

Recurrent Neural Networks

Kristian Wichmann

December 20, 2017

This follows Goodfellow and Bengio's book, borrowing/adapting a lot of the illustrations from this.

1 Simple cyclical graph

Imagine a situation, where a state $s = s^{(t)}$ depends on "time" t . Here t is discrete, a whole number, usually, running from 1 to τ . Further assume, that each time step is connected to the next through the same function:

$$s^{(t)} = f(s^{(t-1)}, \theta) \quad (1.1)$$

I.e., there is only an explicit dependence on the state from the last time step. Here θ is a shorthand for any parameters in the model. The unfolded graph of this model is shown in figure 1.

1.1 Cyclical graph with input

To elaborate on this model, assume that at each time step, there is also some kind of input $x^{(t)}$. So now the state - now called $h^{(t)}$ for hidden state - depends on this as well:

$$s^{(t)} = f(s^{(t-1)}, x^{(t)}, \theta) \quad (1.2)$$

Figure 2 shows both the folded and unfolded versions of this model.

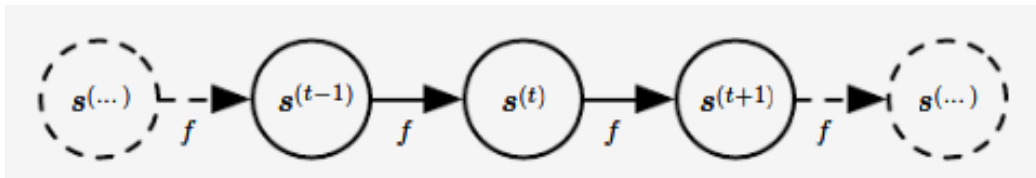


Figure 1: Simple cyclical graph.

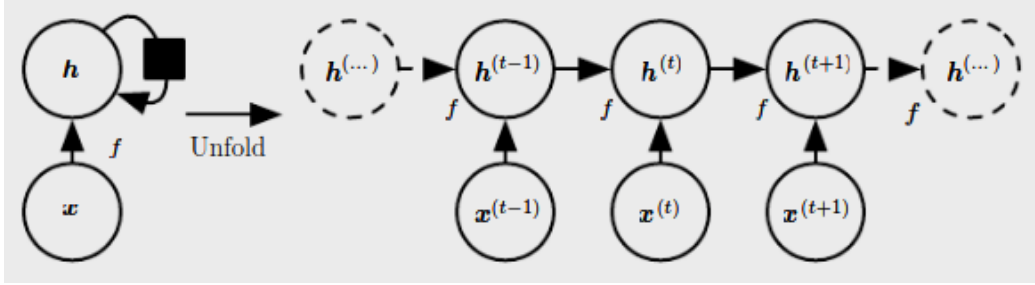


Figure 2: Cyclical graph with input.

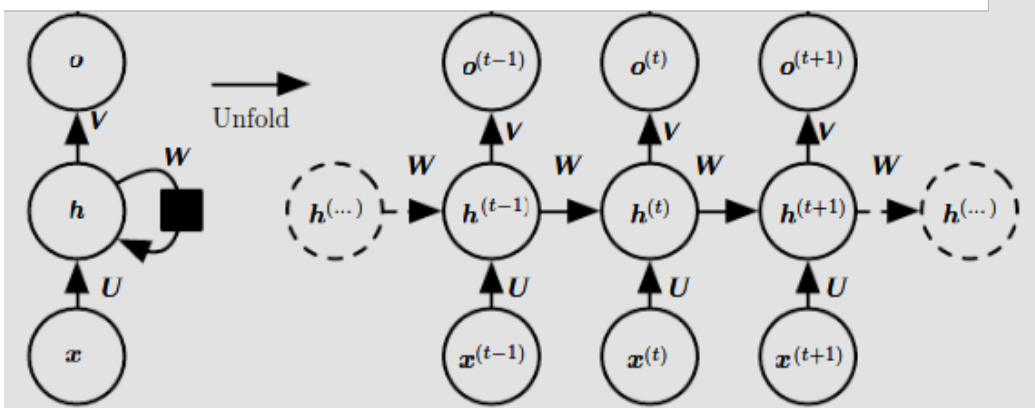


Figure 3: Cyclical graph with input as well as output.

1.2 Cyclical graph with both input and output

Finally, we would also like to model to provide some kind of output. We will assume there's output at every step, but in some applications, such as classification, we usually only care about the final output.

Such a model is shown in figure 3. Here, the output at time t depends only on the hidden state:

$$o^{(t)} = o(h^{(t)}) \quad (1.3)$$

2 Recurrent neural networks

As in feed forward neural networks, we will consider cases where the functions involved are linear followed by some kind of non-linearity. There's several ways to do this. We will explore some of them here.