Applied Deep Learning

Dr. Philippe Blaettchen
Bayes Business School (formerly Cass)

www.bayes.city.ac.uk

Learning objectives of today

Goals: Introduce recurrent neural networks (RNNs) as a means to work with sequence data

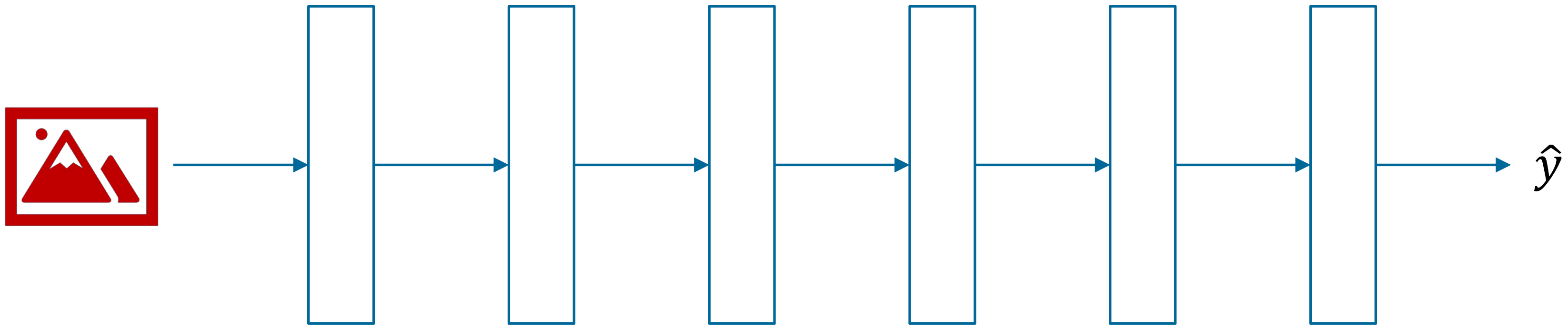
- Understand the importance of sequences in everyday life and machine learning
- Recognize the complexities of working with sequences
- Grasp the theoretical foundations of RNNs

How will we do this?

- We introduce sequence data and discuss its prevalence
- We then discuss the concept of recurrence of neurons, neural network layers, and entire networks
- We briefly discuss how gradient descent works for neural networks
- We then consider different RNN architectures as well as the challenges in training them

Working with sequences

What we did so far



But what if the order between  and  matters?

Sequences

Sequences are collections of multiple elements (i.e., data points), where:

- The order matters
- Elements may be repeated
- The length is variable

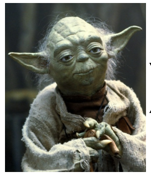
"Why do we care about sequences?"

Motivating sequences

“Why” “do” “we” “care” “about” “sequences” “?”

$\neq$

“care” “about” “sequences” “Why” “do” “we” “?”

(unless you are )

Motivating sequences

“a” “b” “o” “u” “t”

“Why” “do” “we” “care” “about” “sequences” “?”

“w” “e”

“s” “e” “q” “u” “e” “n” “c” “e” “s”

Keep in mind:

- The order matters
- We may repeat individual elements
- Sequences vary in length

Not just the input length varies, also the output length can vary

Warum kümmern wir uns um Sequenzen?

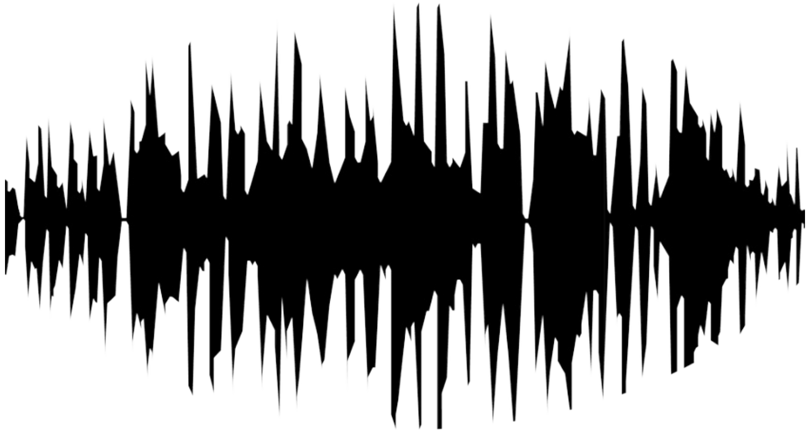
↑

“Why” “do” “we” “care” “about” “sequences” “?”

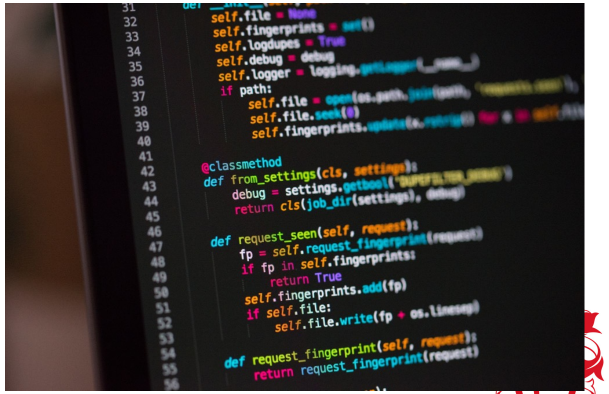
↓

¿Por qué nos preocupamos por las secuencias?

Sequences in real life

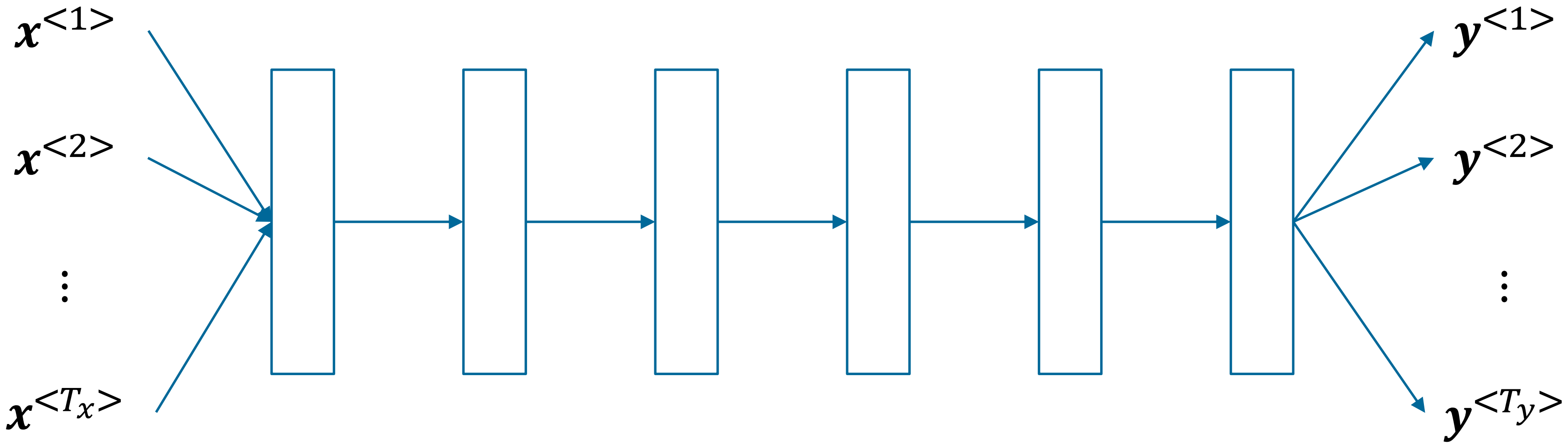


“And here I am,
for all my lore,
The wretched
fool I was
before”



Modeling sequences with neural networks

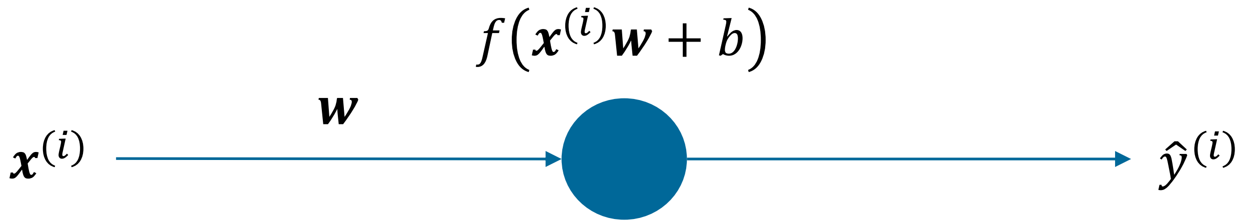
Why don’t we do this?



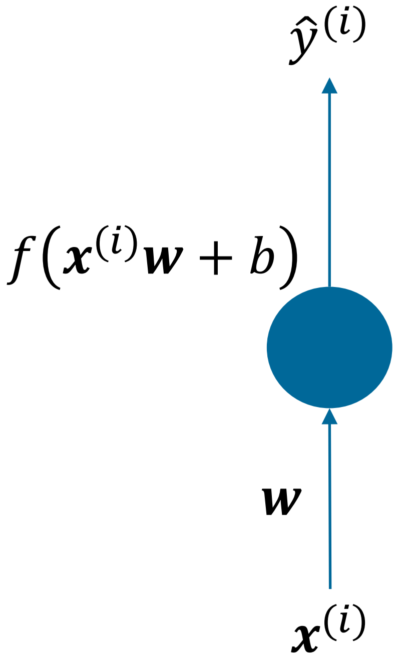
Issues:

- Sequence lengths vary
- No definition of order
- Lack of parameter sharing: imagine a minute-long ECG

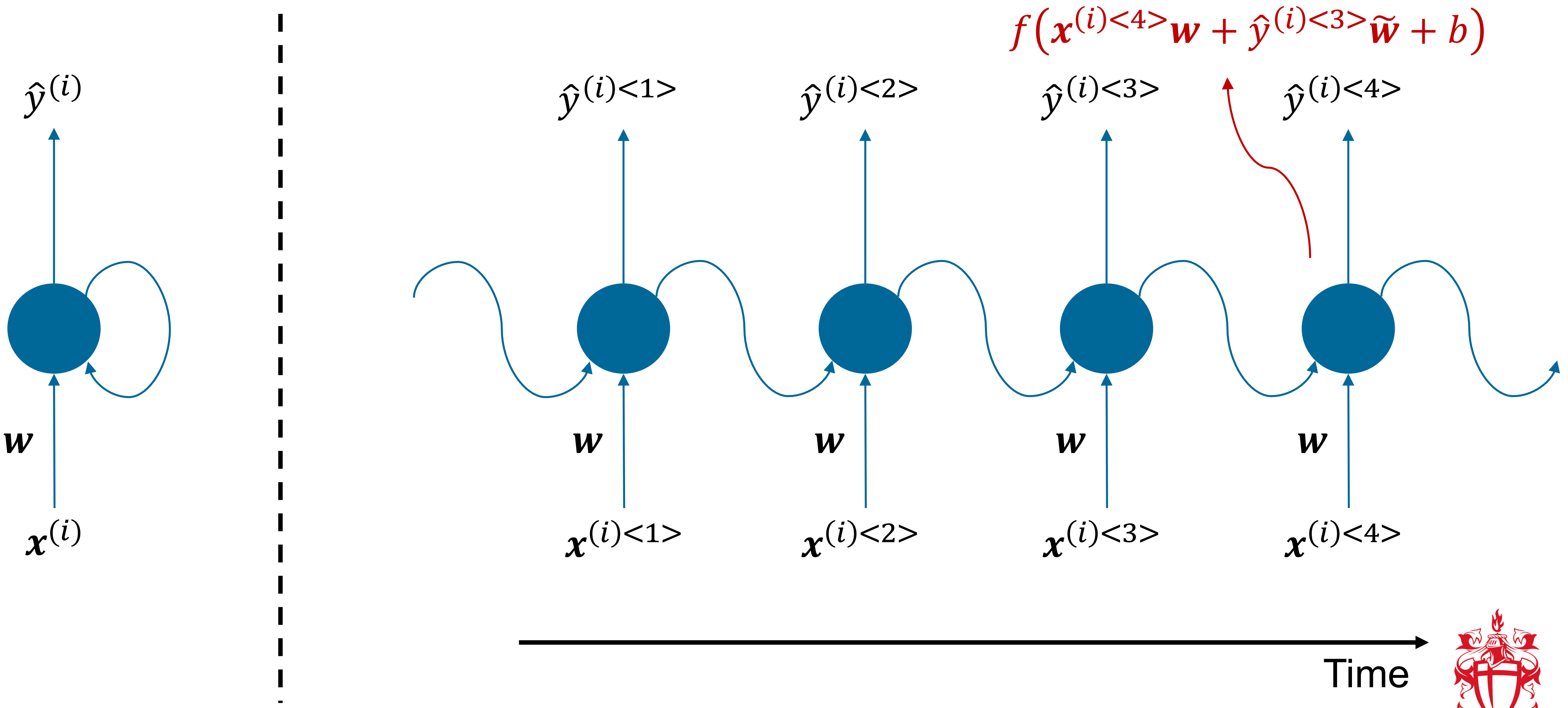
What we do instead – let's start with a single neuron



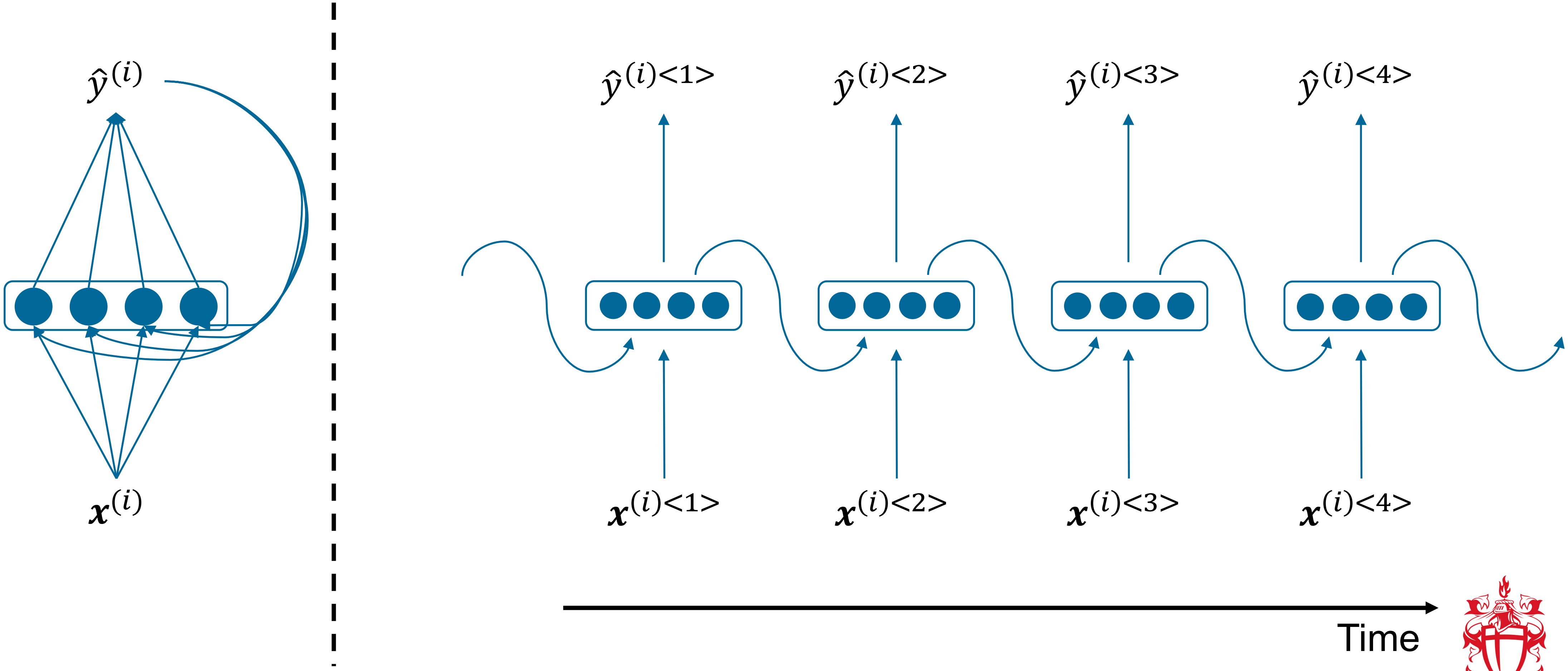
What we do instead – let's start with a single neuron



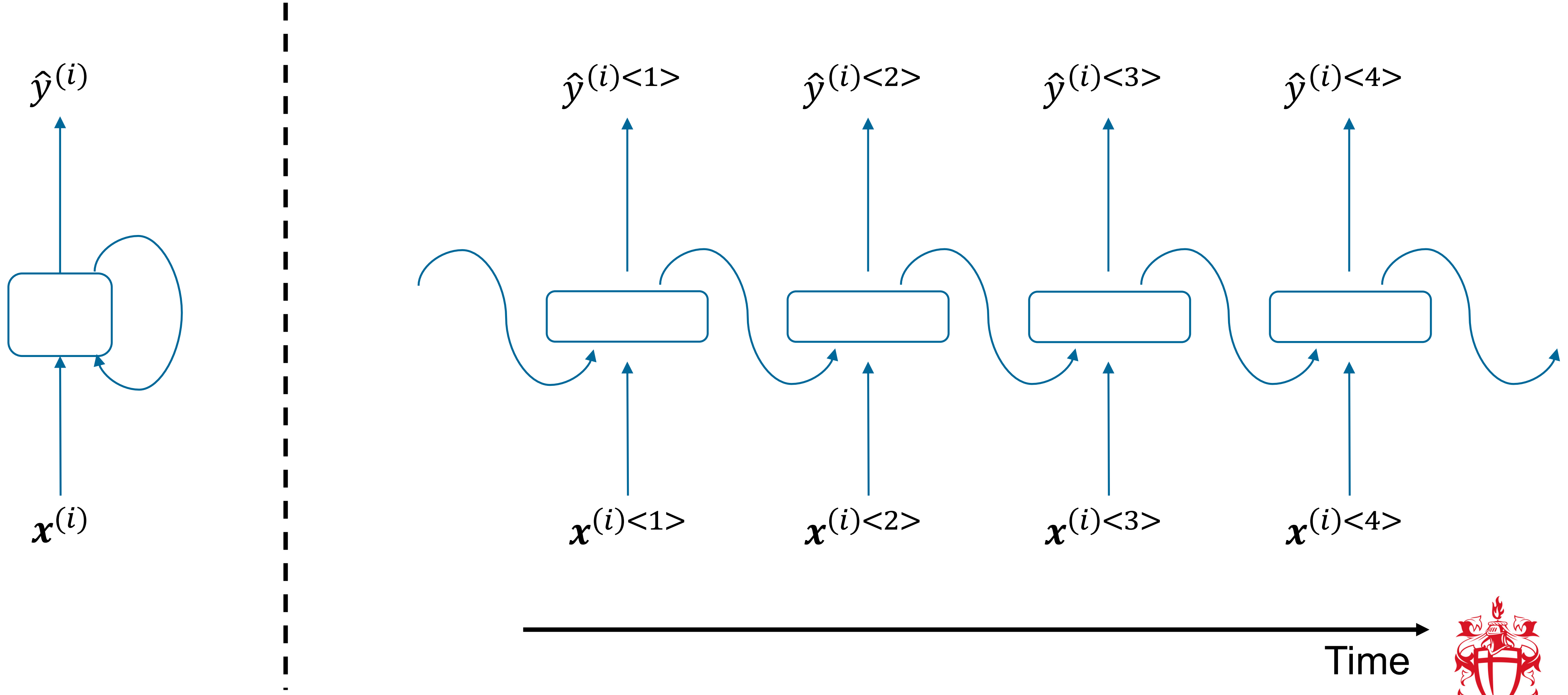
A recurrent neuron



Layers of recurrent neurons – a recurrent neural network (RNN)

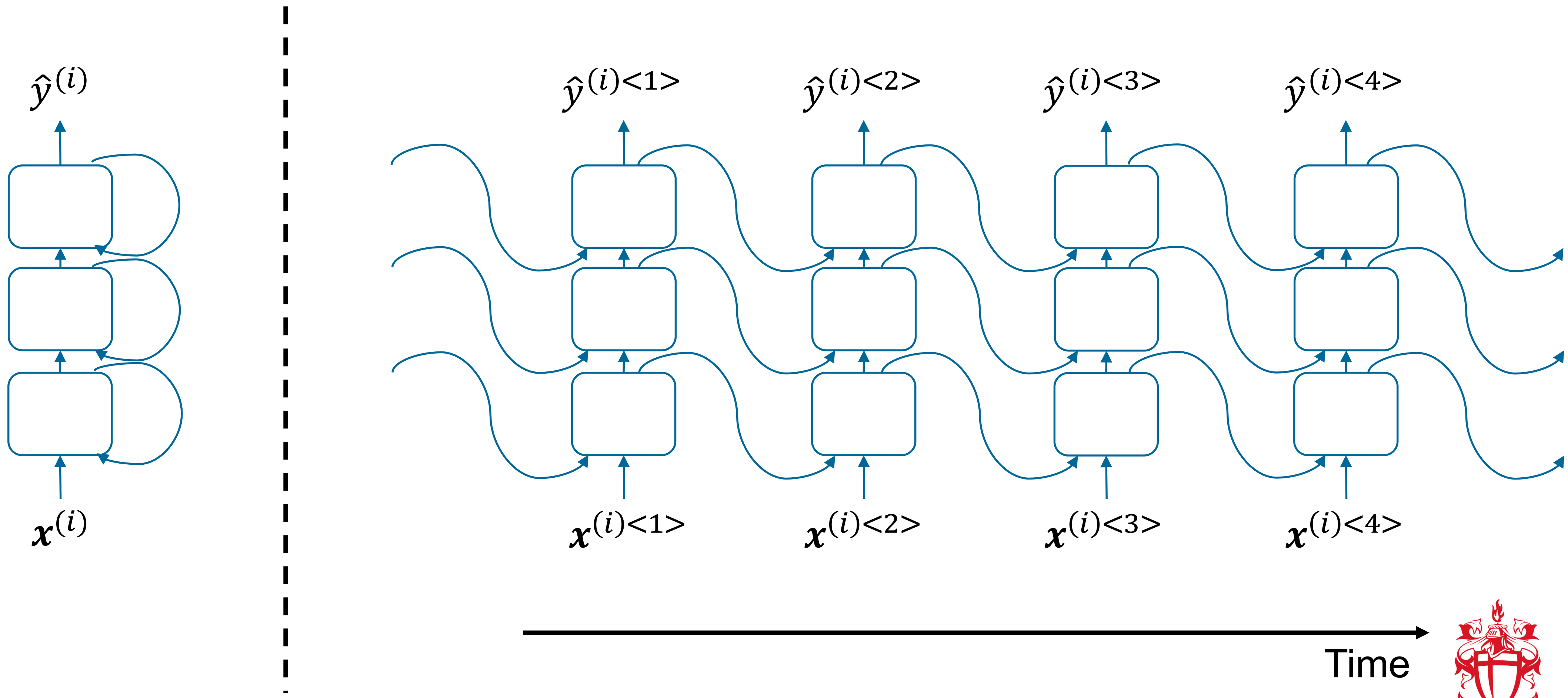


Layers of recurrent neurons – a recurrent neural network (RNN)

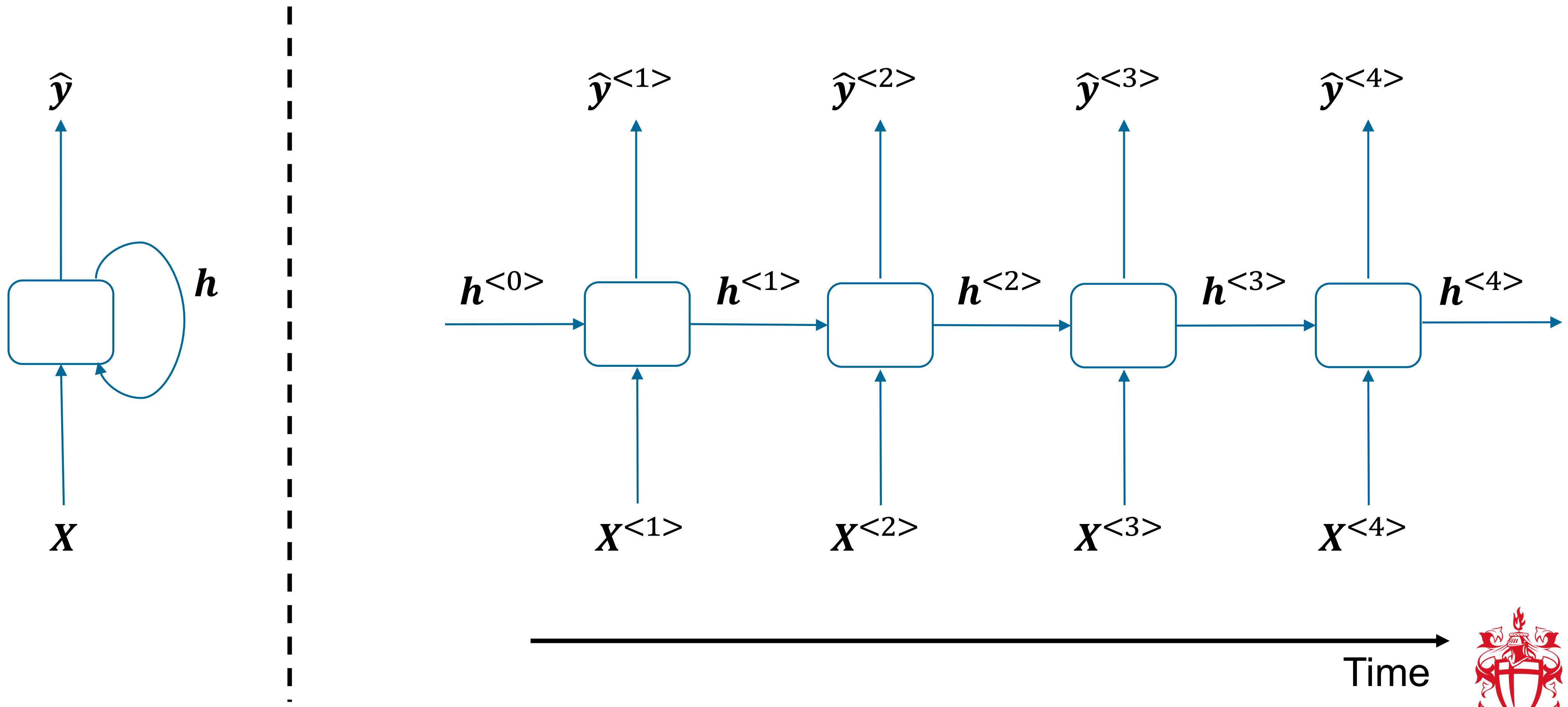


BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Deep RNNs



Representing RNNs and memory more generally



BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

A brief summary of RNNs so far

- At each time step, take the input and the “memory” (or *state*) from the previous time step to compute the output

- Use the same parameters (and, also, activation functions) across different time steps
    - Similar idea to parameter sharing in convolutional layers → we want to detect (recurring) patterns

- Usually, the loss is computed by summing up the losses on all time steps (but there are many variations)

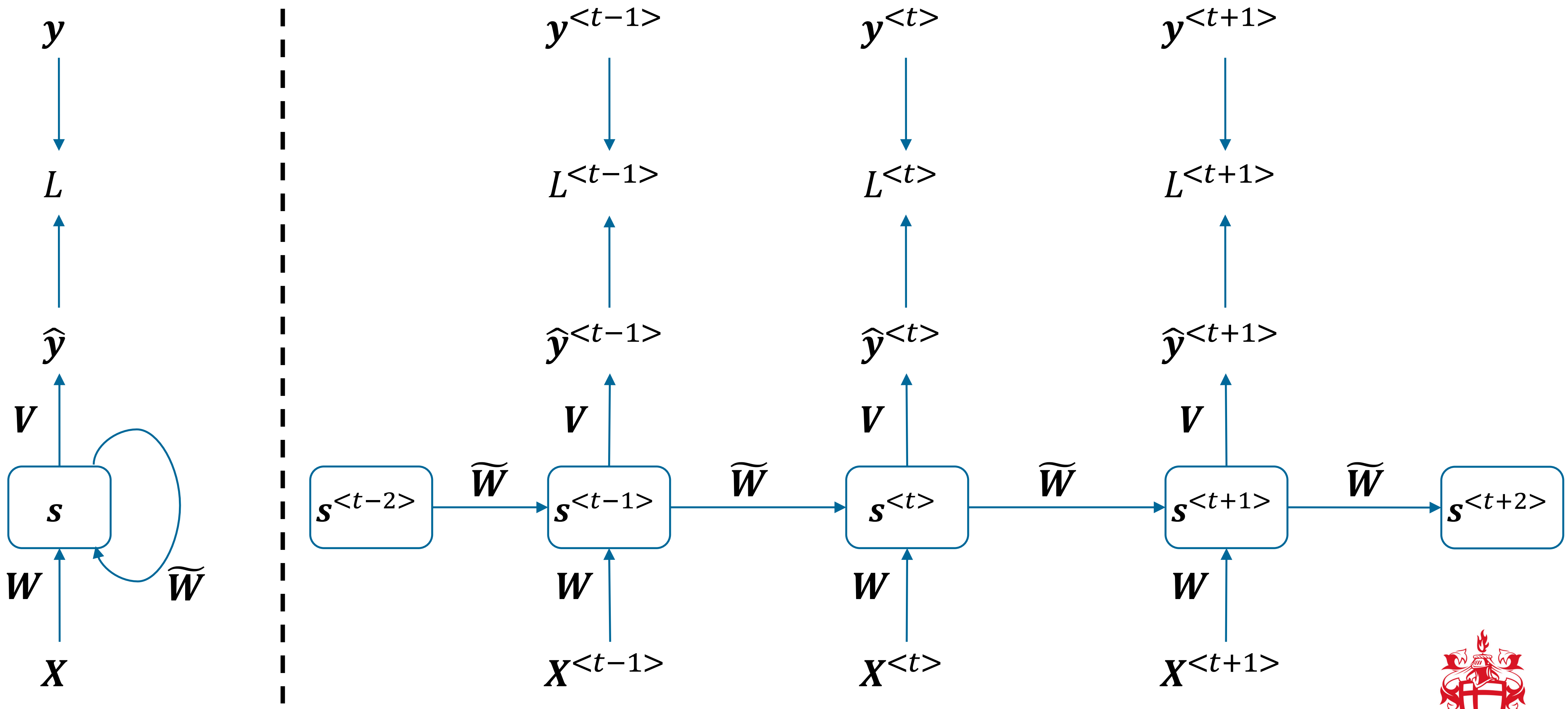
Back-propagation through time

Training an RNN

In principle, unfold the sequence to get to a computational graph, and use back-propagation

- “Back-propagation through time” algorithm (BPTT)
- Once computational graph has been established, can apply any of the known gradient-based optimization algorithms

Back-propagation through time

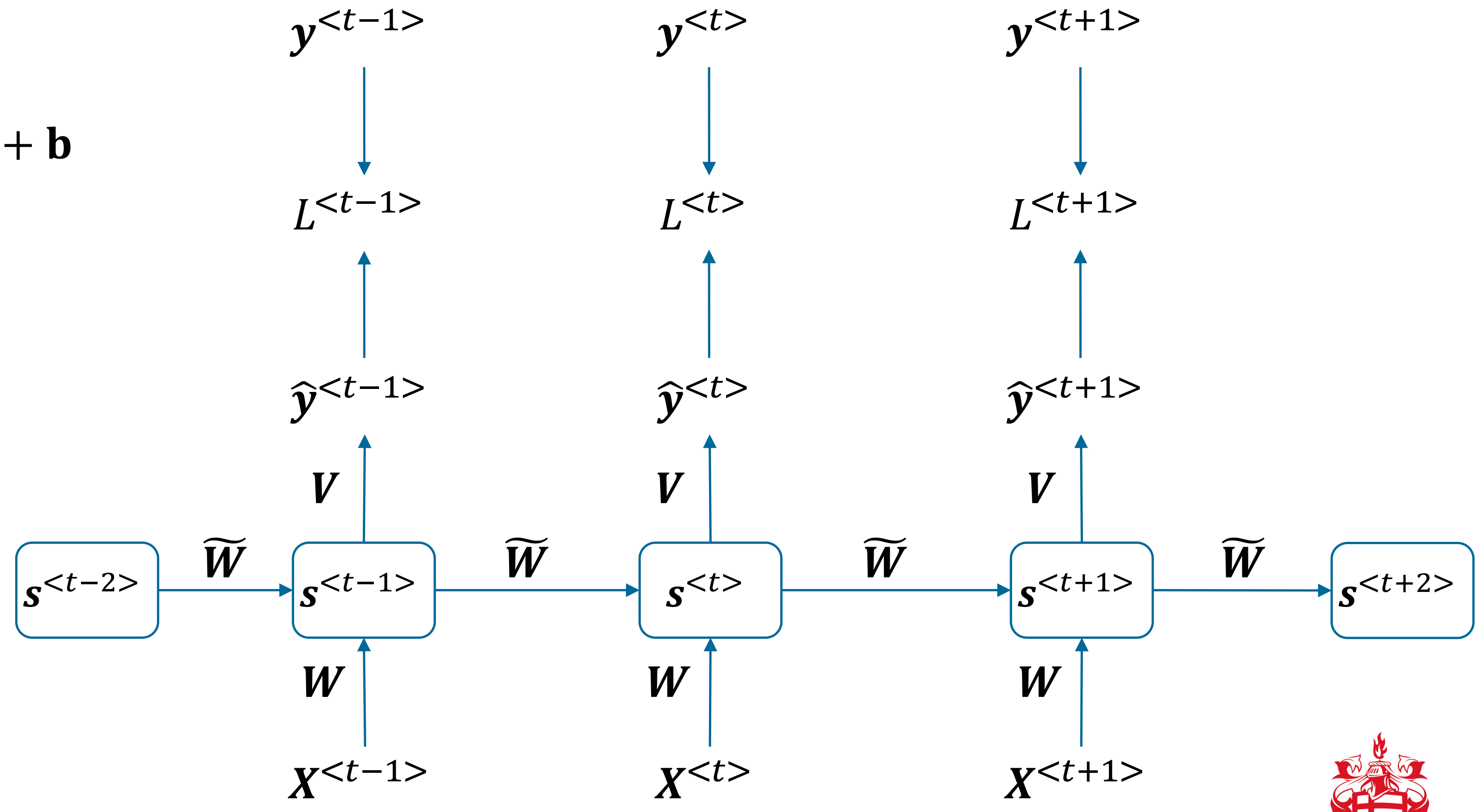


Back-propagation through time

$$\boldsymbol{a}^{<t>} = \boldsymbol{x}^{<t>}\boldsymbol{W} + \boldsymbol{s}^{<t-1>}\widetilde{\boldsymbol{W}} + \mathbf{b}$$

$$\boldsymbol{s}^{<t>} = f(\boldsymbol{a}^{<t>})$$

$$\hat{\boldsymbol{y}}^{<t>} = g(\boldsymbol{s}^{<t>}\boldsymbol{V} + \boldsymbol{c})$$

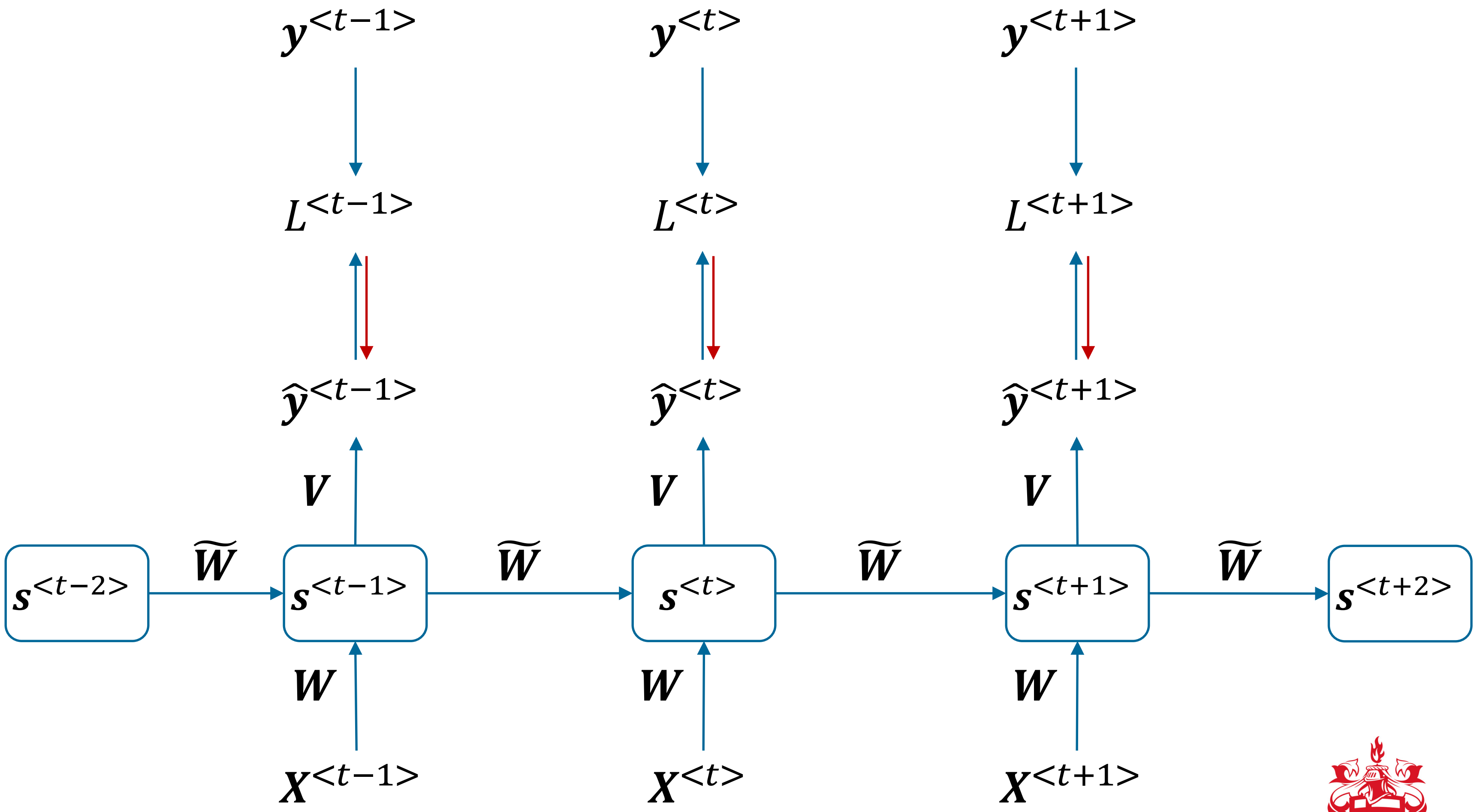


Back-propagation through time

Gradient of loss to $\hat{\boldsymbol{y}}^{<t>}$:

$$\frac{\partial L}{\partial \hat{\boldsymbol{y}}^{<t>}} = \underbrace{\frac{\partial L}{\partial L^{<t>}}} \frac{\partial L^{<t>}}{\partial \hat{\boldsymbol{y}}^{<t>}}$$

We have $L = \sum_t L^{<t>}$, so the gradient is 1

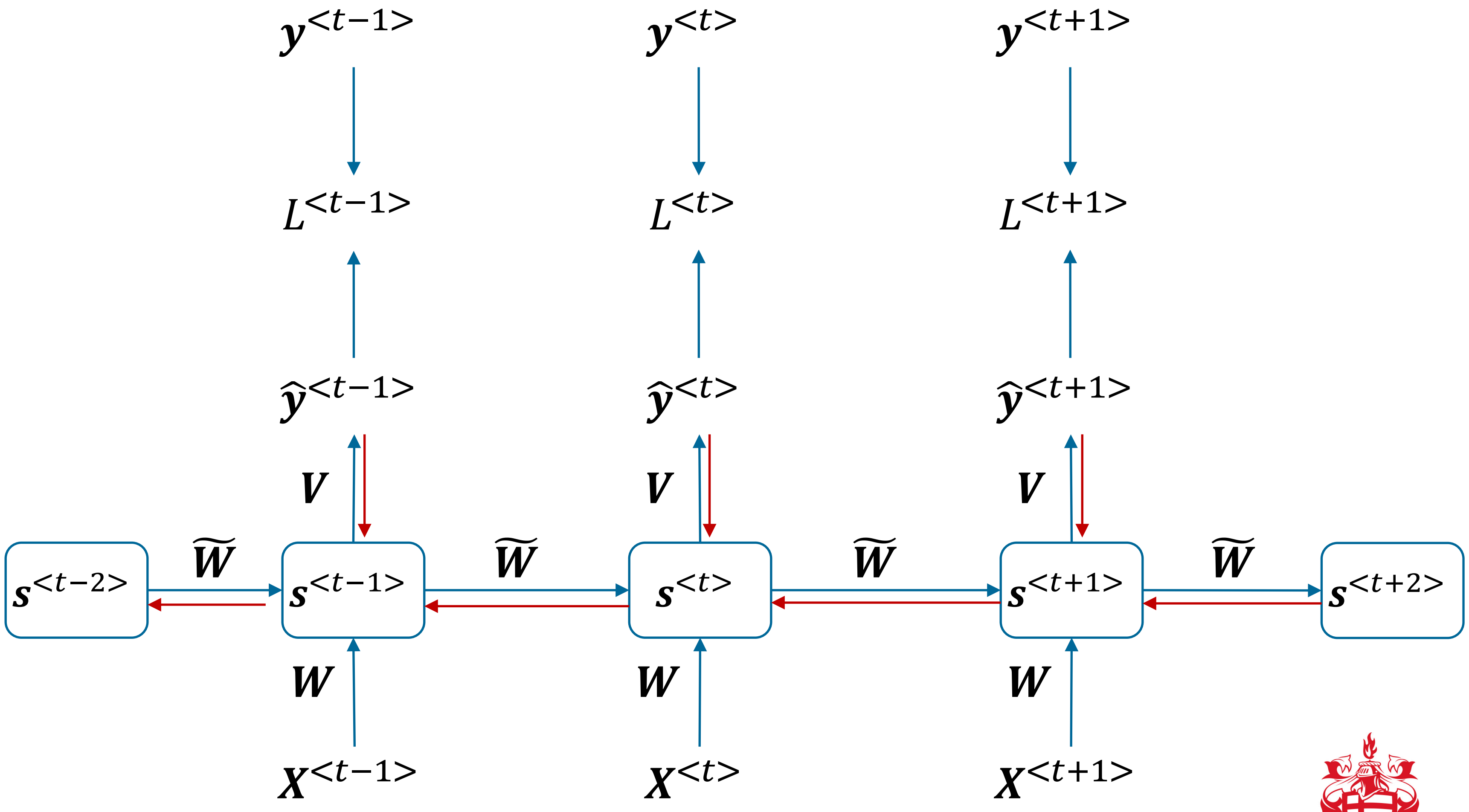


Back-propagation through time

Gradient of loss to $\boldsymbol{s}^{<t>}$:

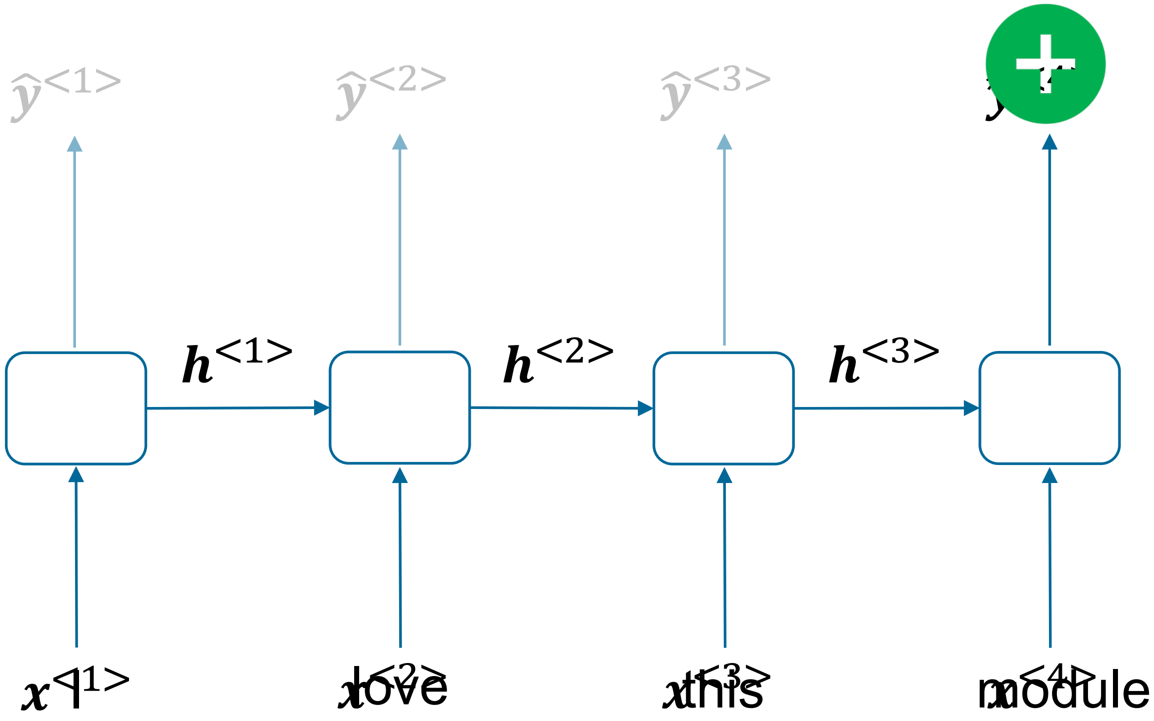
$$\frac{\partial L}{\partial \boldsymbol{s}^{<t>}} = \frac{\partial L}{\partial \widehat{\boldsymbol{y}}^{<t>}} \frac{\partial \widehat{\boldsymbol{y}}^{<t>}}{\partial \boldsymbol{s}^{<t>}} \; ?$$

No, because future periods (and their losses) also depend on $s^{<t>}$!



RNN variants and their applications

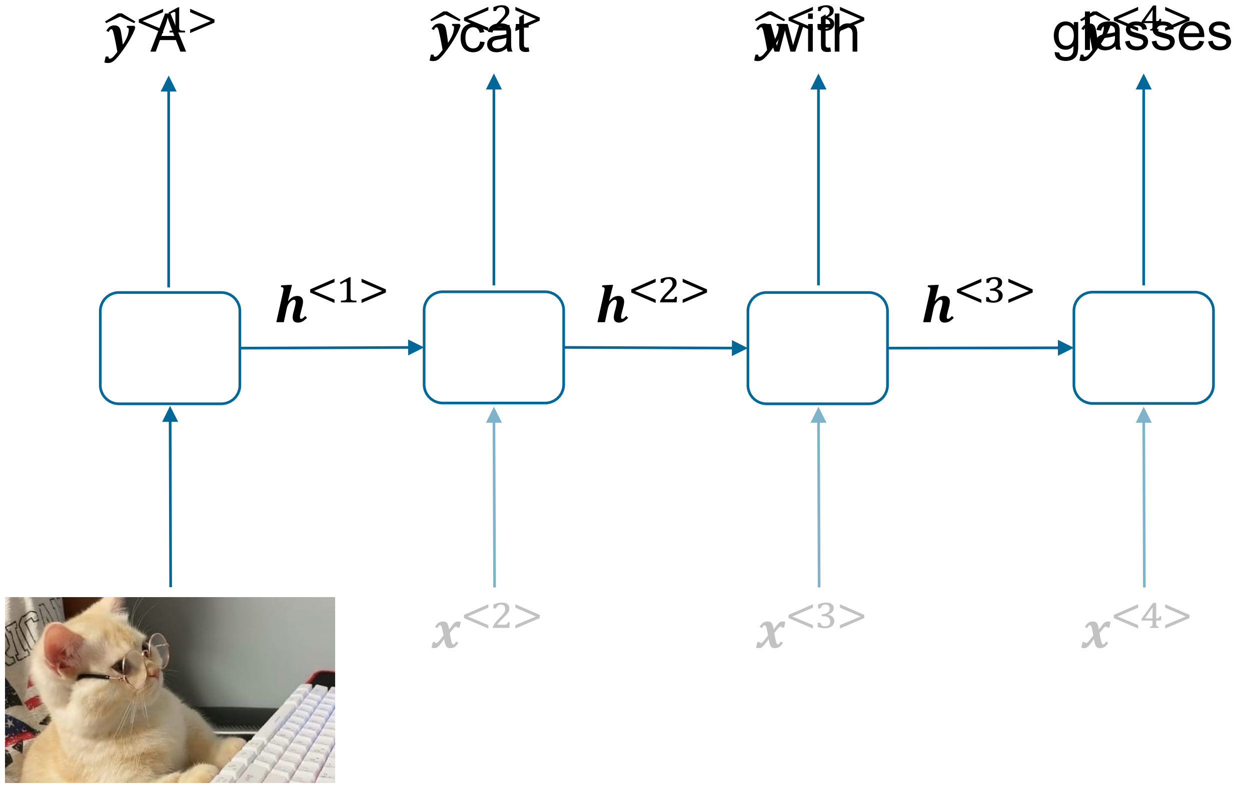
Sequence-to-vector networks



For example:

- Video activity recognition
- DNA sequence probing
- Sentiment classification

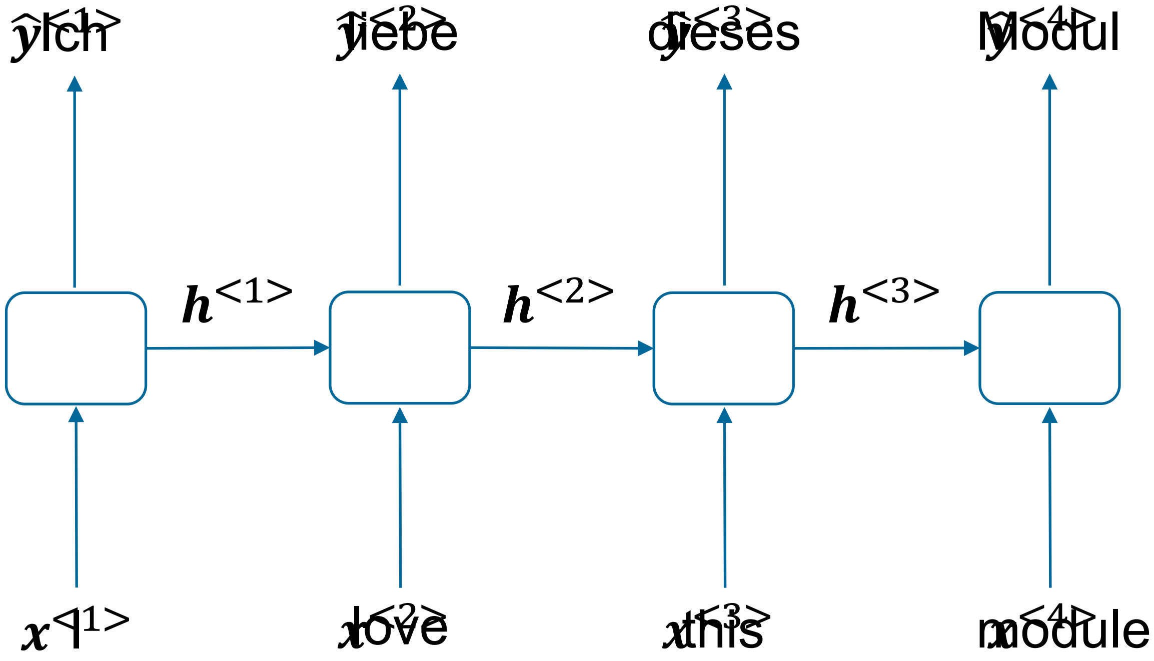
Vector-to-sequence networks



For example:

- Text generation
- Music generation
- Image captions

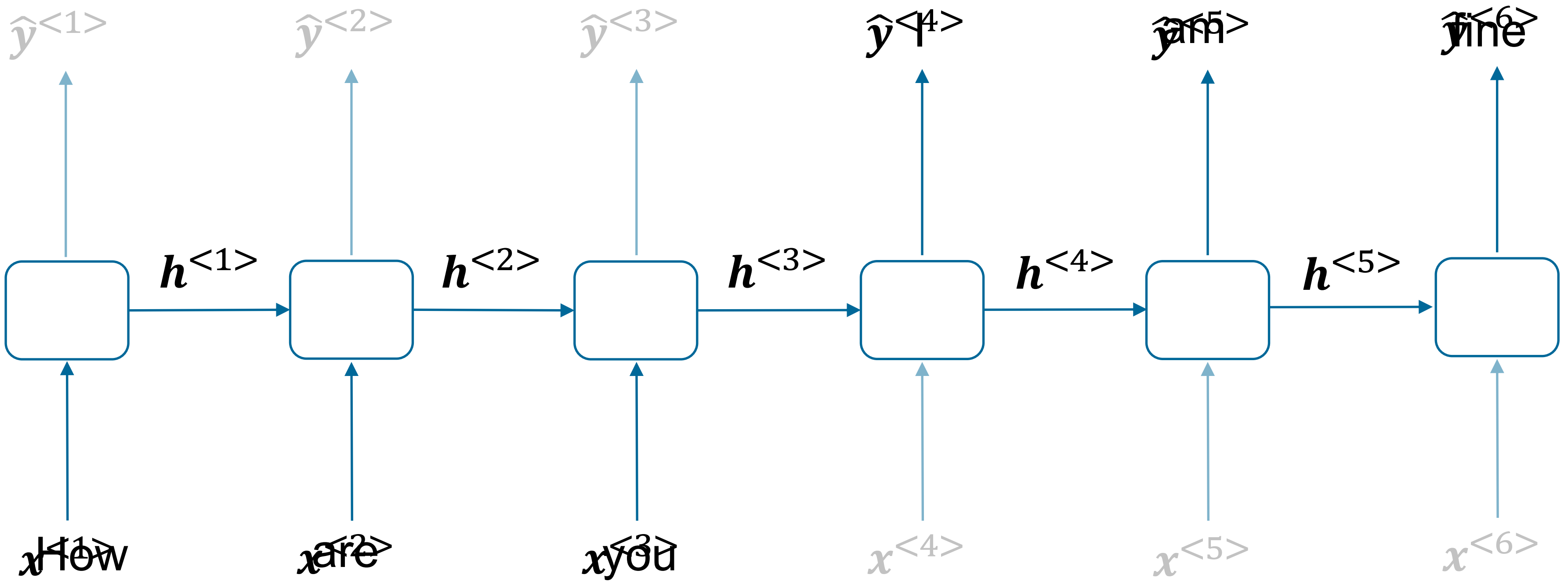
Sequence-to-sequence networks



For example:

- Speech recognition
- Price predictions
- Translations

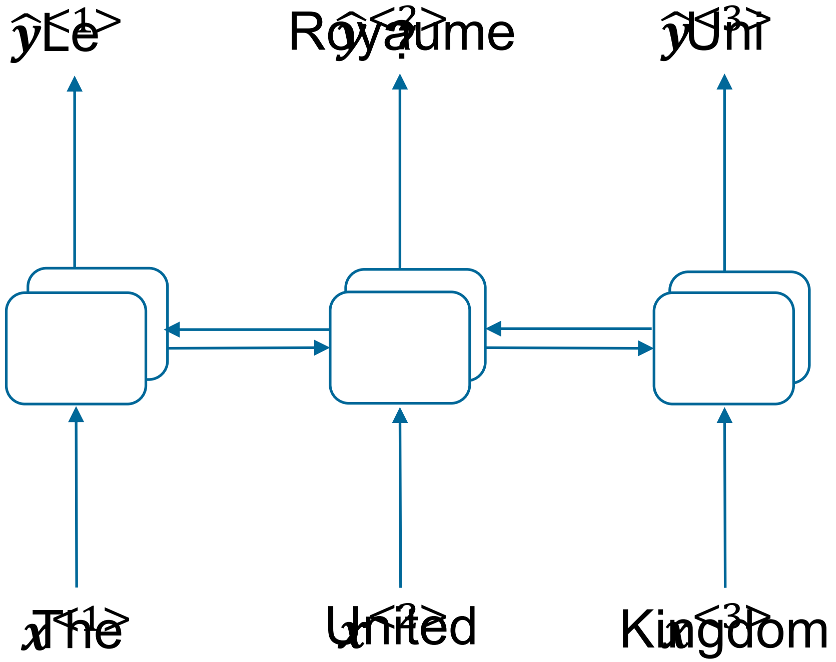
Encoder-decoder networks



For example:

- Translations
- Dialogue

Bidirectional RNNs – looking into the future



For example:

- All sorts of NLP
- Also, in combination with the previous

The issues with training RNNs

Problem 1 – vanishing and exploding gradients

- In principle, same as with other networks

- Before, we mostly focused on vanishing gradients
  → use of non-saturating activation functions such as ReLU

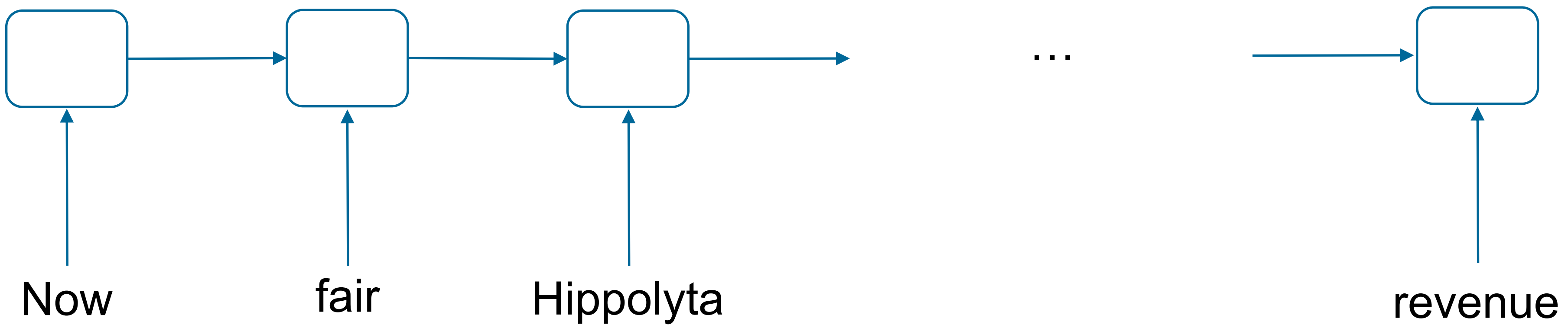
- With RNNs, exploding gradients become more of a problem
  - Same weights used for different time steps can lead to self-reinforcing increases of gradients

  → We frequently use saturating activation functions, such as tanh, or other methods such as gradient clipping

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Problem 2 – memory issues

- Vanishing gradients are still a problem (sometimes even more so than in other networks):



- This is essentially a very very deep neural network!
    - Some information is lost at each time step

- After just a few time steps, there is virtually no more information about the first input

When memory loss can be a problem

The BA students, which had been working for days on end, was finally done with their project.

When memory loss can be a problem

The BA students, which had been working for days on end, was finally done with their project.

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

See you in class!

Sources

- DeepLearning.AI, n.d.: deeplearning.ai
- Garnelo, 2020, Lecture 6: Sequences and Recurrent Networks: https://storage.googleapis.com/deepmind-media/UCLxDeepMind_2020/L6%20-%20UCLxDeepMind%20DL2020.pdf
- Géron, 2019, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow
- Goodfellow, Bengio, Courville, 2016, The Deep Learning Book: http://www.deeplearningbook.org
- Liang, 2016, Introduction to Deep Learning: https://www.cs.princeton.edu/courses/archive/spring16/cos495/
- Soleimany, 2022, Deep Sequence Modeling: http://introtodeeplearning.com/slides/6S191_MIT_DeepLearning_L2.pdf