10.2.3 Recurrent Networks as Directed Graphical Models

In the example recurrent network we have developed so far, the losses $L^{(t)}$ were cross-entropies between training targets $\boldsymbol{y}^{(t)}$ and outputs $\boldsymbol{o}^{(t)}$. As with a feedforward network, it is in principle possible to use almost any loss with a recurrent network. The loss should be chosen based on the task. As with a feedforward network, we usually wish to interpret the output of the RNN as a probability distribution, and we usually use the cross-entropy associated with that distribution to define the loss. Mean squared error is the cross-entropy loss associated with an output distribution that is a unit Gaussian, for example, just as with a feedforward network.

When we use a predictive log-likelihood training objective, such as equation 10.12, we train the RNN to estimate the conditional distribution of the next sequence element $\boldsymbol{y}^{(t)}$ given the past inputs. This may mean that we maximize the log-likelihood

$$\log p(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(t)}), \tag{10.29}$$

or, if the model includes connections from the output at one time step to the next time step,

$$\log p(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(t)}, \boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(t-1)}). \tag{10.30}$$

Decomposing the joint probability over the sequence of $\boldsymbol{y}$ values as a series of one-step probabilistic predictions is one way to capture the full joint distribution across the whole sequence. When we do not feed past $\boldsymbol{y}$ values as inputs that condition the next step prediction, the directed graphical model contains no edges from any $\boldsymbol{y}^{(i)}$ in the past to the current $\boldsymbol{y}^{(t)}$. In this case, the outputs $\boldsymbol{y}$ are conditionally independent given the sequence of $\boldsymbol{x}$ values. When we do feed the actual $\boldsymbol{y}$ values (not their prediction, but the actual observed or generated values) back into the network, the directed graphical model contains edges from all $\boldsymbol{y}^{(i)}$ values in the past to the current $y^{(t)}$ value.

As a simple example, let us consider the case where the RNN models only a sequence of scalar random variables $\mathbb{Y} = \{\mathrm{y}^{(1)}, \dots, \mathrm{y}^{(\tau)}\}$, with no additional inputs x. The input at time step t is simply the output at time step $t-1$. The RNN then defines a directed graphical model over the y variables. We parametrize the joint distribution of these observations using the chain rule (equation 3.6) for conditional probabilities:

$$P(\mathbb{Y}) = P(\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(\tau)}) = \prod_{t=1}^{\tau} P(\mathbf{y}^{(t)} \mid \mathbf{y}^{(t-1)}, \mathbf{y}^{(t-2)}, \dots, \mathbf{y}^{(1)}) \tag{10.31}$$

where the right-hand side of the bar is empty for $t = 1$, of course. Hence the negative log-likelihood of a set of values $\{y^{(1)}, \dots, y^{(\tau)}\}$ according to such a model