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 and the difficulty of working with them using the neural network architectures we have learned about so far
- Develop the knowledge to use basic RNNs in practice, as well as critical extensions

How will we do this?

- We then introduce sequence data and its relevance to machine learning tasks
- We build up the concept of recurrence underlying RNNs
- We consider limitations of standard RNNs and introduce extensions that allow for "long-term memory"

But first: Data handling with TensorFlow

The data problem with deep learning

Complex data preprocessing

Instead of a "normal" dataset, we often work with TensorFlow's Data API

The TensorFlow Data API

- Go through part 1 of the notebook “*ADL_Week 9_Recurrent Neural Networks.ipynb*”
- We introduce some of the key functionalities of the TensorFlow Data API
- This is useful for models in general, but it is particularly important for RNNs, since we often need to do a lot of data manipulation

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

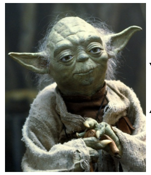
Working with sequences

"Why do we care about sequences?"

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

$\neq$

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

(unless you are )

Sequences

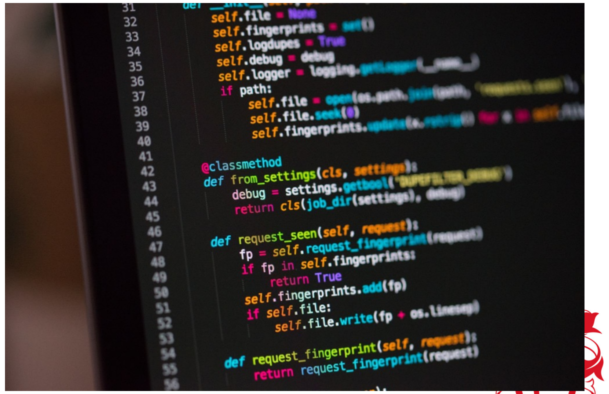
Sequences are collections of multiple elements (i.e., data points), where:

- The order matters
- Elements may be repeated
- The length is variable (and lengths of inputs and outputs don’t have to match)

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Sequences in real life

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

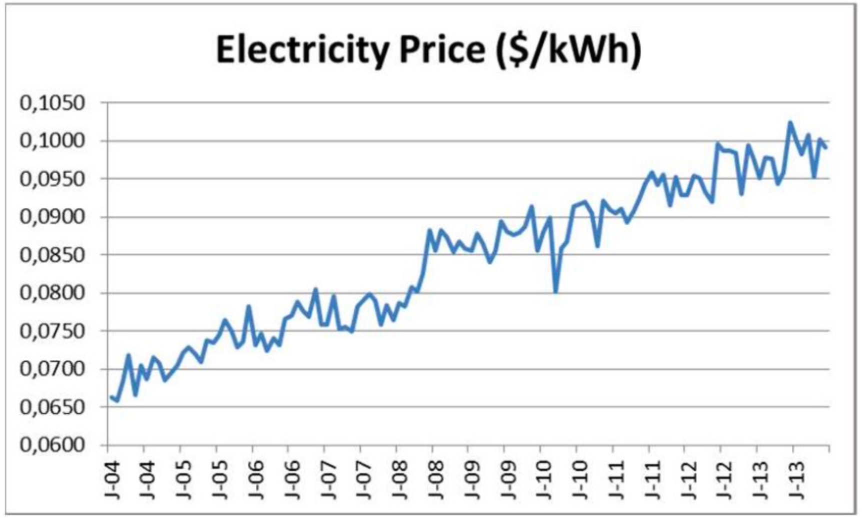


BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Remember time series?

- Say you want to pitch a solar power system to a bank to install on top of its branches

- You will certainly need to compute the NPV. But for that, you need long-term electricity price information (360 months into the future)

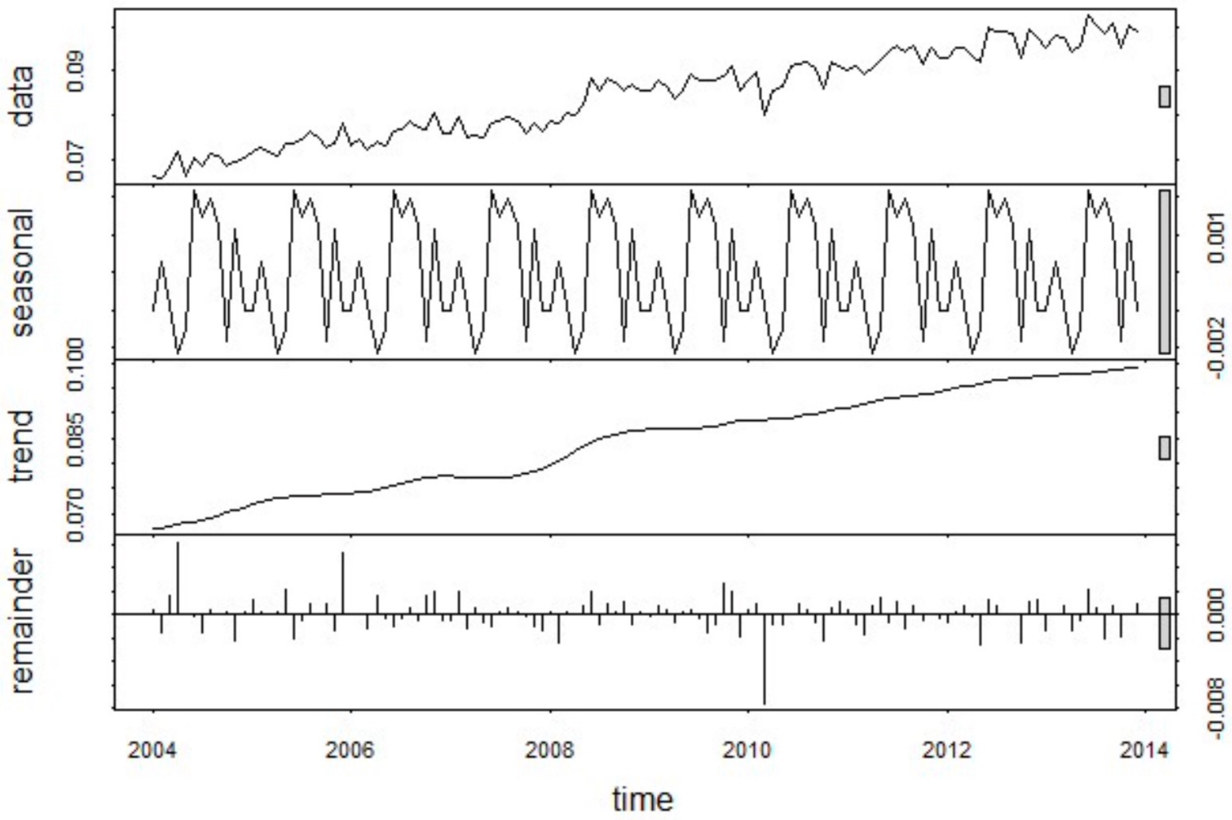


Sheri Lucas

Time series = series of data points indexed in chronological order

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Understanding time series: level, noise, trend, and seasonality

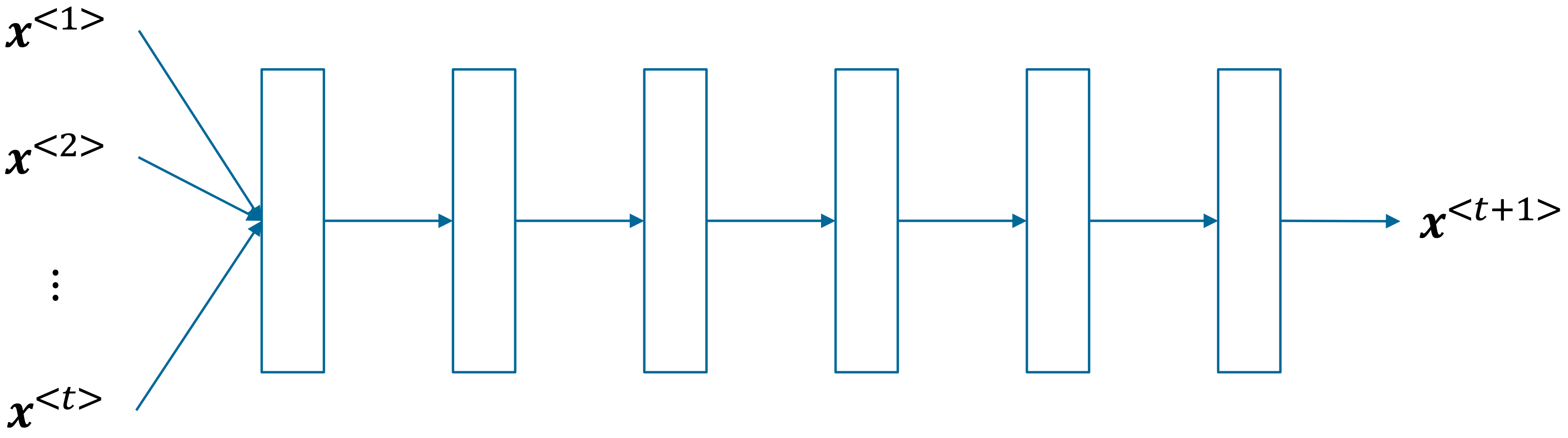


- **Level** = value of the last datapoint = starting value before trend or seasonal adjustment are added
- **Seasonality** = repetitive “short”–term pattern [seasonal indices vs smooth seasonality]
- **Trend** = long-term movement of the data. Not to confuse with **cycles**: up- or down-movements with irregular/unpredictable turning points
- **Noise / error** = remaining / random variation in the data after accounting for trend and seasonality/ies

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

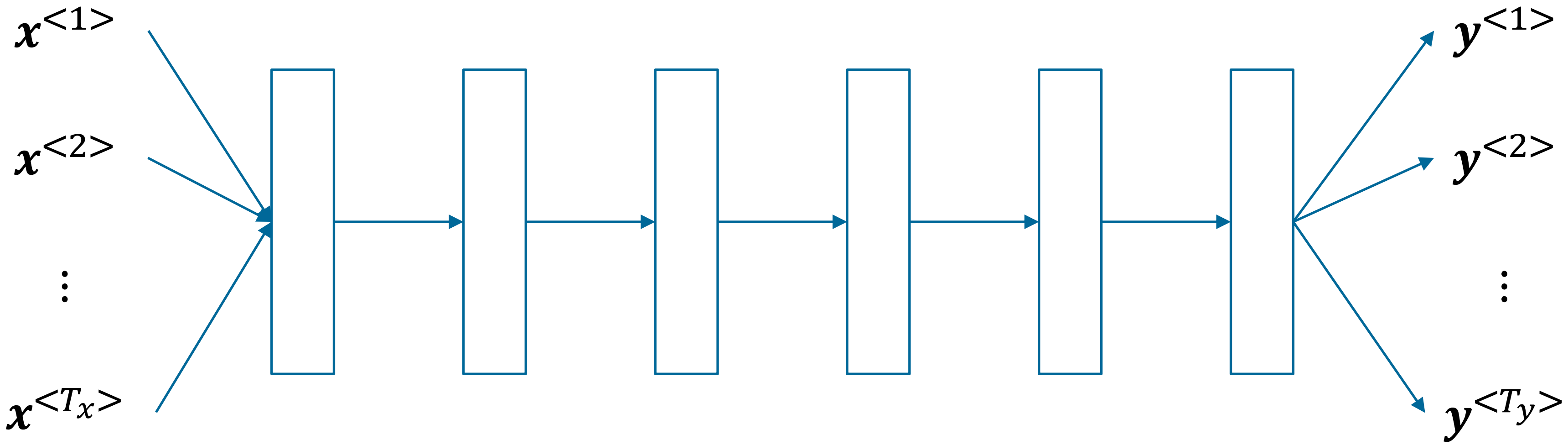
Initial approaches

A basic model – our comparison point



Go through Code Parts 2.1-2.2

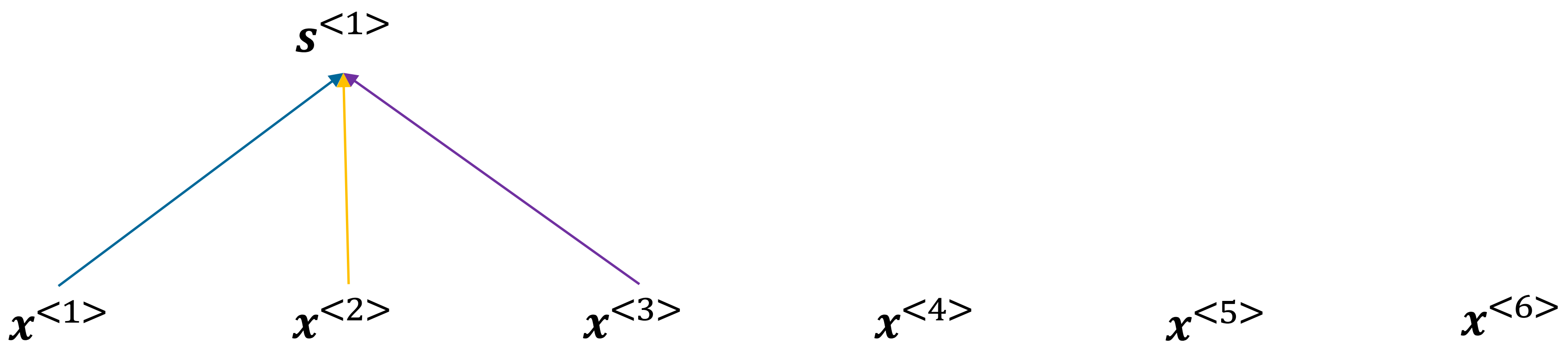
A more general version



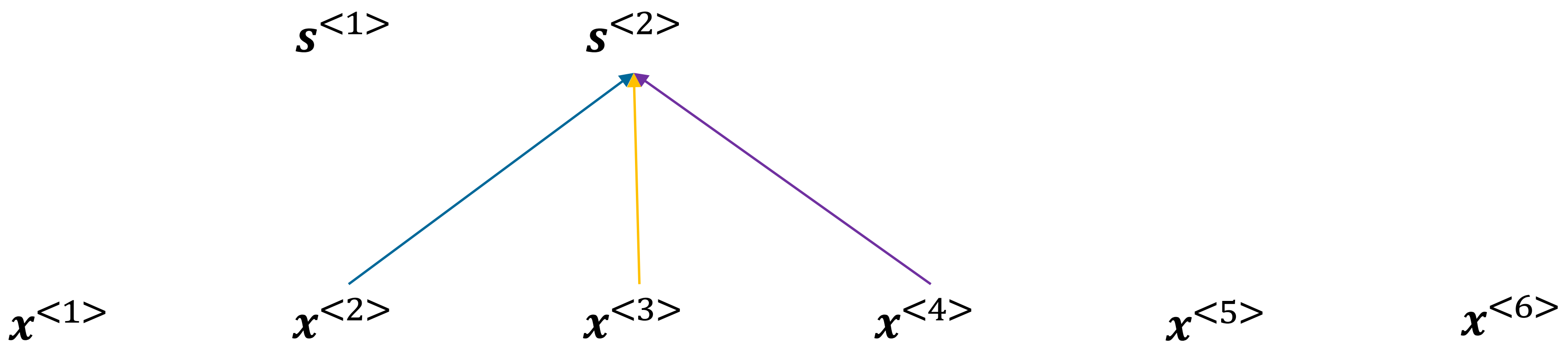
Issues:

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

1D convolutional layers

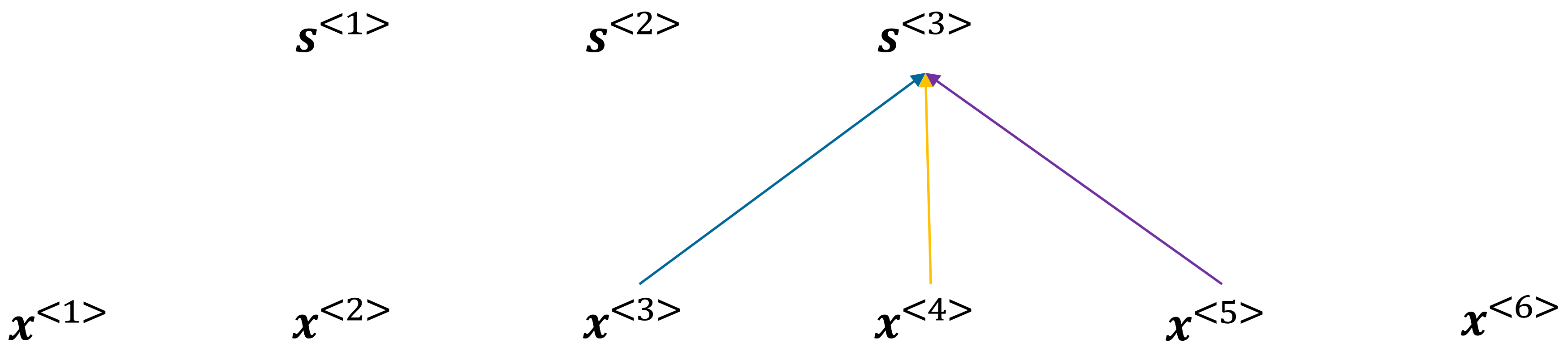


1D convolutional layers

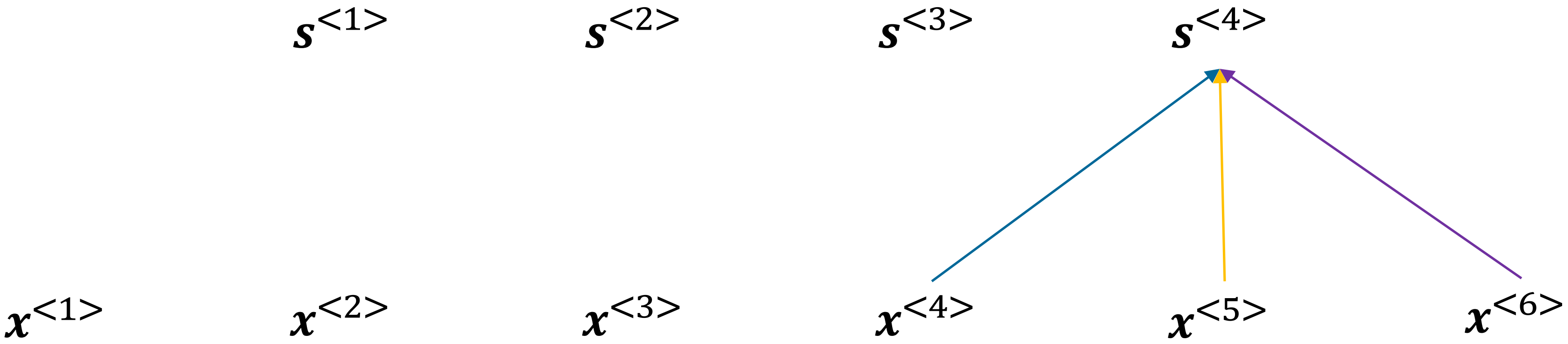


BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

1D convolutional layers



1D convolutional layers



1D convolutional layers – a few considerations

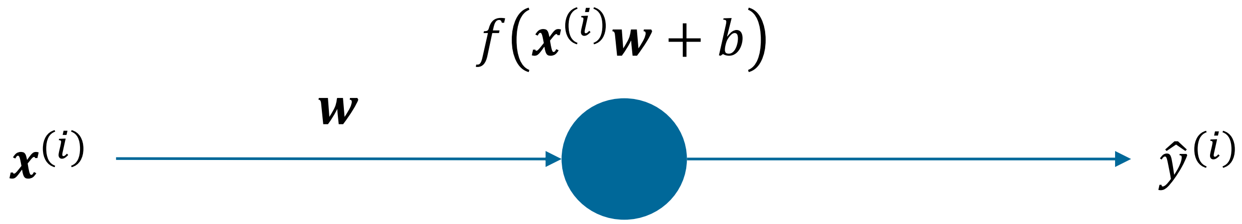
- 1D-convolutions can help summarize lengthy sequences and speed things up

- However, due to parameter sharing, 1D-convolutions assume "translation invariance"
  - That is, the content is independent of the location
  - With weather data:
    - Patterns in the morning may be different from those in the evening
    - Translation invariance only works at specific scale of days, years, etc.

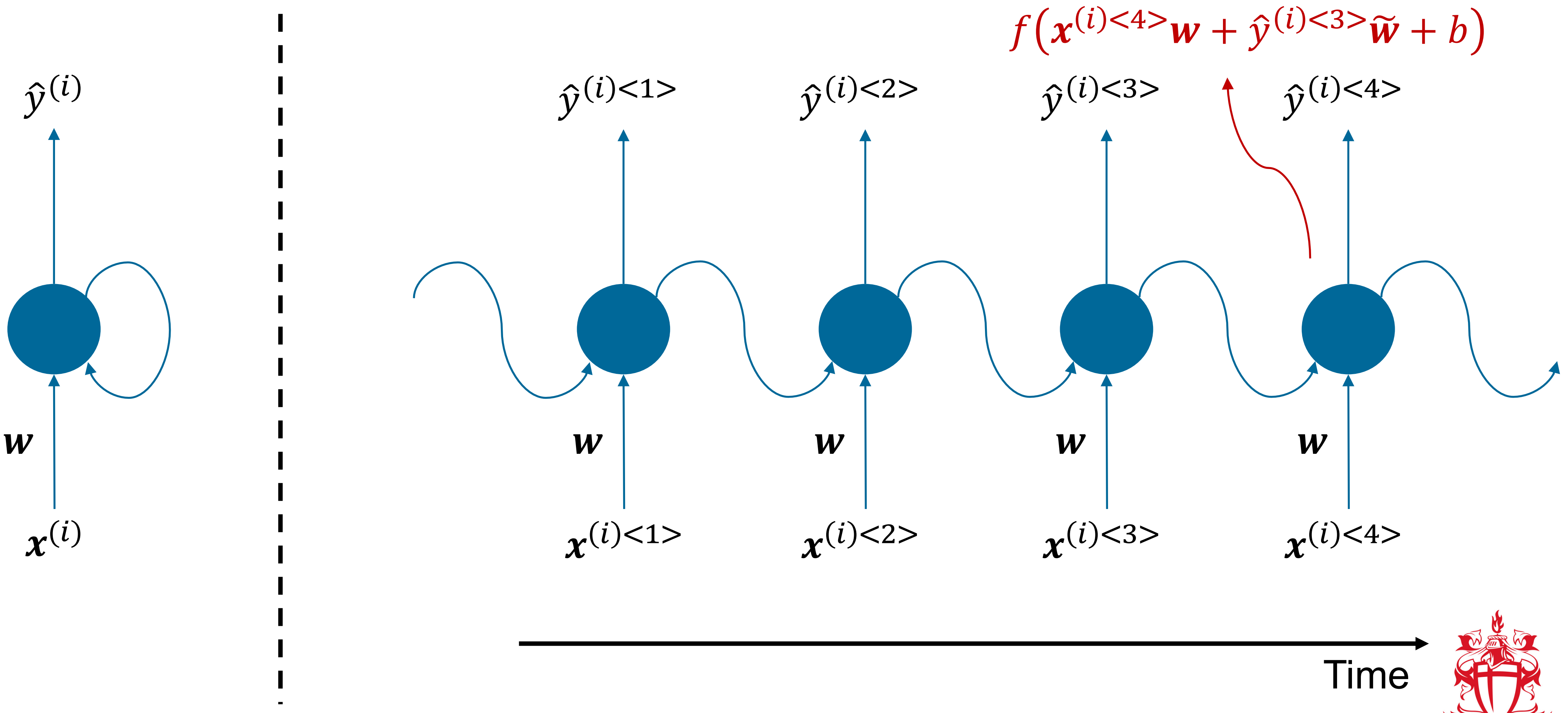
- Also, in many cases, recent data is more relevant than older data
  - 1D-convolutions don't leverage this specifically
  - Pooling actually destroys a lot of order information

Recurrence in neural networks

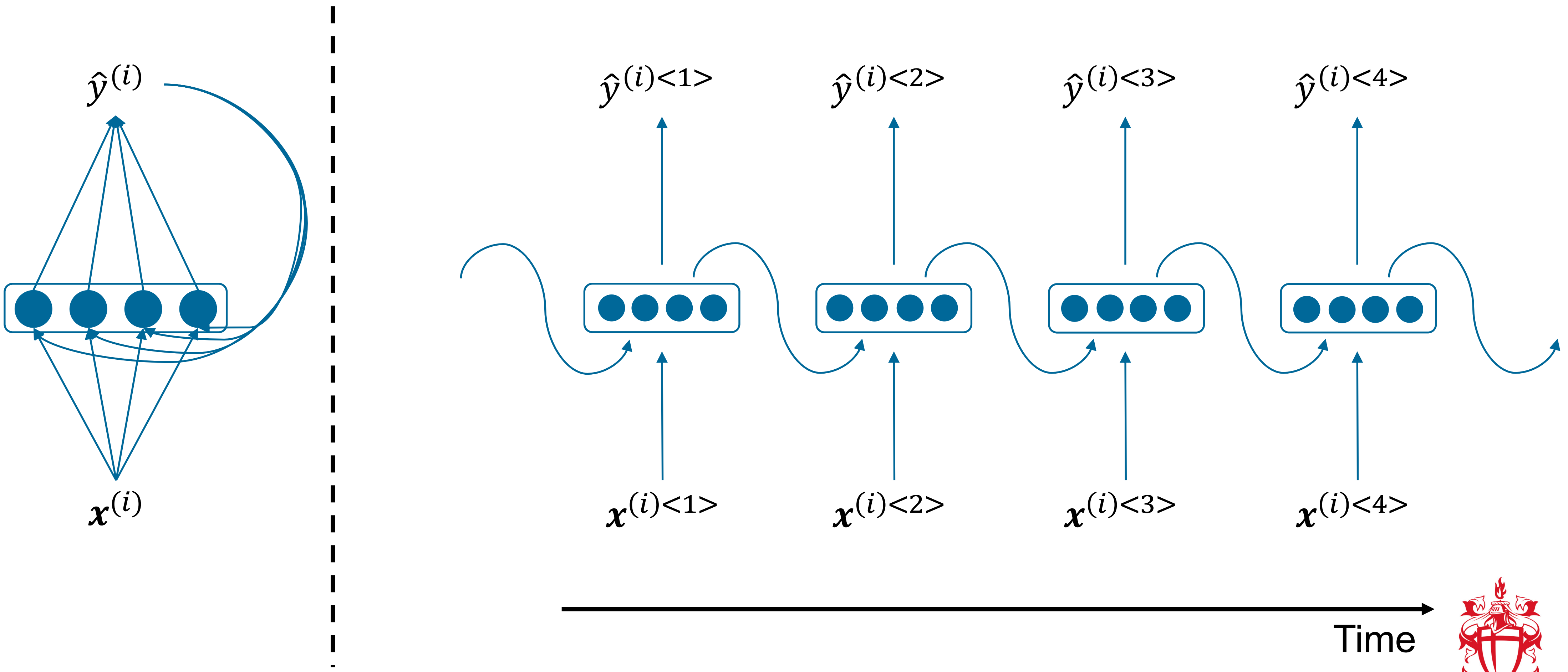
What we do instead – let's start with a single neuron



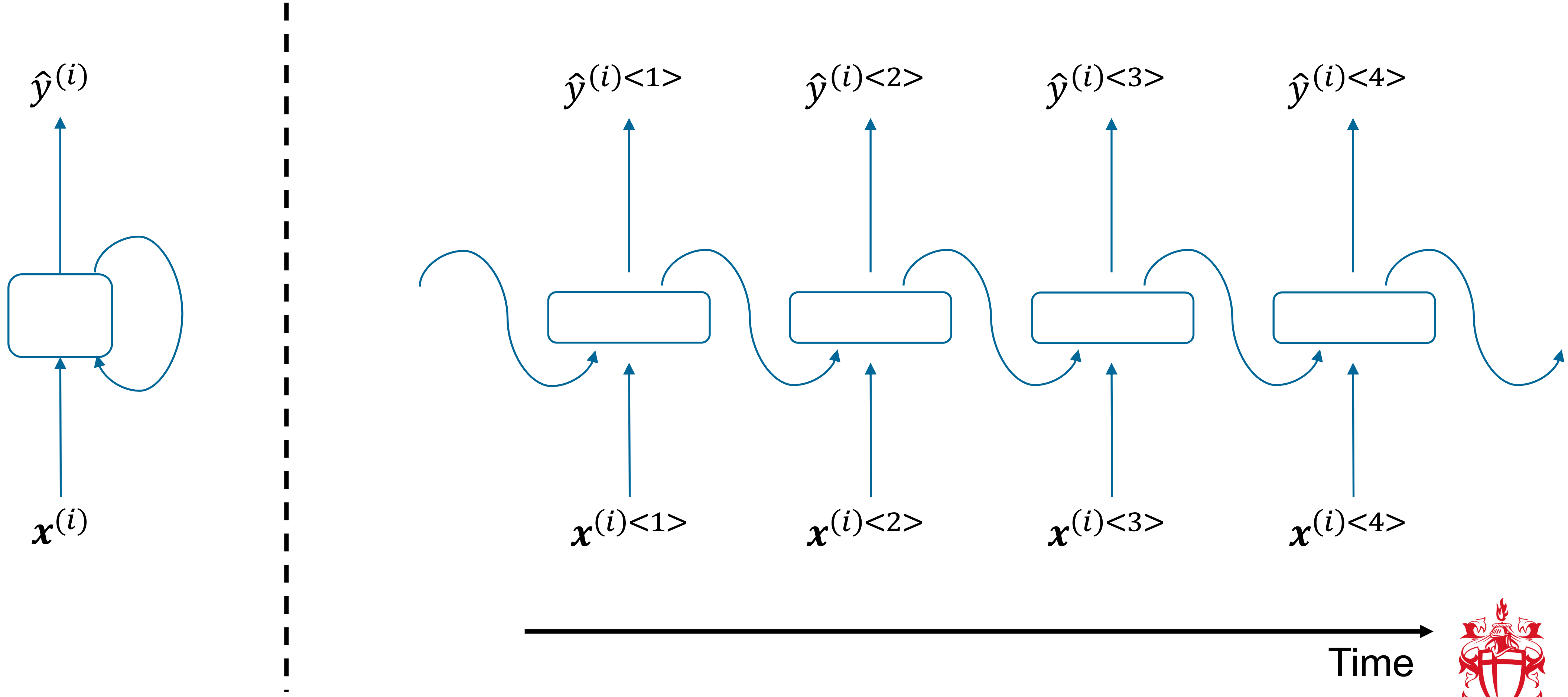
A recurrent neuron



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

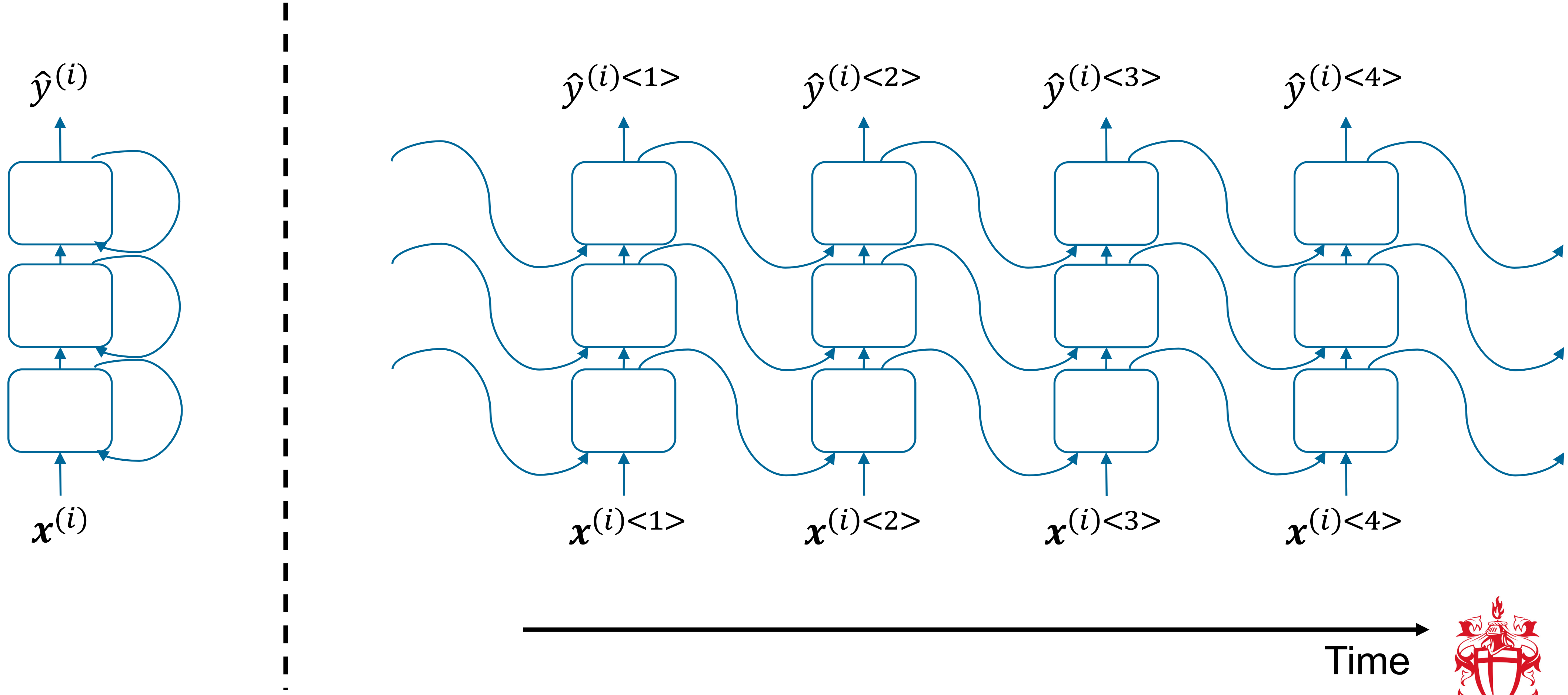


Layers of recurrent neurons – a recurrent neural network (RNN)

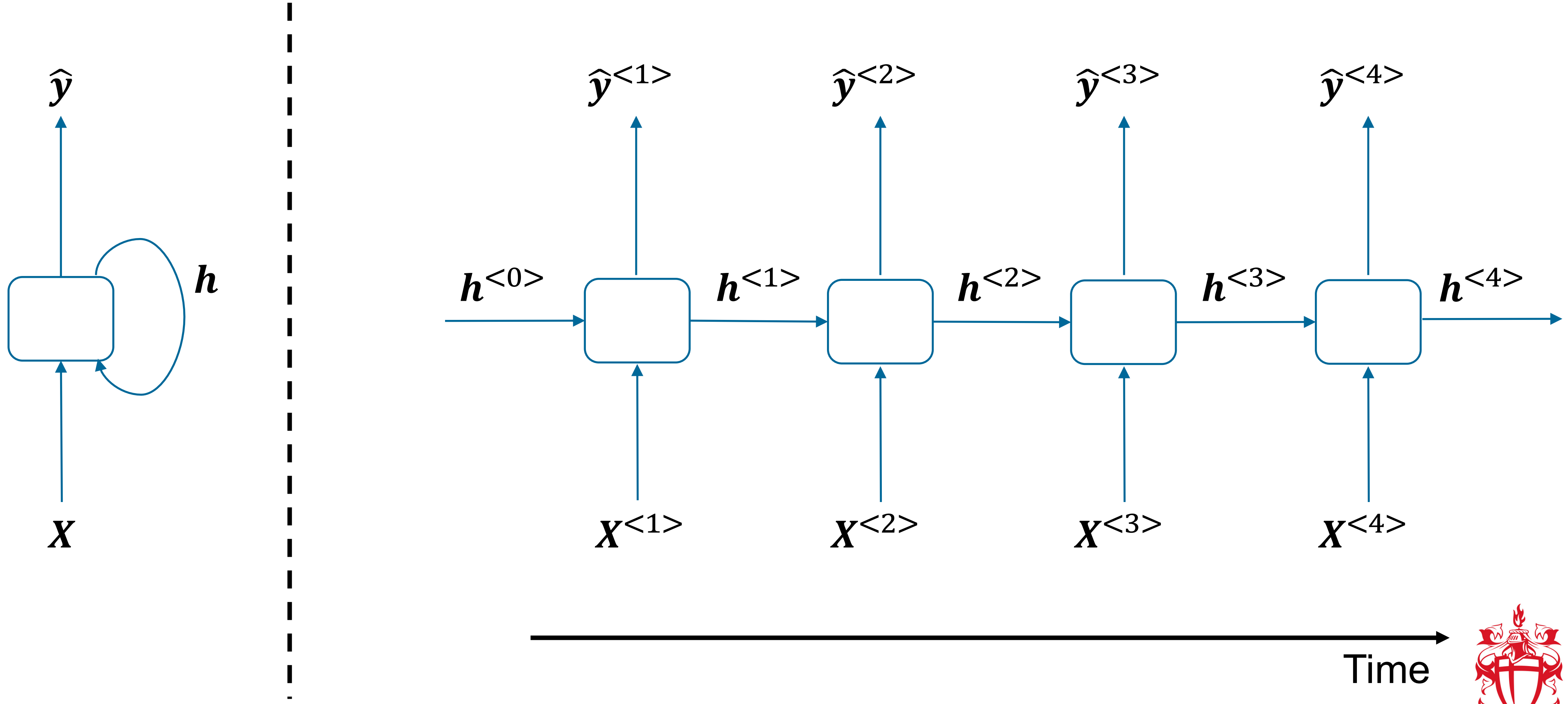


BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Deep RNNs



Representing RNNs and memory more generally



Let’s see recurrent neurons and layers in TensorFlow

The issues with training RNNs

Problem 1 – vanishing and exploding gradients

- In principle, same as with other networks

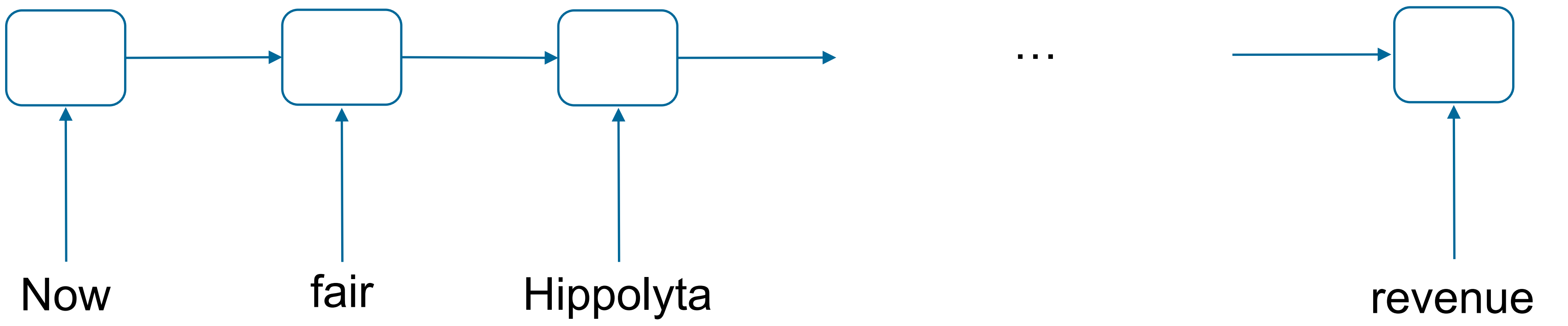
- Before, we mostly focused on vanishing gradients
  → use of non-saturating activation functions such as ReLU

- 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

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 projects.

When memory loss can be a problem

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

Adding long-term memory

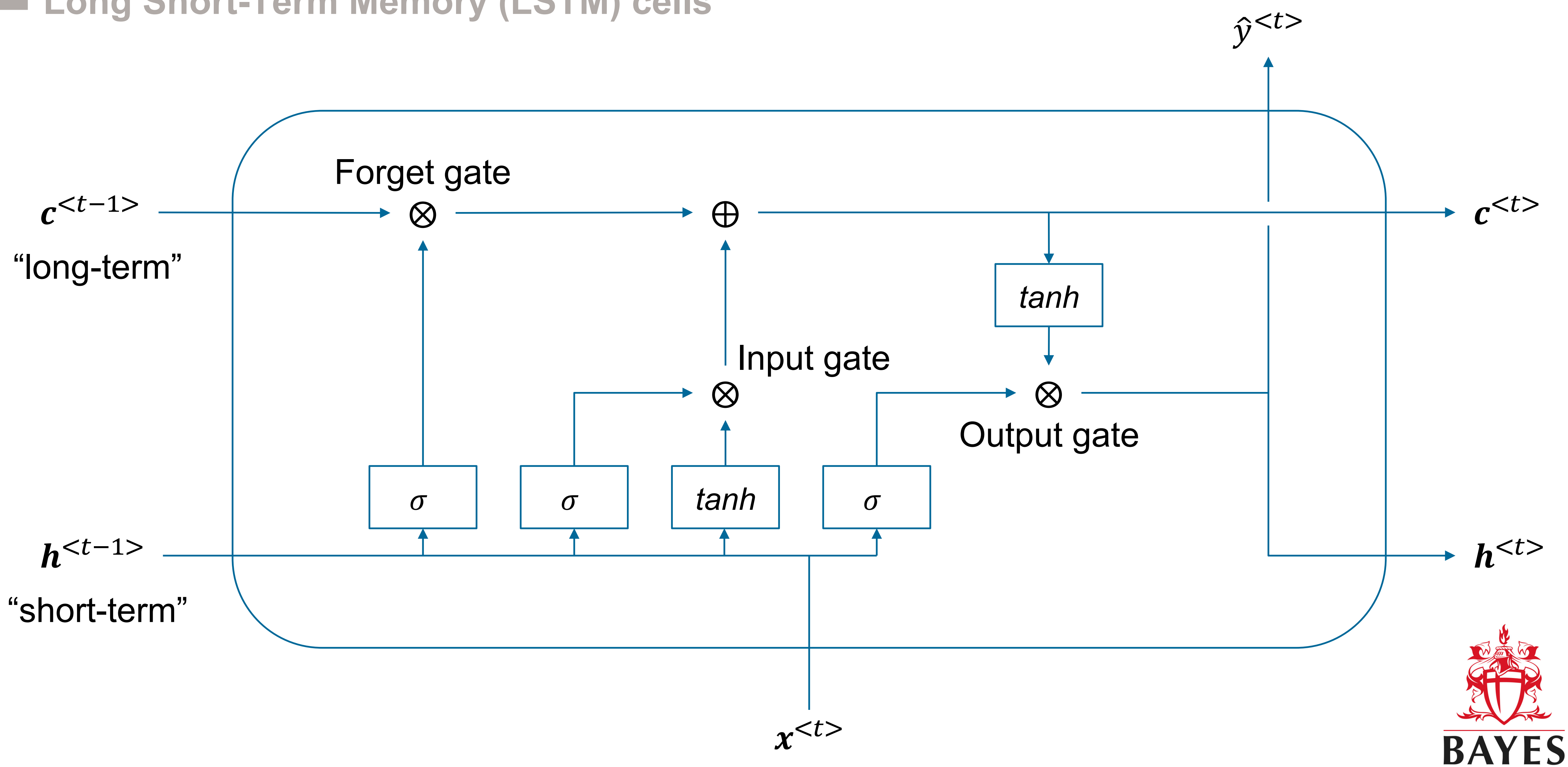
General idea

- Knowledge acquired so far as a state that is managed

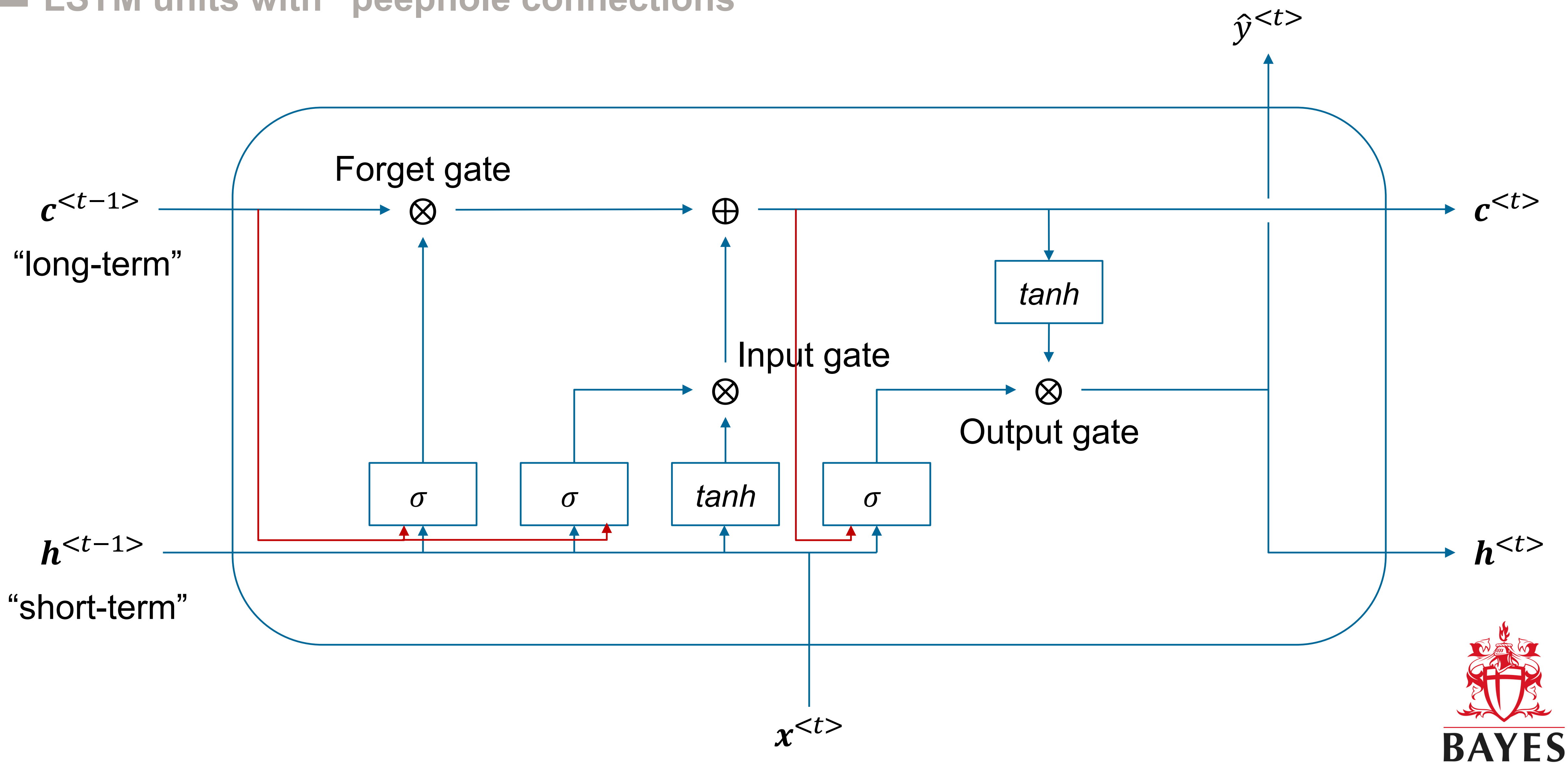
- Use “gates” to add or remove information in each recurrent unit
  - Remove information that is no longer relevant
  - Selectively add information from current input that will be relevant later down the line

- Output based on the state and the input

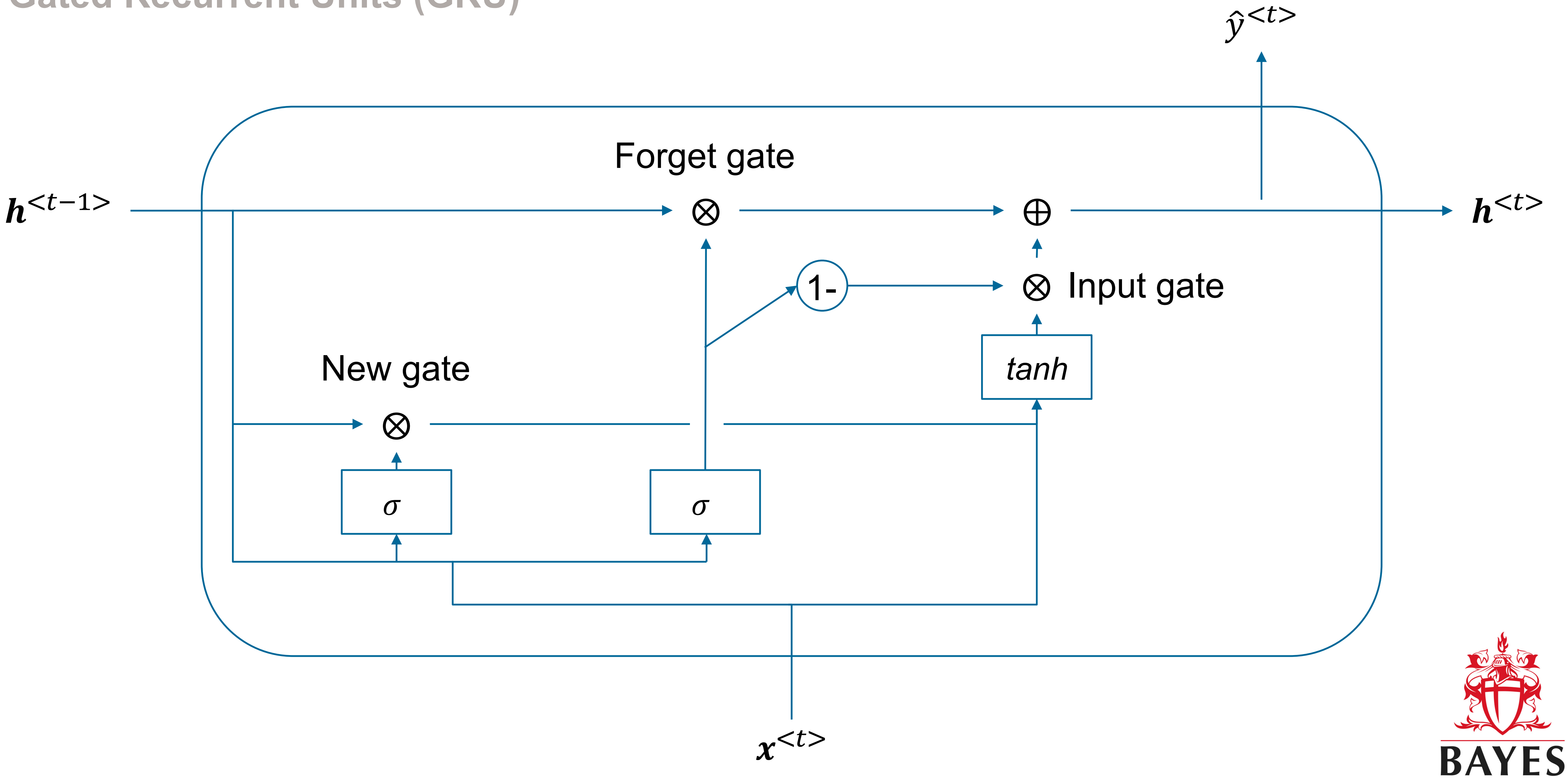
Long Short-Term Memory (LSTM) cells



LSTM units with "peephole connections"



Gated Recurrent Units (GRU)

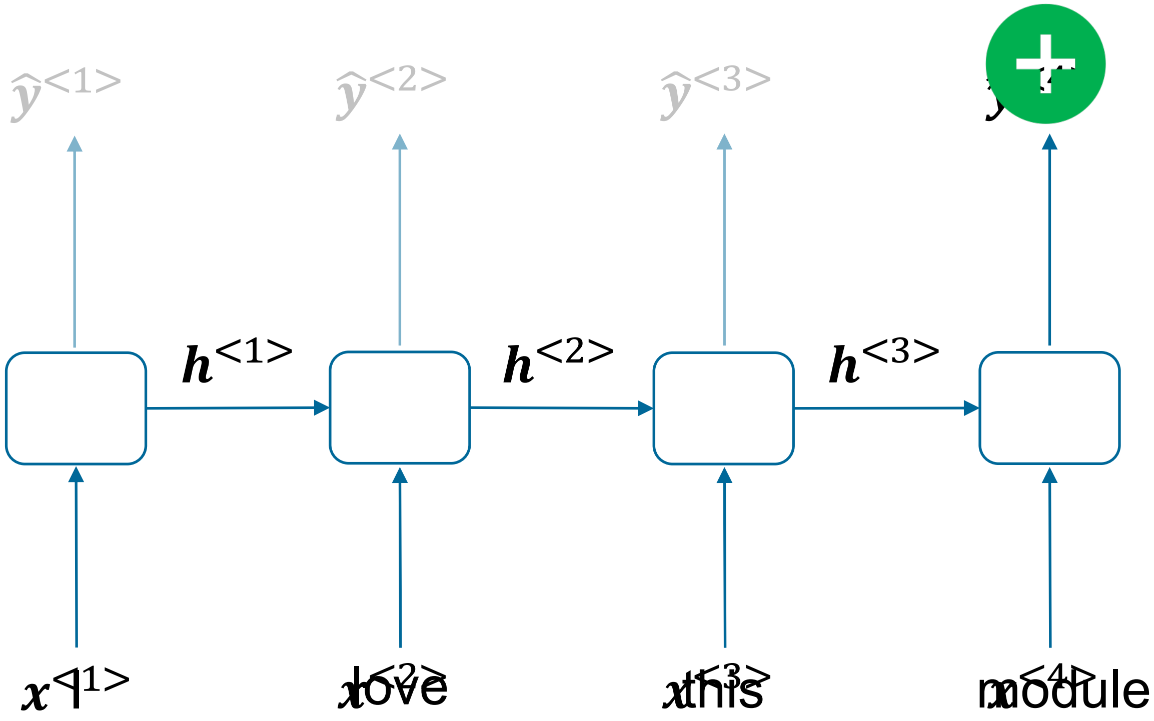


BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Long-term memory in practice

RNN variants and their applications (Time permitting)

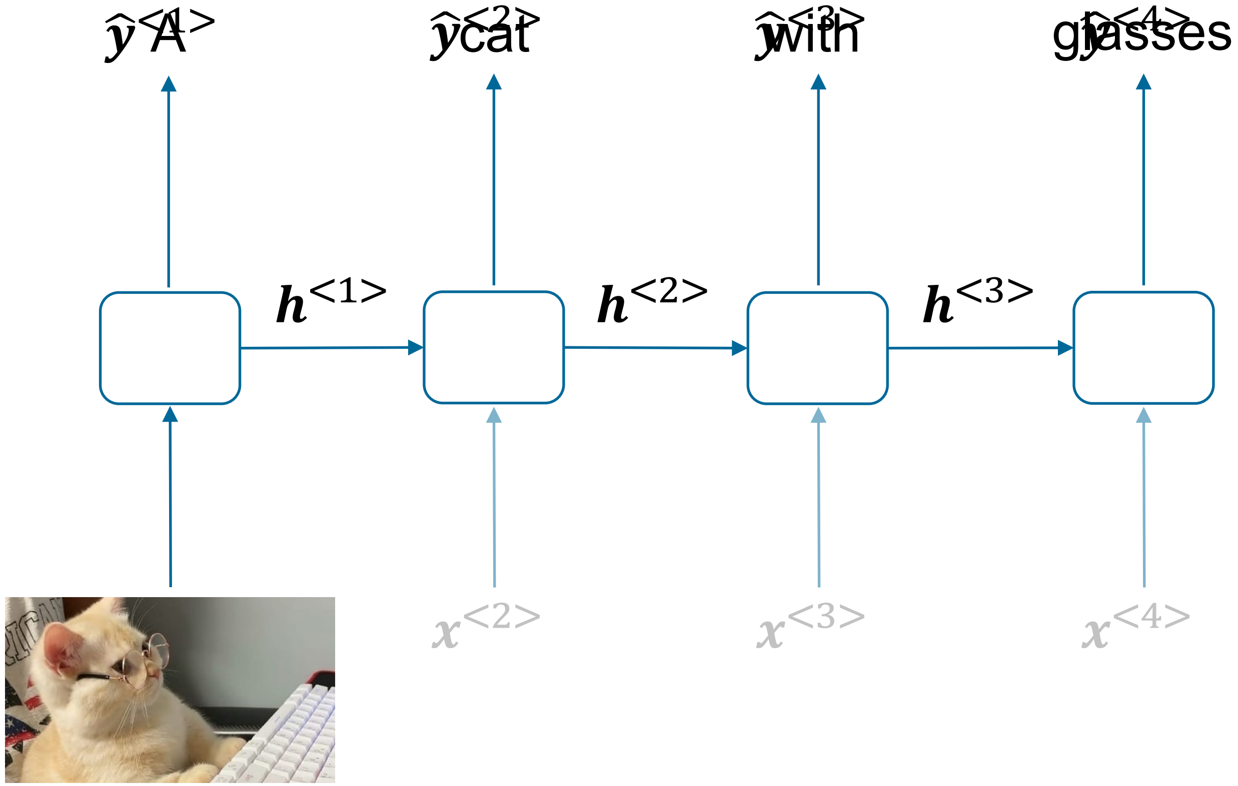
Sequence-to-vector networks



For example:

- Video activity recognition
- DNA sequence probing
- Sentiment classification

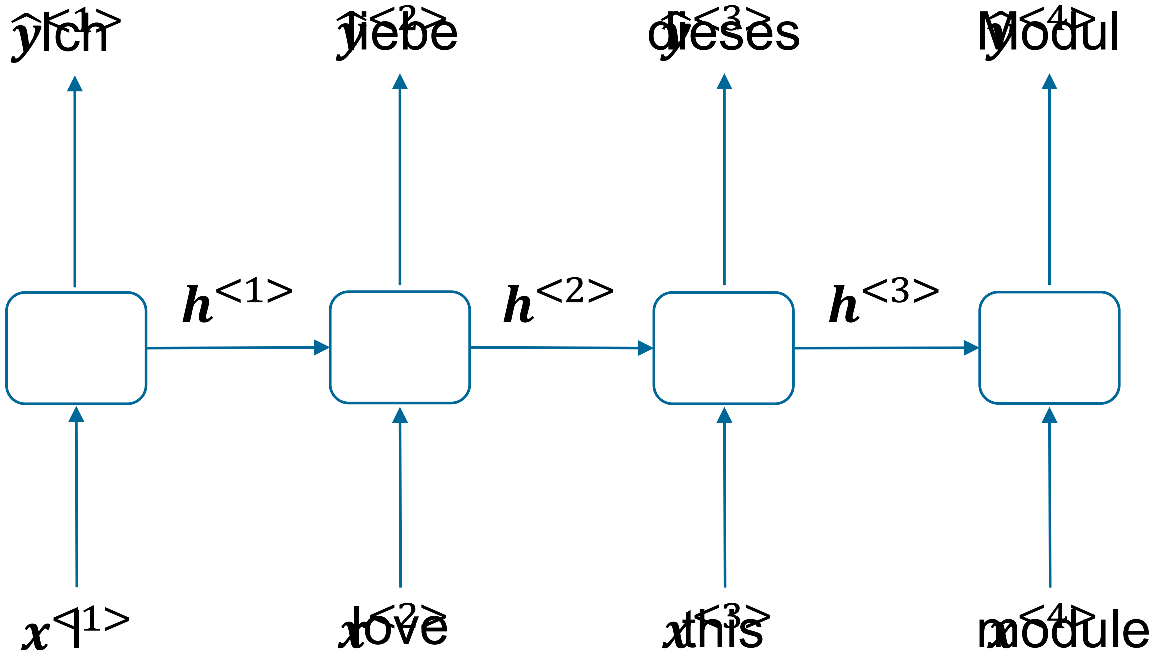
Vector-to-sequence networks



For example:

- Text generation
- Music generation
- Image captions

Sequence-to-sequence networks

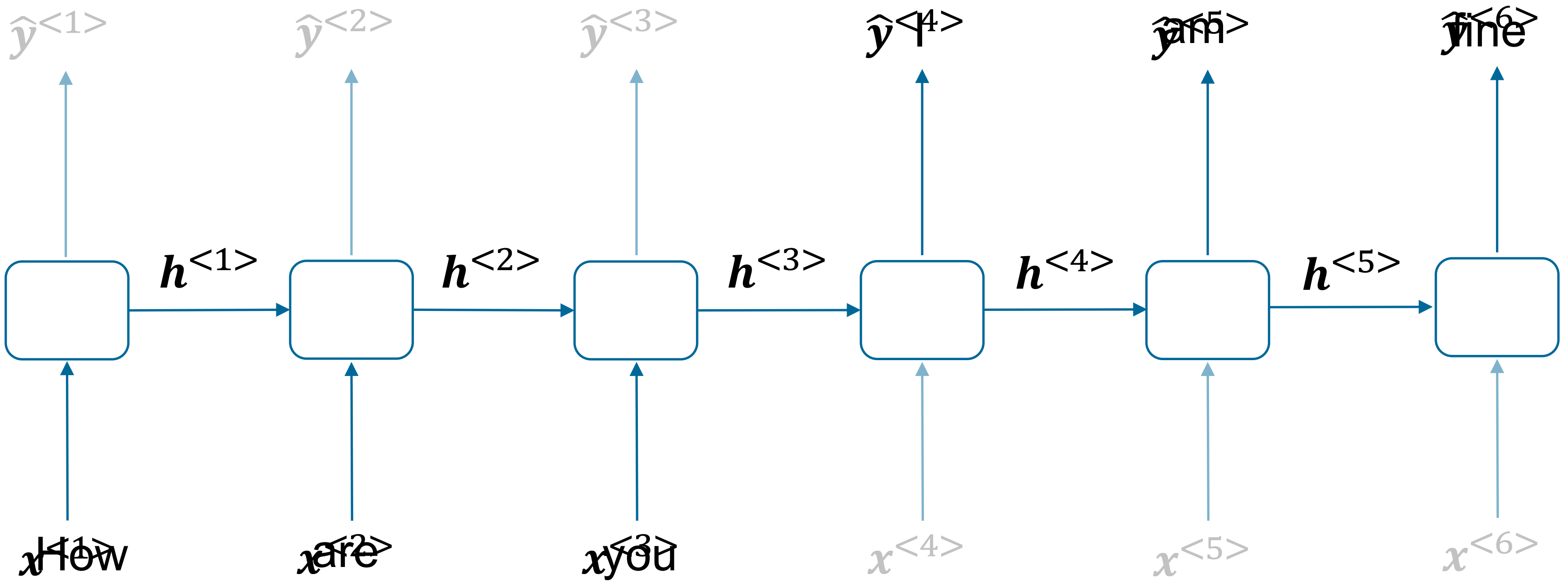


For example:

- Speech recognition
- Price predictions
- Translations

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Encoder-decoder networks

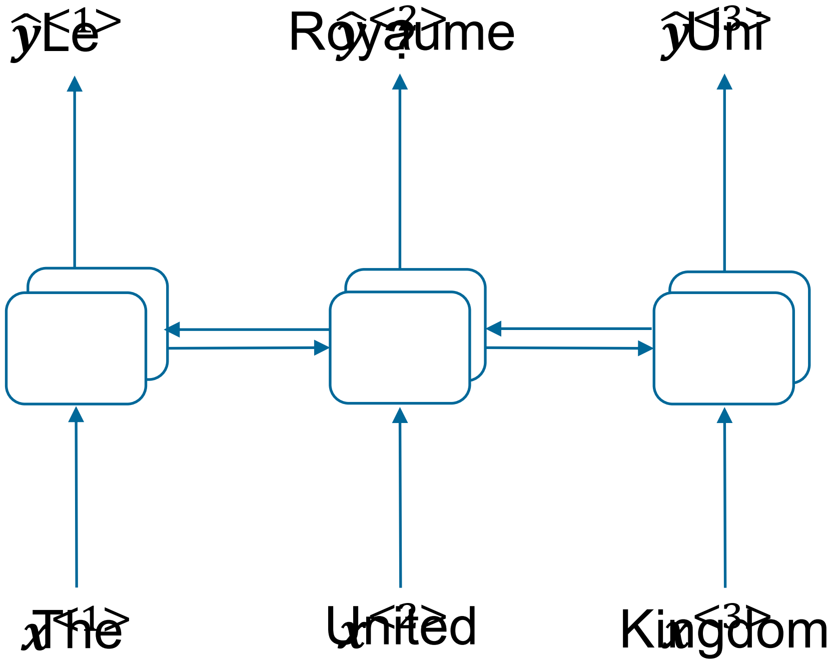


For example:

- Translations
- Dialogue

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Bidirectional RNNs – looking into the future



For example:

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

Let’s use a sequence-to-vector network to forecast multiple time steps ahead

Please fill out the module evaluation

https://city.surveys.evasysplus.co.uk/

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

See you next week!

Sources

- DeepLearning.AI, n.d.: deeplearning.ai
- Chollet, 2021, Deep Learning with Python (2nd edition)
- Garnelo, 2020, Lecture 6: Sequences and Recurrent Networks: https://storage.googleapis.com/deepmind-media/UCLxDeepMind_2020/L6%20-%20UCLxDeepMind%20DL2020.pdf
- Géron, 2022, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd edition)
- 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

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON