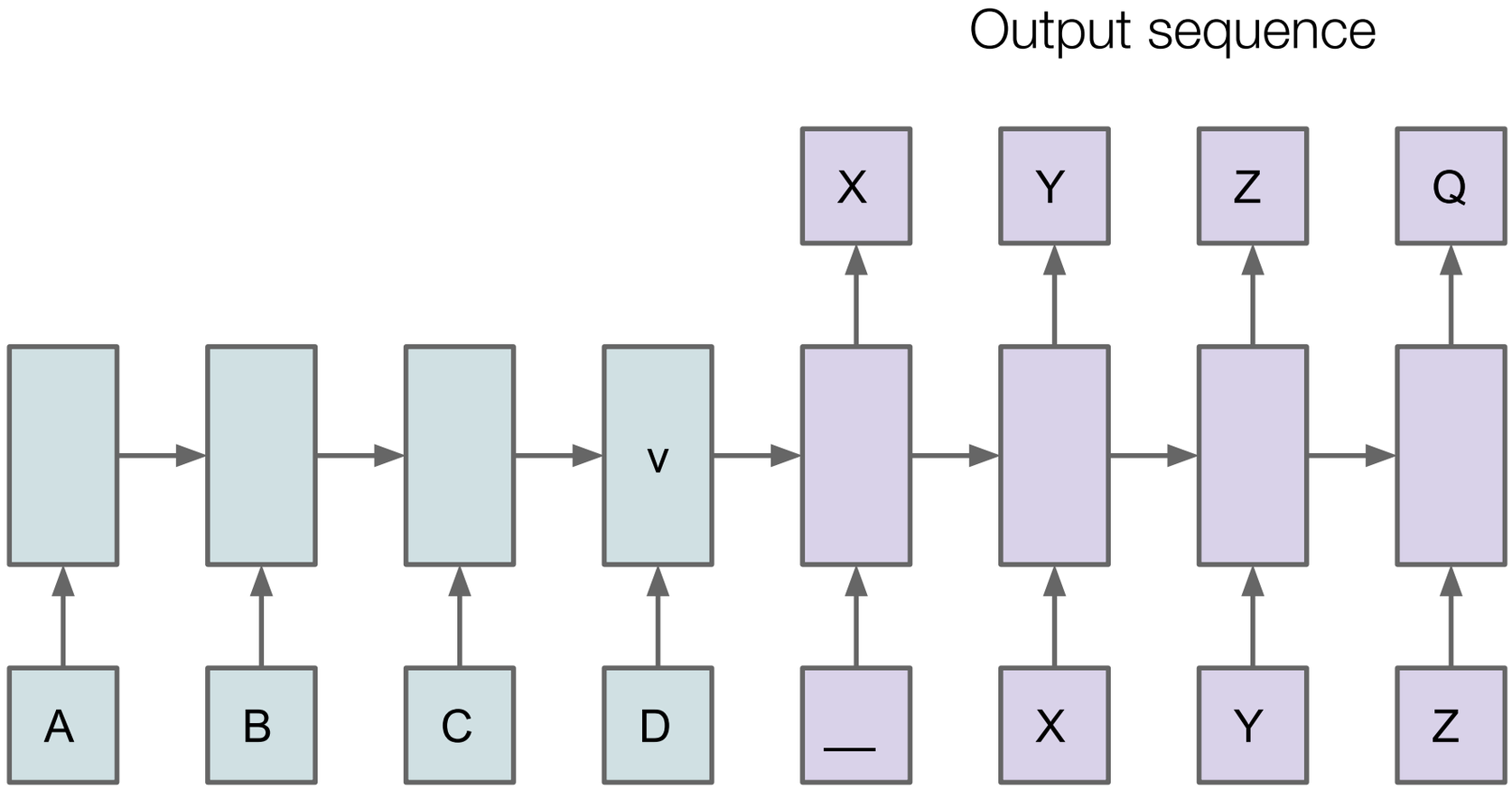
Two giraffes standing in a grassy field

A man riding a dirt bike on a dirt track

Seq2Seq y modelos de atención

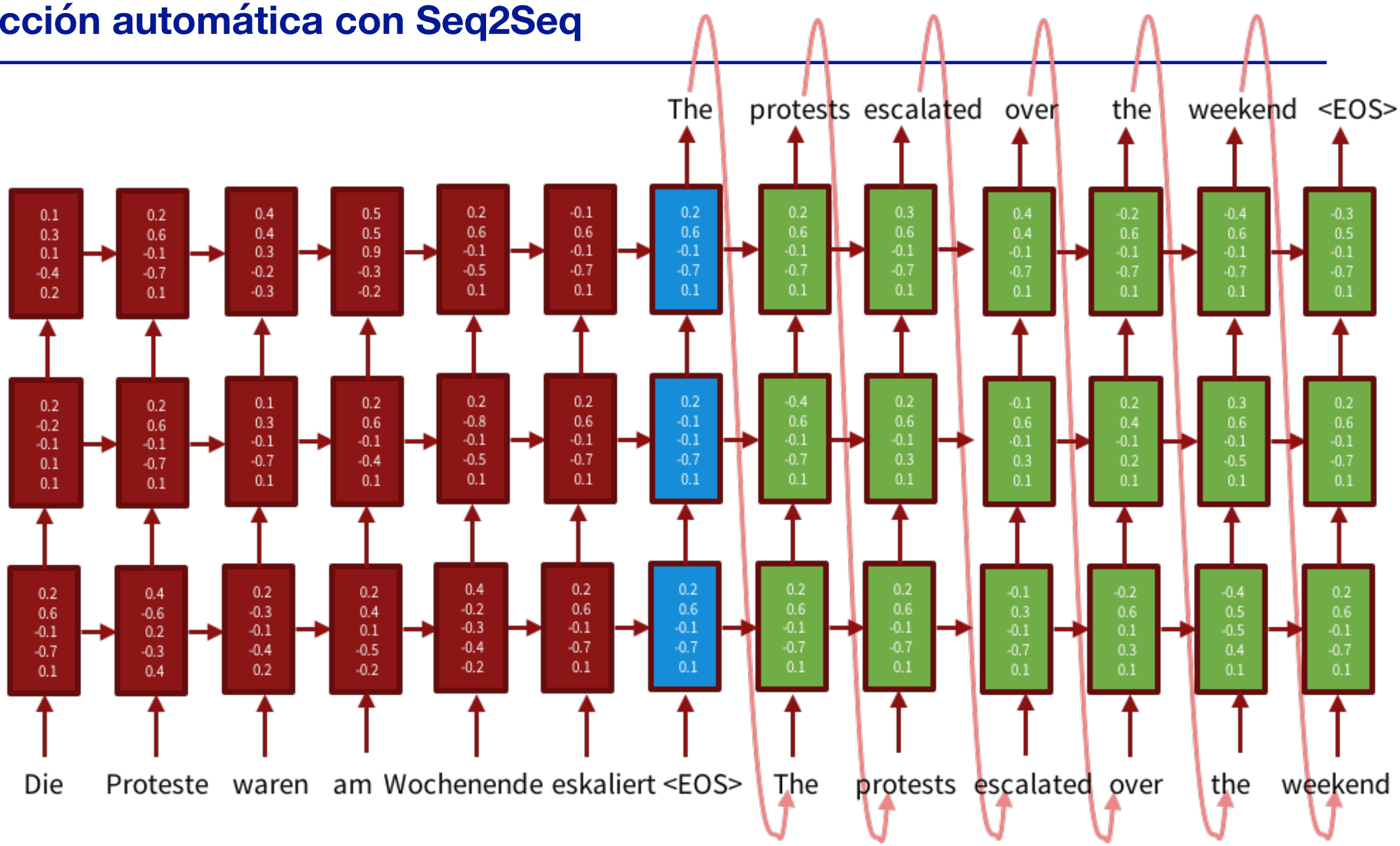
Procesamiento de Lenguaje Natural

Modelo Seq2Seq



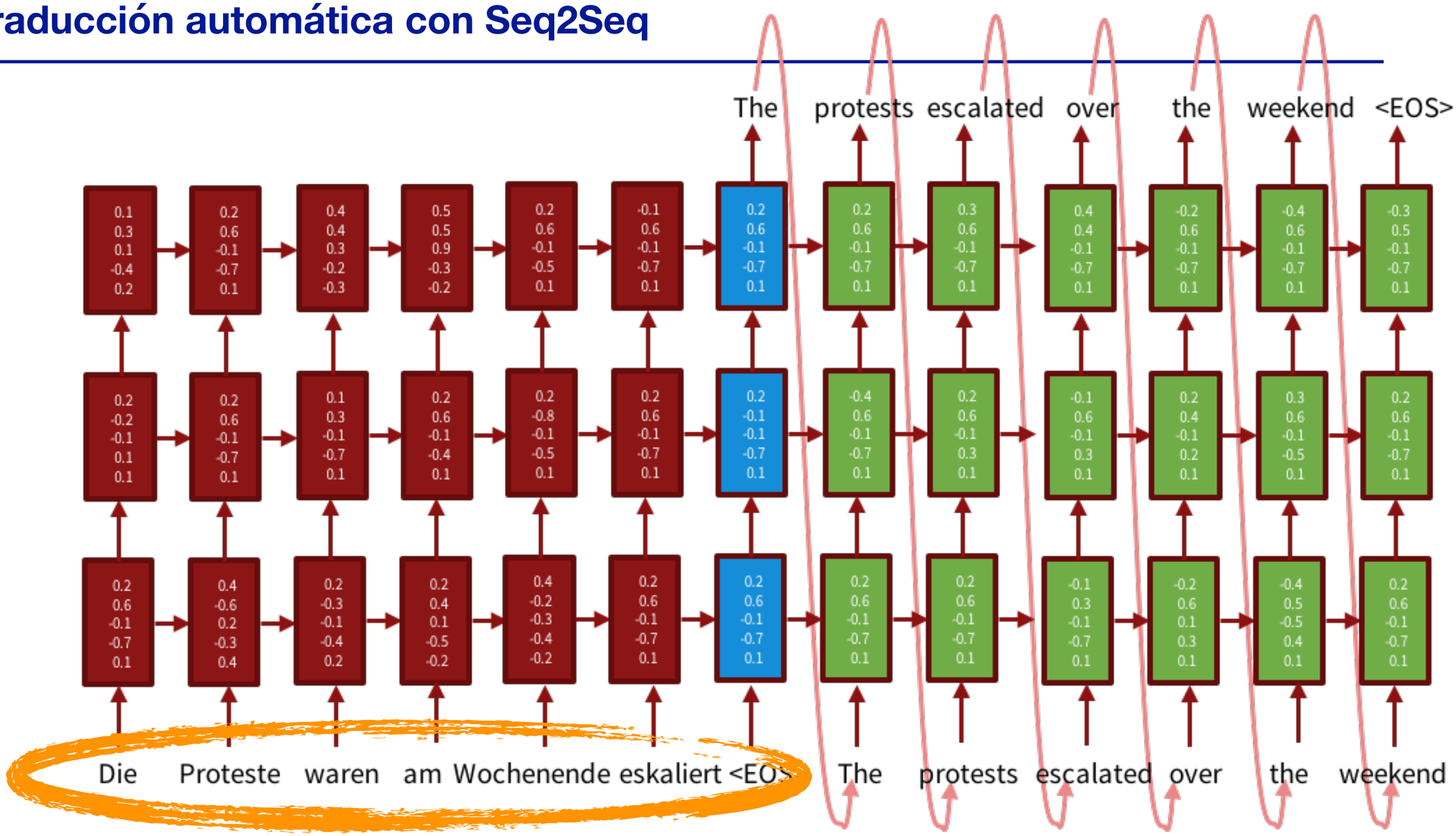
$$P\left(y_1, \ldots, y_{T'} | x_1, \ldots, x_T\right) = \prod_{t=1}^{T'} p\left(y_t | v, y_1, \ldots, y_{t-1}\right)$$

Traducción automática con Seq2Seq



1. Auli, M., et al. "Joint Language and Translation Modeling with Recurrent Neural Networks." *EMNLP (2013)*
2. Kalchbrenner, N., et al. “Recurrent Continuous Translation Models.” *EMNLP (2013)*
3. Cho, K., et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical MT.” *EMNLP (2014)*
4. Sutskever, I., et al. “Sequence to Sequence Learning with Neural Networks.” *NIPS (2014)*

Traducción automática con Seq2Seq

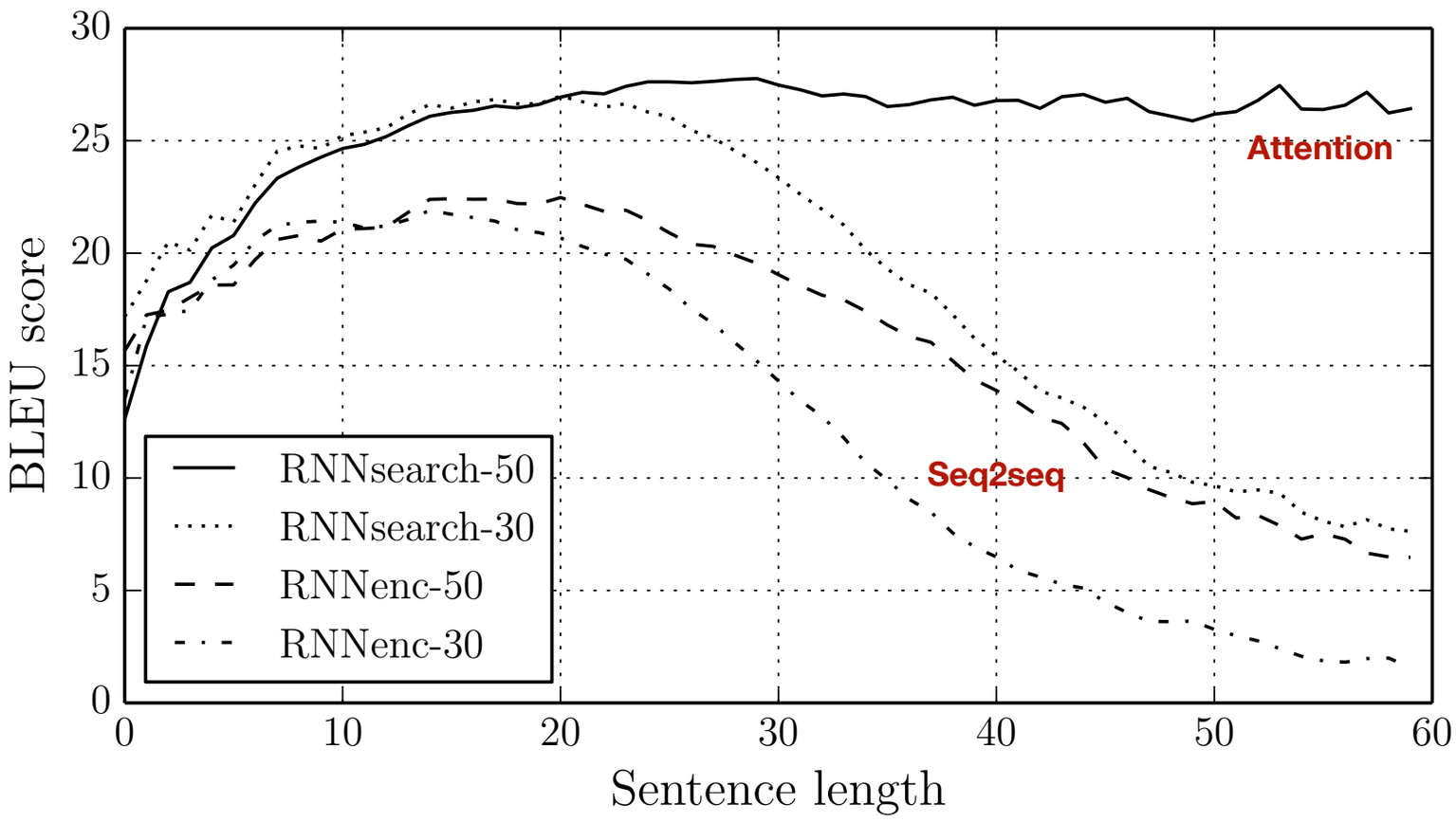


Word Embeddings!

1. Auli, M., et al. "Joint Language and Translation Modeling with Recurrent Neural Networks." *EMNLP (2013)*
2. Kalchbrenner, N., et al. “Recurrent Continuous Translation Models.” *EMNLP (2013)*
3. Cho, K., et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical MT.” *EMNLP (2014)*
4. Sutskever, I., et al. “Sequence to Sequence Learning with Neural Networks.” *NIPS (2014)*

Limitaciones Seq2Seq para traducción automática

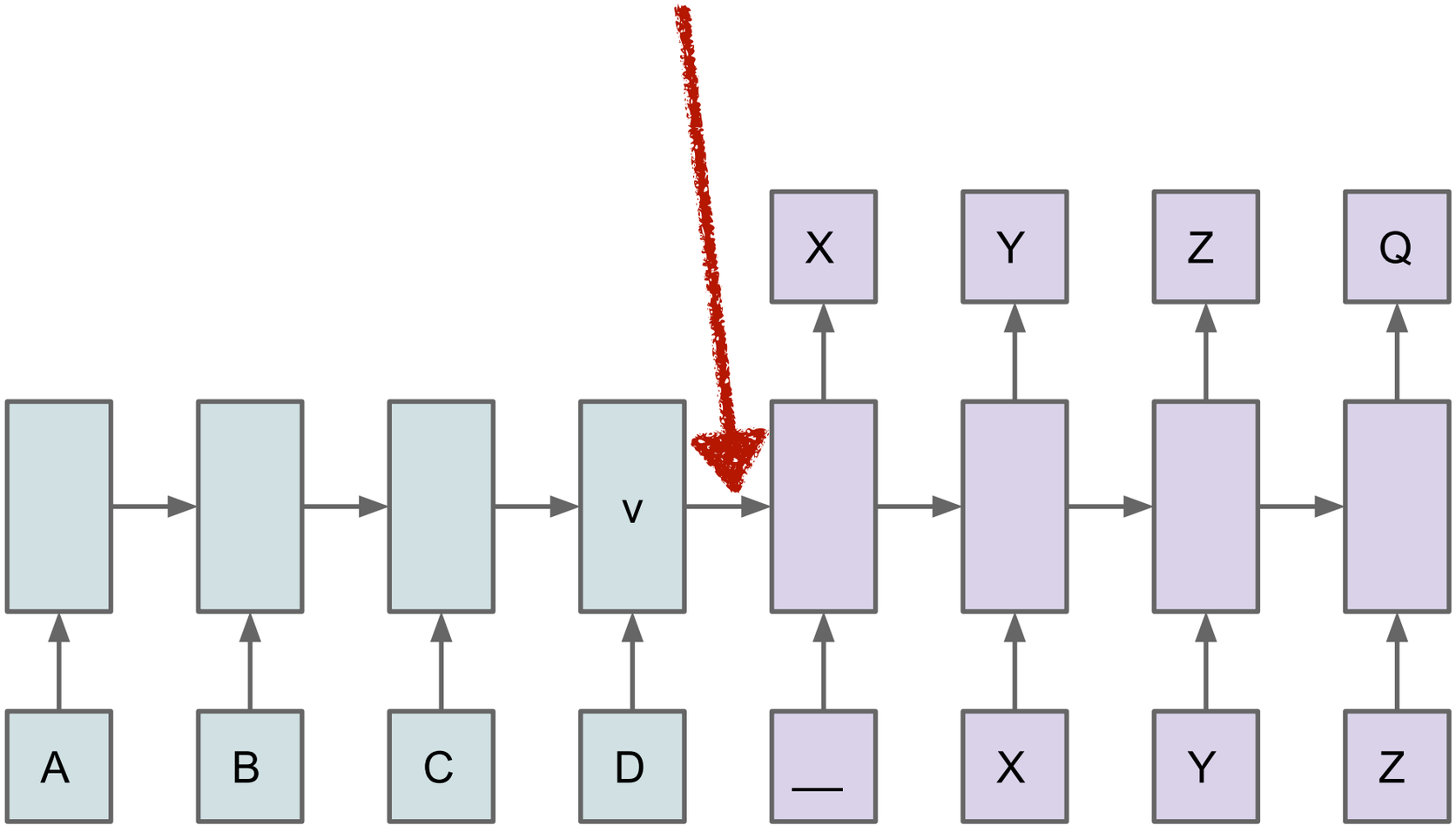
- Rendimiento cae rápidamente con la longitud de la secuencia
- BLEU (bilingual evaluation understudy) es uno métrica para evaluar la calidad de una traducción



1. Sutskever, I., et al. “Sequence to Sequence Learning with Neural Networks.” *NIPS (2014)*
2. Bahdanau, D., et al. “Neural Machine Translation by Jointly Learning to Align and Translate.” *ICLR (2015)*

Limitaciones Seq2Seq para traducción automática

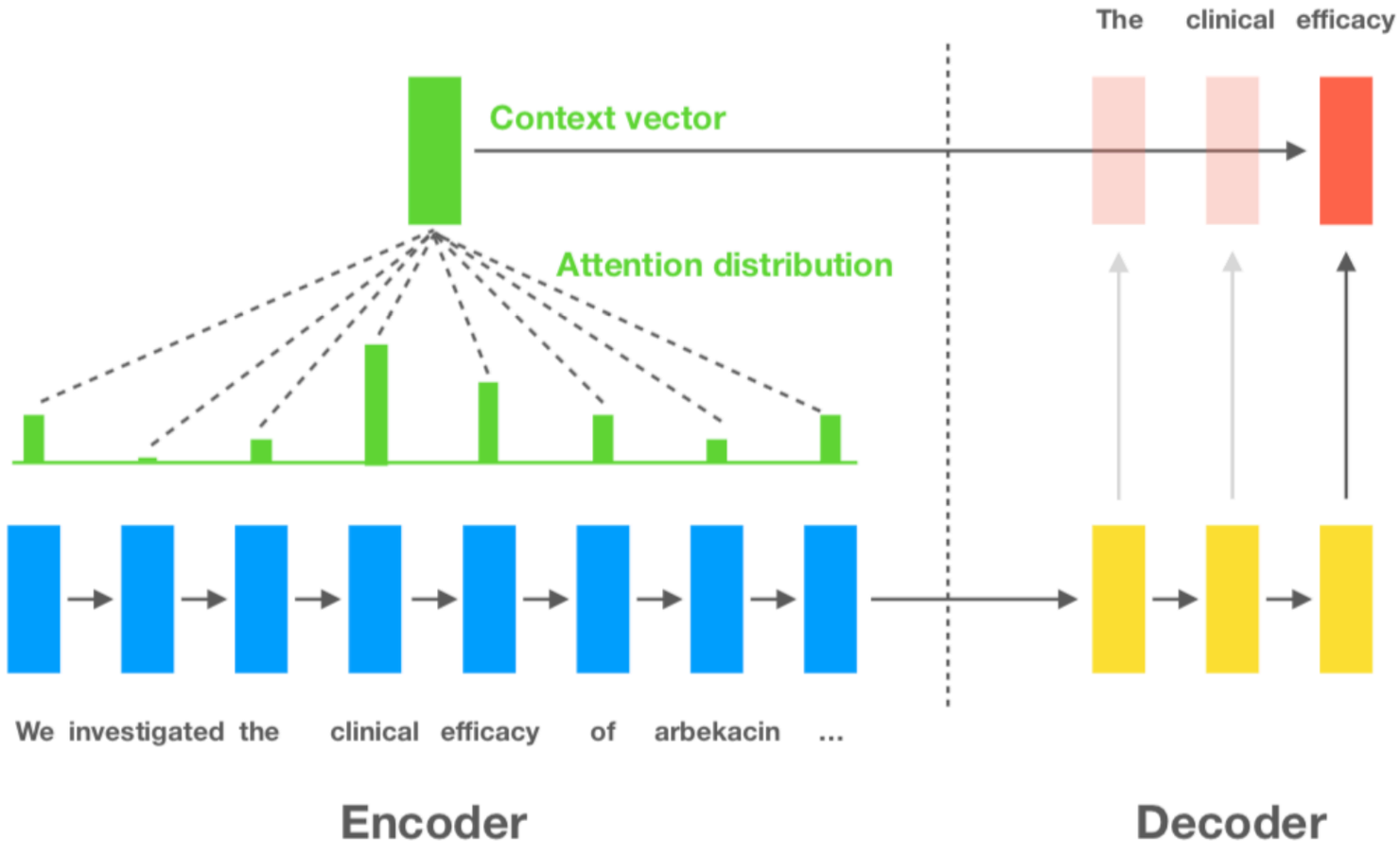
Un único embedding para representar toda la secuencia!



Seq2Seq con mecanismos de atención

- Permitimos al decodificador prestar **atención a diferentes partes de la secuencia de entrada** en cada paso de la generación de la secuencia de salida.

- Cada palabra generada por el decodificador estará condicionada a una **suma ponderada de los estados en el codificador**.

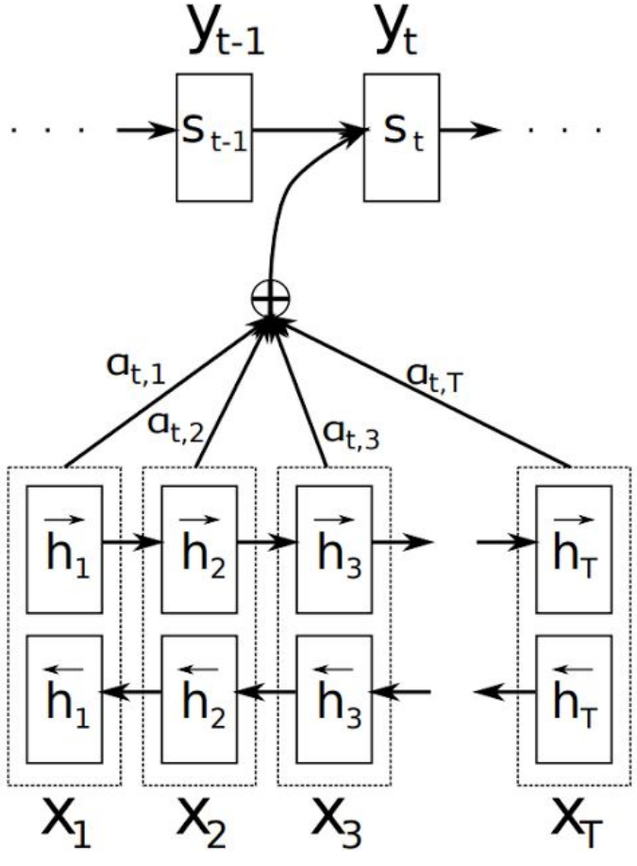


Seq2Seq con mecanismos de atención

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{T_x} \exp\left(e_{ik}\right)},$$

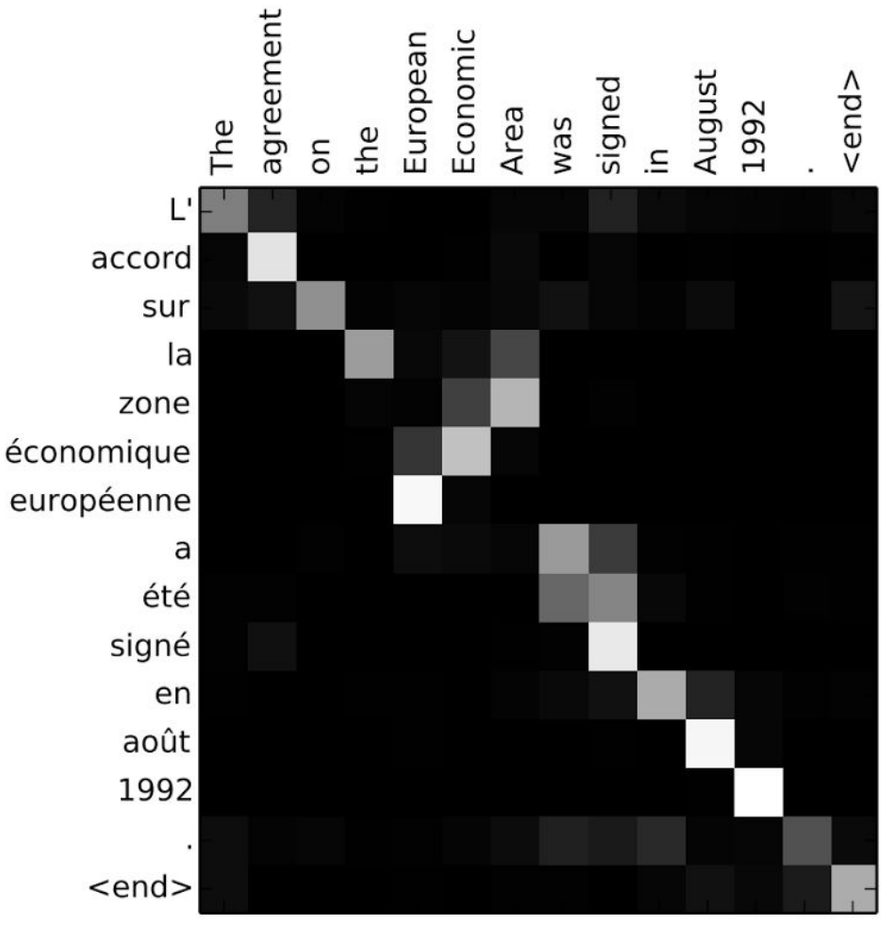
$$e_{ij} = a(s_{i-1}, h_j)$$



NEURAL MACHINE TRANSLATION
BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau
Jacobs University Bremen, Germany

KyungHyun Cho **Yoshua Bengio***
Université de Montréal



Bahdanau, D., et al. “Neural Machine Translation by Jointly Learning to Align and Translate.” *ICLR (2015)*

Trasformer networks

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

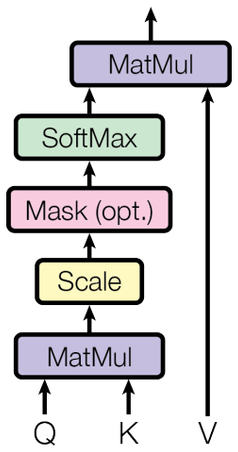
Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

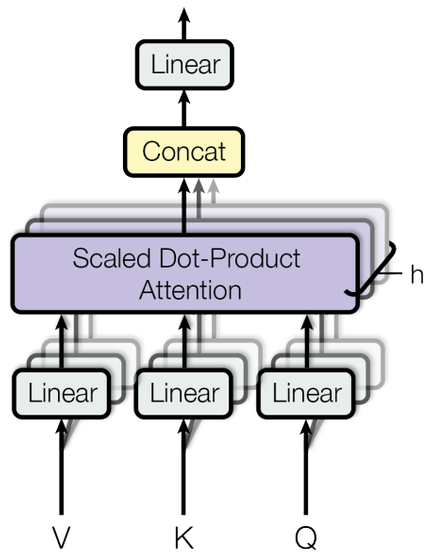
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Scaled Dot-Product Attention



Multi-Head Attention

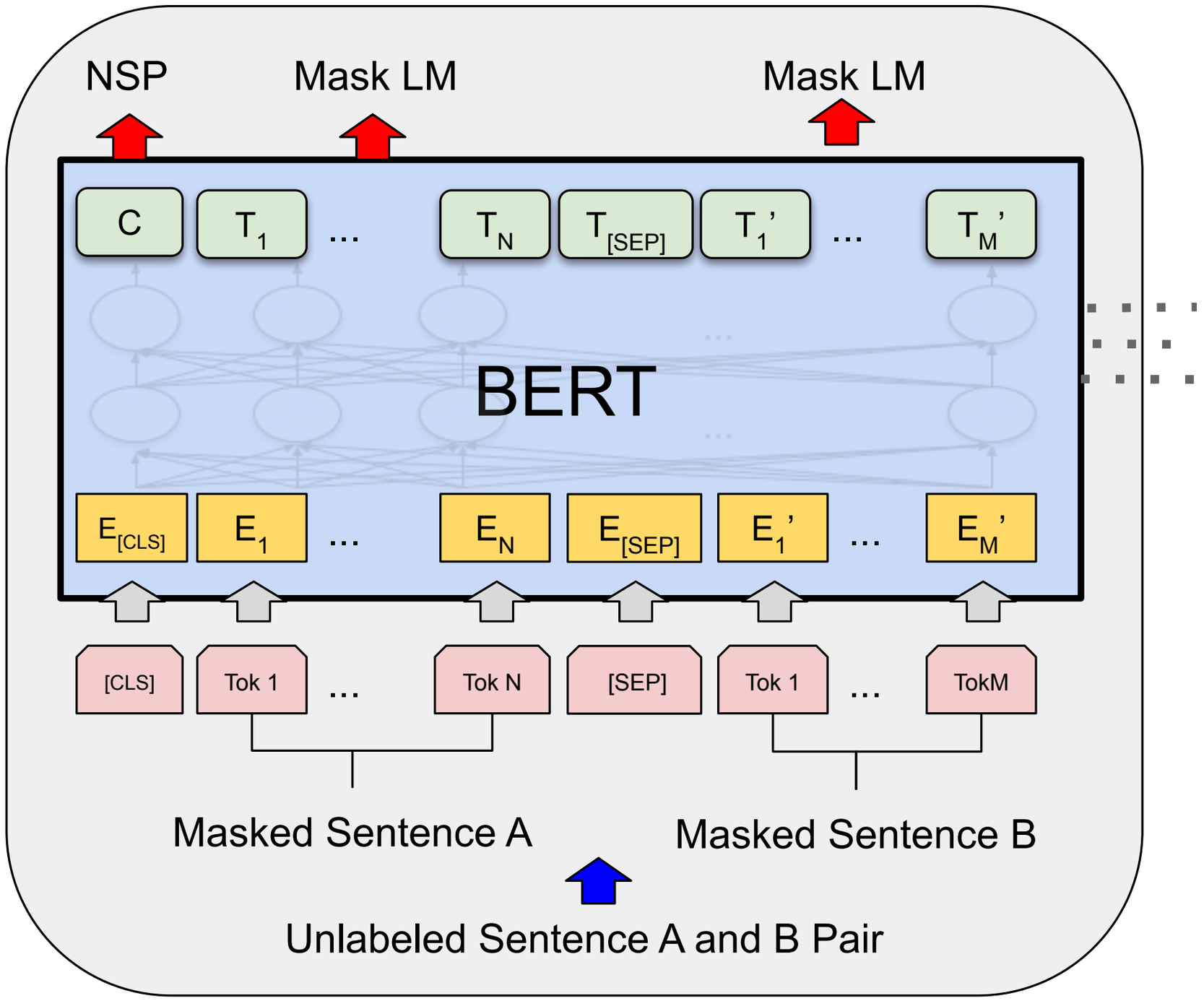


Dec 2017

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-

In this work we propose the Transformer, a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output.

BERT: Pre-training of Deep bidirectional Transformers for Language Understanding



Pre-training

AN NLP TIMELINE AND THE TRANSFORMER FAMILY

BAG OF WORDS (BOW)
Count the occurrences of each word in the documents and use them as features.
1954

TF-IDF
The BOW scores are modified so that rare words have high scores and common words have low scores.
1972

WORD2VEC
Each word is mapped to a high-dimensional vector called word embedding, which captures its semantic. Word embeddings are learned by a neural network looking for word correlations on a large corpus.
2013

RNN
RNNs compute document embeddings leveraging word context in sentences, which was not possible with word embeddings alone.

LSTM
Capture long term dependencies.
1997

Bidirectional RNN
Capture left-to-right and right-to-left dependencies.
1997

Encoder-decoder RNN
An RNN creates a document embedding (i.e. the encoder) and another RNN decodes it into text (i.e. the decoder).
2014

1986

TRANSFORMER
An encoder-decoder model that leverages attention mechanisms to compute better embeddings and to better align output to input.
2017

BERT
Bidirectional Transformer pretrained using a combination of Masked Language Modeling and Next Sentence Prediction objectives. It uses global attention.
2018

GPT
The first autoregressive model based on the Transformer architecture.

GPT-2
A bigger and optimized version of GPT, pre-trained on WebText.
2019

GPT-3
A bigger and optimized version of GPT-2, pre-trained on Common Crawl.
2020

2018

CTRL
Similar to GPT but with control codes for conditional text generation.
2019

TRANSFORMER-XL
It's an autoregressive Transformer which can reuse previously computed hidden-states to attend to longer context.
2019

ALBERT
A lighter version of BERT, where (1) Next Sentence Prediction is replaced by Sentence Order Prediction, and (2) parameter-reduction techniques are used for lower memory consumption and faster training.
2019

ROBERTA
Better version of BERT, where (1) the Masked Language Modeling objective is dynamic, (2) the Next Sentence Prediction objective is dropped, (3) the BPE tokenizer is employed, and (4) better hyperparameters are used.
2019

XLM
Transformer pre-trained on a corpus of several languages using objectives like Causal Language Modeling, Masked Language Modeling, and Translation Language Modeling.
2019

XLNET
Transformer-XL with a generalized autoregressive pre-training method that enables learning bidirectional dependences.
2019

PEGASUS
A bidirectional encoder and a left-to-right decoder pre-trained with Masked Language Modeling and Gap Sentence Generation objectives.
2019

DISTILBERT
Same as BERT but smaller and faster, while preserving over 95% of BERT's performances. Trained by distillation of the pre-trained BERT model.
2019

XLM-ROBERTA
RoBERTa trained on a multilanguage corpus with the Masked Language Modeling objective.
2019

BART
A bidirectional encoder and a left-to-right decoder trained by corrupting text with an arbitrary noising function and learning a model to reconstruct the original text.
2019

CONVBERT
Better version of BERT, where self-attention blocks are replaced with new ones that leverage convolutions to better model global and local context.
2019

FUNNEL TRANSFORMER
A type of Transformer that gradually compresses the sequence of hidden states to a shorter one and hence reduces the computation cost.
2020

REFORMER
A more efficient Transformer thanks to local-sensitive hashing attention, axial position encoding and other optimizations.
2020

T5
A bidirectional encoder and a left-to-right decoder pre-trained on a mix of unsupervised and supervised tasks.
2020

LONGFORMER
A Transformer model replacing the attention matrices with sparse matrices for higher training efficiency.
2020

PROPHETNET
A Transformer model trained with the Future N-gram Prediction objective and with a novel self-attention mechanism.
2020

ELECTRA
Same as BERT but lighter and better. The model is trained with the Replaced Token Detection objective.
2020

SWITCH TRANSFORMER
A sparsely-activated expert Transformer model that aims to simplify and improve over Mixture of Experts.
2021

NLPLANET The community of NLP enthusiasts!
https://www.linkedin.com/company/nlplanet
https://twitter.com/nlplanet_
https://medium.com/nlplanet

By Fabio Chiusano

Librería para NLP con transformer networks : Hugging Face library

The AI community building the future.

https://huggingface.co

Build, train and deploy state of the art models powered by the reference open source in natural language processing.