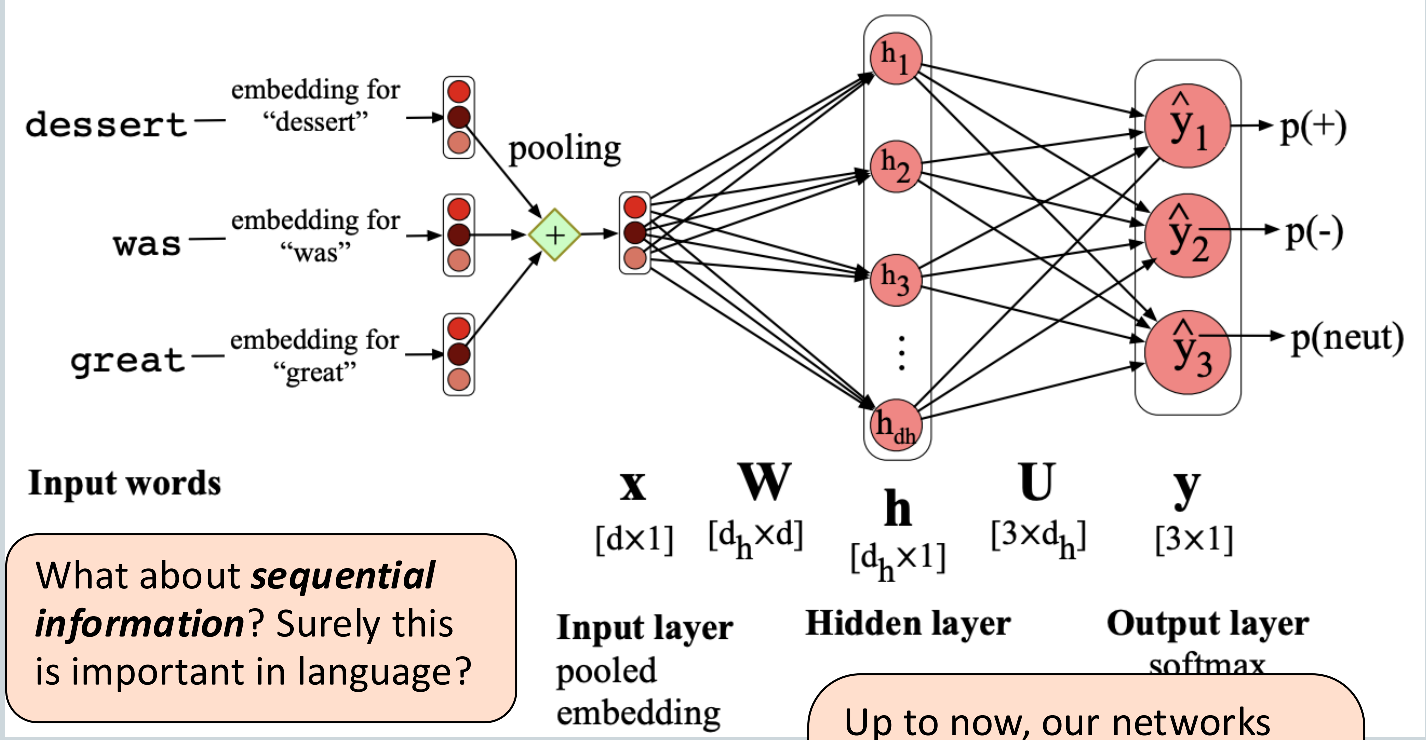
Further neural network architectures

NLP in one day

KING'S College LONDON

NIHR | Maudsley Biomedical Research Centre

Other architectures



What about ***sequential information***? Surely this is important in language?

Up to now, our networks have consisted of ***dense*** layers where all neurons in one layer are connected to all neurons in the following layer

(Jurafsky and Martin, Fig. 7.11)

We can design network topologies different neural processing units to increase the efficiency and representational power of our layers:

- Connect regions of input or hidden layers to single nodes in the next layer (***locally connected***)
- Learn features from one region that can be applied to others (***convoluting***)
- Carry over information from one step to the next (***memory***)

Updating our weights becomes more complicated!

Convolution - sharpening

0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

From the GIMP manual, V2.8, https://docs.gimp.org/en/index.html

Convolution - blurring

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

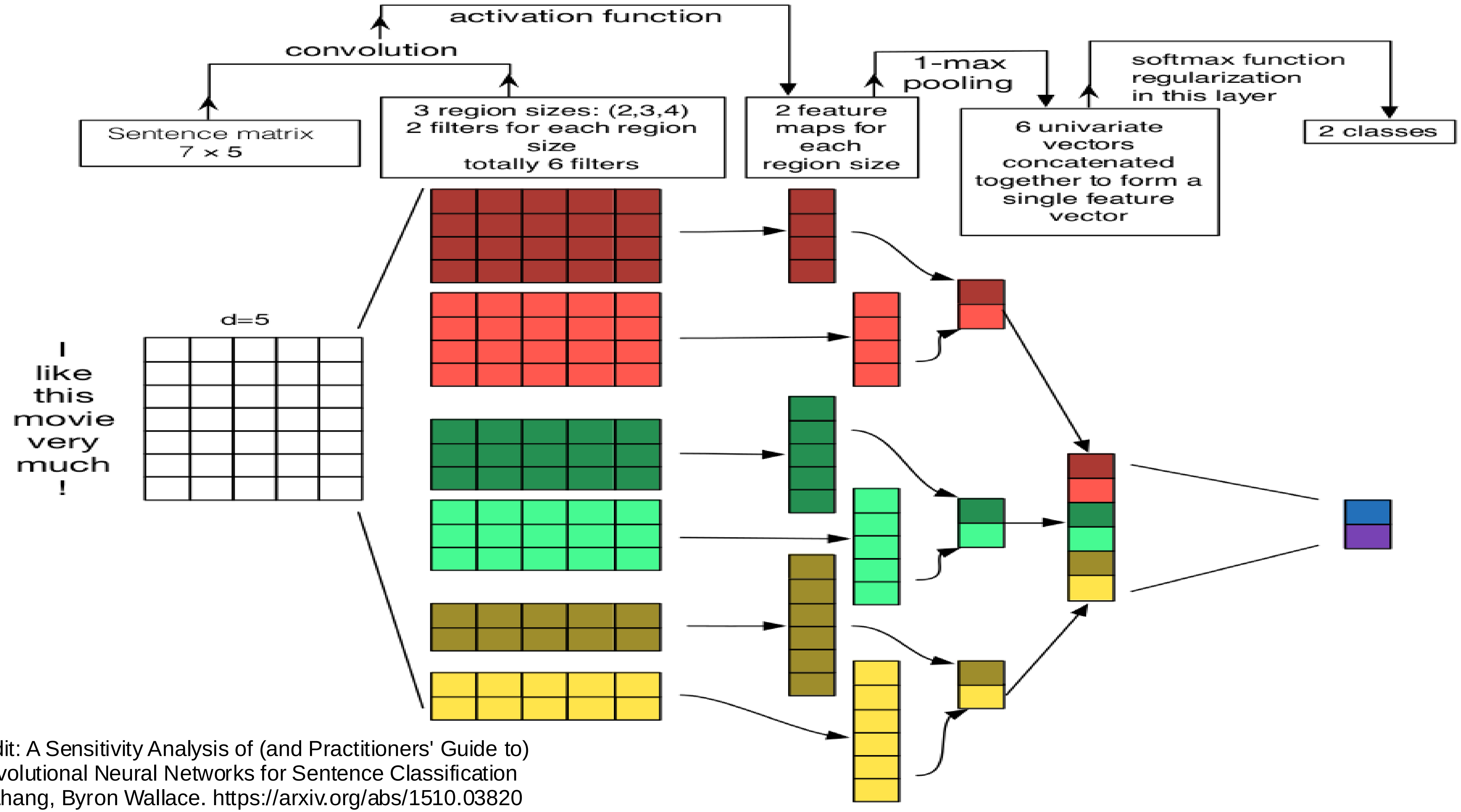
From the GIMP manual, V2.8, https://docs.gimp.org/en/index.html

Convolution - edge detection

	0	1	0	
	1	-4	1	
	0	1	0	

From the GIMP manual, V2.8, https://docs.gimp.org/en/index.html

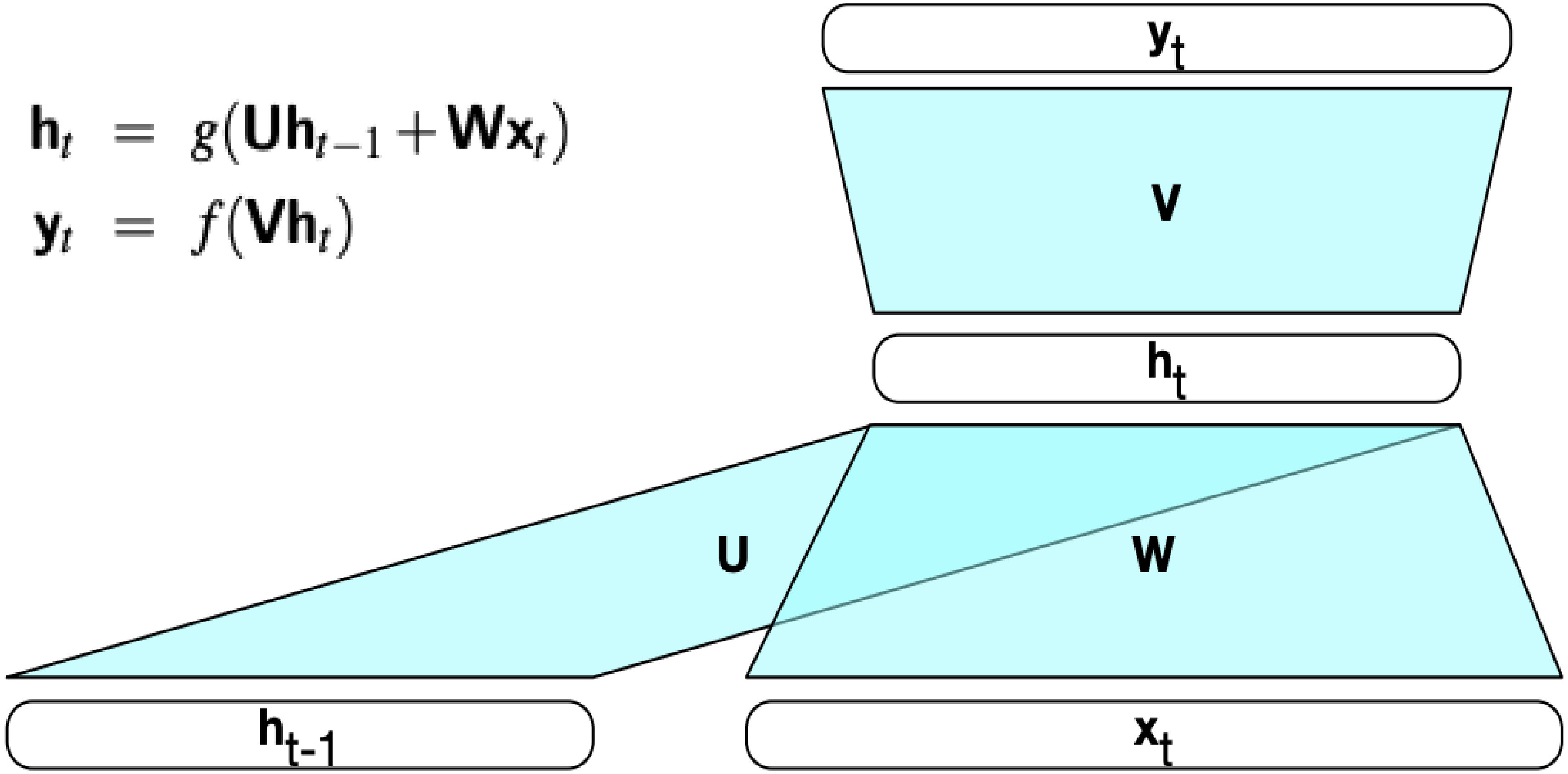
Applying convolution to text with CNNs



Credit: A Sensitivity Analysis of (and Practitioners' Guide to)
Convolutional Neural Networks for Sentence Classification
Ye Zhang, Byron Wallace. https://arxiv.org/abs/1510.03820

Recurrent neural network (RNN)

$$\mathbf{h}_t = g(\mathbf{U}\mathbf{h}_{t-1} + \mathbf{W}\mathbf{x}_t)$$
$$\mathbf{y}_t = f(\mathbf{V}\mathbf{h}_t)$$



(Jurafsky and Martin, Fig. 9.3)

To take account of sequential information, we could feed in the output from the previous step to each new step

RNN: the signals degrade over time

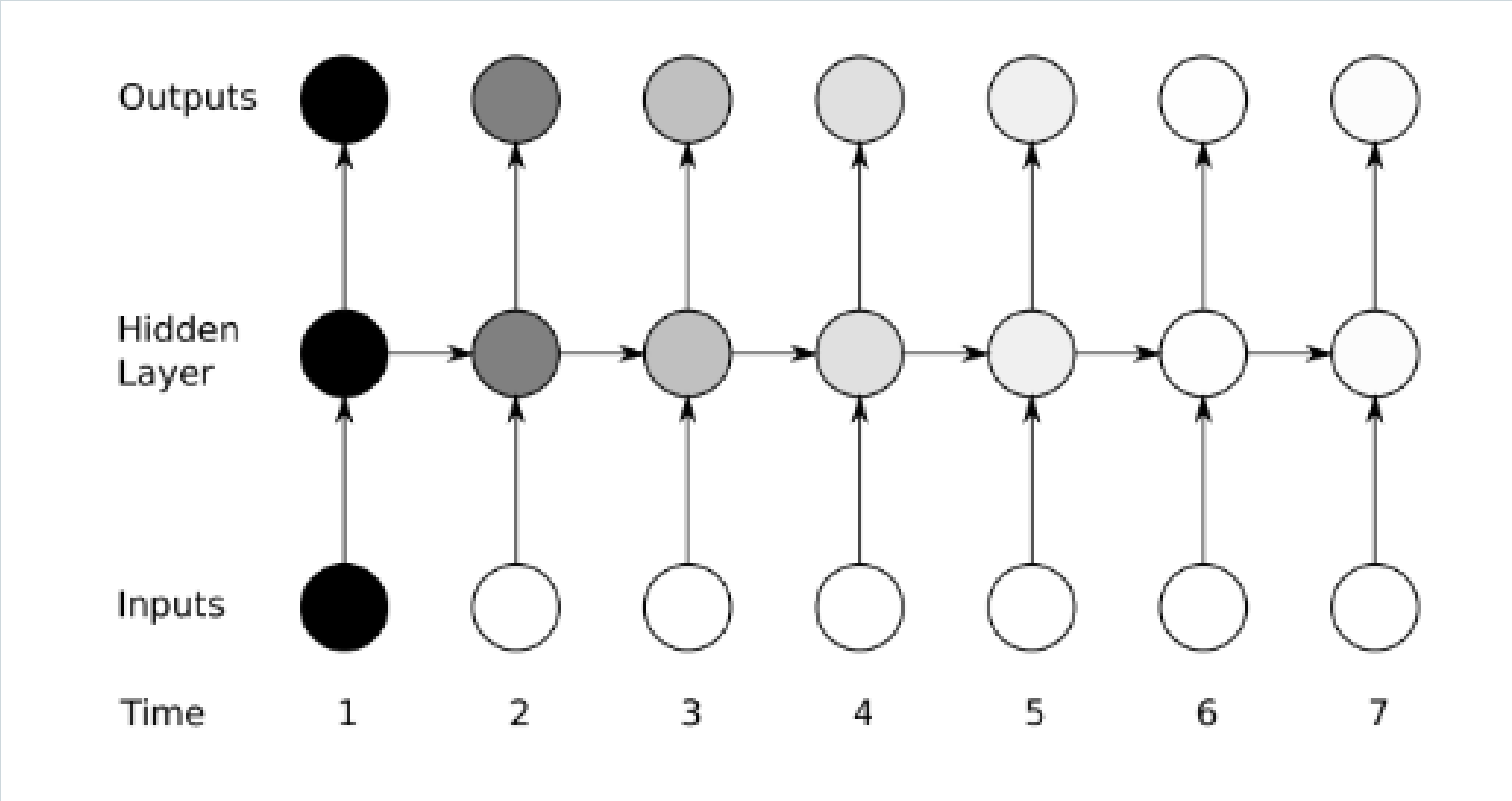


Image: Alex Graves, Supervised sequence labelling. Springer. 2012.

Long Short Term Memory (LSTM)

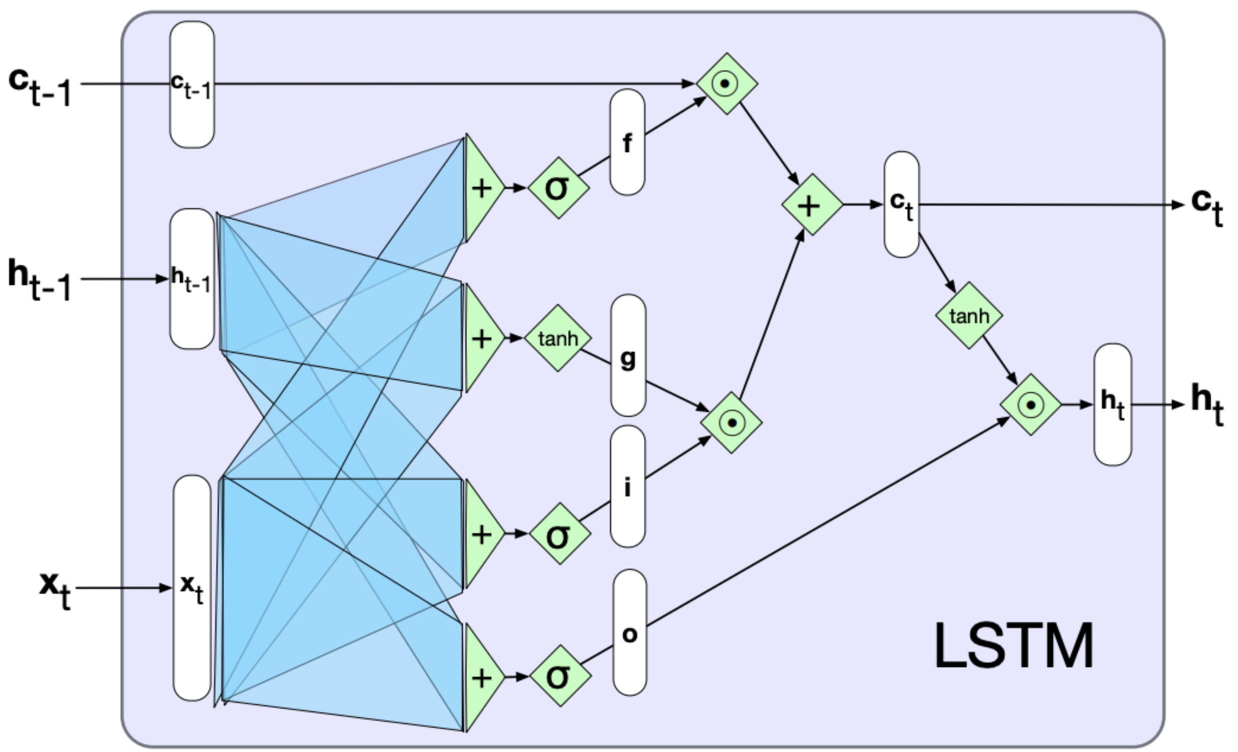
We can overcome the decay of RNNs by learning what we should forget and what we should remember at each step

Context **c** and hidden state **h** vectors are passed through from previous step

Forget gate **f** deletes information from the context no longer needed

Add gate **g** selects information to add to the current context

Output gate **o** selects what information is required for the current hidden state (not future ones, which were dealt with above)



(Jurafsky and Martin, Fig. 9.13)

LSTM selectively forgets and remembers information at each step

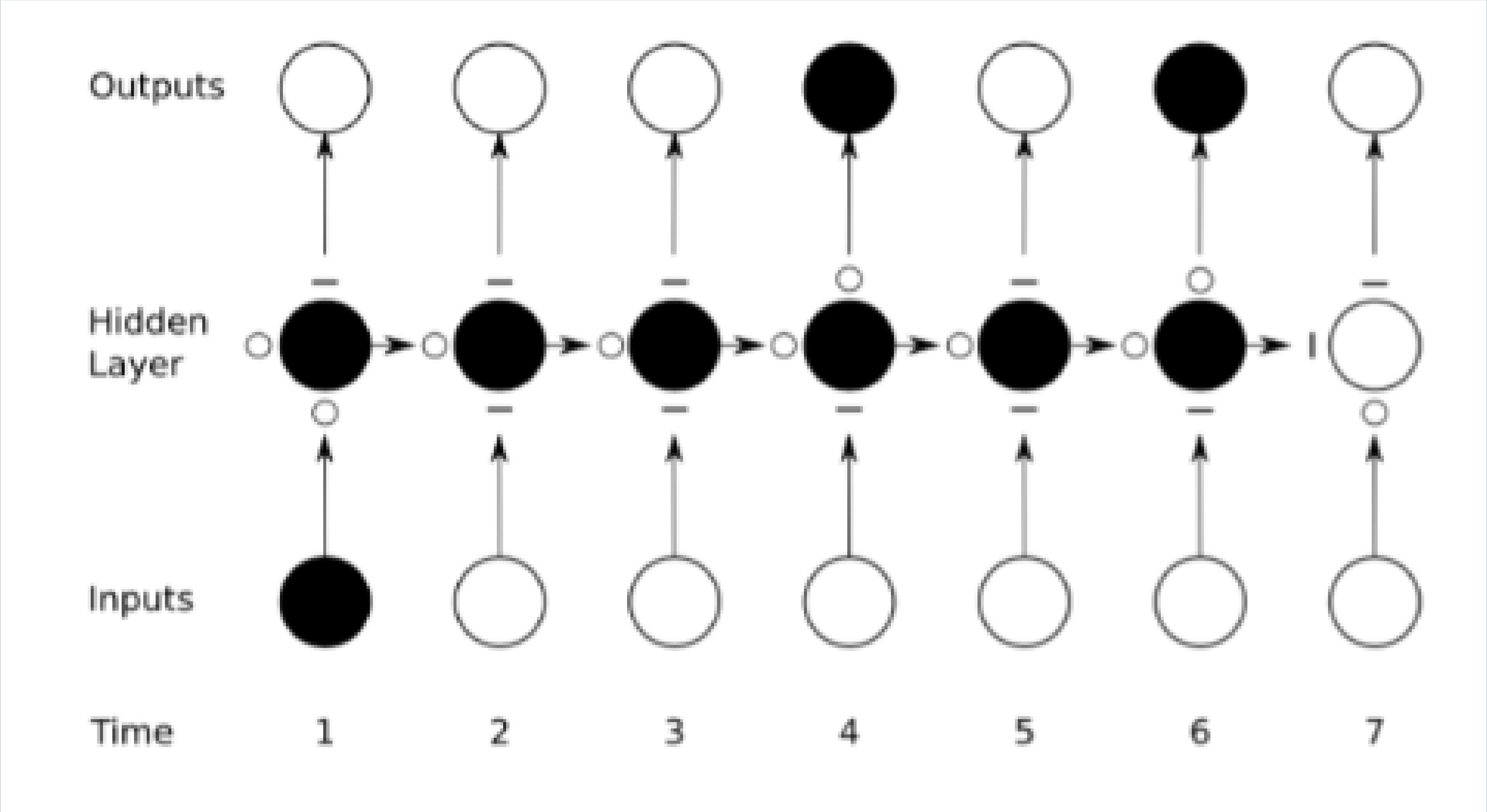
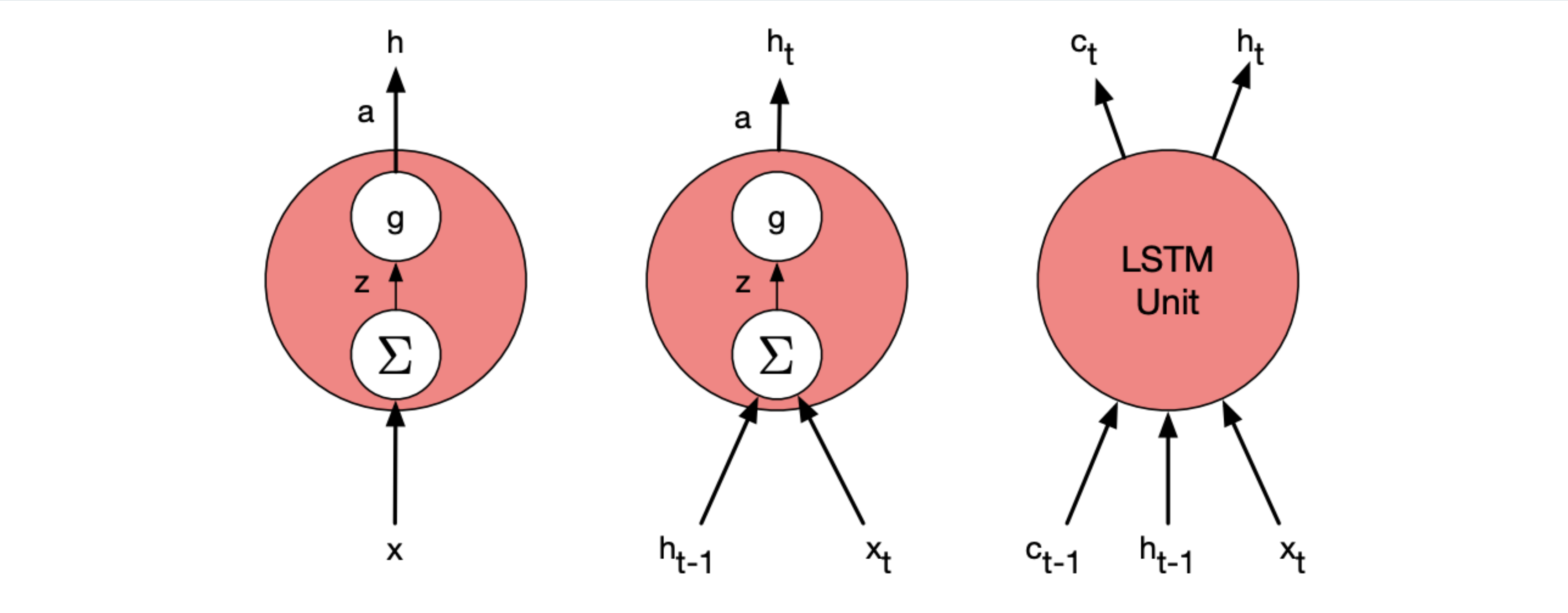


Image: Alex Graves, Supervised sequence labelling. Springer. 2012.

We can see these as different processing units



Simple feedforward unit

Recurrent unit

LSTM unit – as with the others, it is self-contained - modular

(Jurafsky and Martin, Fig. 9.14)

Thank you

angus.roberts@kcl.ac.uk
https://www.kcl.ac.uk/people/angus-roberts