

where we see that the state now contains information about the whole past sequence.

Recurrent neural networks can be built in many different ways. Much as almost any function can be considered a feedforward neural network, essentially any function involving recurrence can be considered a recurrent neural network.

Many recurrent neural networks use equation 10.5 or a similar equation to define the values of their hidden units. To indicate that the state is the hidden units of the network, we now rewrite equation 10.4 using the variable \mathbf{h} to represent the state:

$$\mathbf{h}^{(t)} = f(\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}; \boldsymbol{\theta}), \quad (10.5)$$

illustrated in figure 10.2, typical RNNs will add extra architectural features such as output layers that read information out of the state \mathbf{h} to make predictions.

When the recurrent network is trained to perform a task that requires predicting the future from the past, the network typically learns to use $\mathbf{h}^{(t)}$ as a kind of lossy summary of the task-relevant aspects of the past sequence of inputs up to t . This summary is in general necessarily lossy, since it maps an arbitrary length sequence $(\mathbf{x}^{(t)}, \mathbf{x}^{(t-1)}, \mathbf{x}^{(t-2)}, \dots, \mathbf{x}^{(2)}, \mathbf{x}^{(1)})$ to a fixed length vector $\mathbf{h}^{(t)}$. Depending on the training criterion, this summary might selectively keep some aspects of the past sequence with more precision than other aspects. For example, if the RNN is used in statistical language modeling, typically to predict the next word given previous words, it may not be necessary to store all of the information in the input sequence up to time t , but rather only enough information to predict the rest of the sentence. The most demanding situation is when we ask $\mathbf{h}^{(t)}$ to be rich enough to allow one to approximately recover the input sequence, as in autoencoder frameworks (chapter 14).

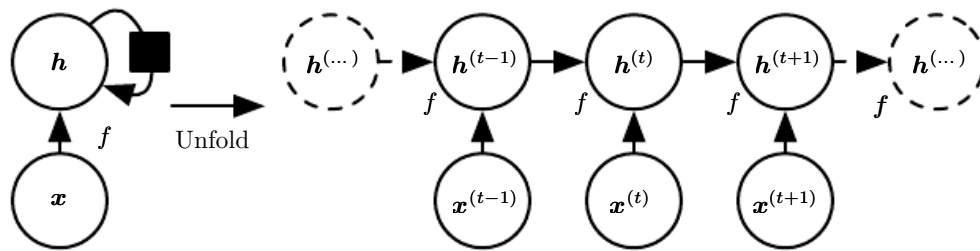


Figure 10.2: A recurrent network with no outputs. This recurrent network just processes information from the input \mathbf{x} by incorporating it into the state \mathbf{h} that is passed forward through time. (Left) Circuit diagram. The black square indicates a delay of a single time step. (Right) The same network seen as an unfolded computational graph, where each node is now associated with one particular time instance.

Equation 10.5 can be drawn in two different ways. One way to draw the RNN is with a diagram containing one node for every component that might exist in a