

behavior (if the same Jacobian was applied iteratively). Even though  $\mathbf{h}^{(t)}$  or a small variation of  $\mathbf{h}^{(t)}$  of interest in back-propagation are real-valued, they can be expressed in such a complex-valued basis. What matters is what happens to the magnitude (complex absolute value) of these possibly complex-valued basis coefficients, when we multiply the matrix by the vector. An eigenvalue with magnitude greater than one corresponds to magnification (exponential growth, if applied iteratively) or shrinking (exponential decay, if applied iteratively).

With a nonlinear map, the Jacobian is free to change at each step. The dynamics therefore become more complicated. However, it remains true that a small initial variation can turn into a large variation after several steps. One difference between the purely linear case and the nonlinear case is that the use of a squashing nonlinearity such as  $\tanh$  can cause the recurrent dynamics to become bounded. Note that it is possible for back-propagation to retain unbounded dynamics even when forward propagation has bounded dynamics, for example, when a sequence of  $\tanh$  units are all in the middle of their linear regime and are connected by weight matrices with spectral radius greater than 1. However, it is rare for all of the  $\tanh$  units to simultaneously lie at their linear activation point.

The strategy of echo state networks is simply to fix the weights to have some spectral radius such as 3, where information is carried forward through time but does not explode due to the stabilizing effect of saturating nonlinearities like  $\tanh$ .

More recently, it has been shown that the techniques used to set the weights in ESNs could be used to *initialize* the weights in a fully trainable recurrent network (with the hidden-to-hidden recurrent weights trained using back-propagation through time), helping to learn long-term dependencies (Sutskever, 2012; Sutskever *et al.*, 2013). In this setting, an initial spectral radius of 1.2 performs well, combined with the sparse initialization scheme described in section 8.4.

## 10.9 Leaky Units and Other Strategies for Multiple Time Scales

One way to deal with long-term dependencies is to design a model that operates at multiple time scales, so that some parts of the model operate at fine-grained time scales and can handle small details, while other parts operate at coarse time scales and transfer information from the distant past to the present more efficiently. Various strategies for building both fine and coarse time scales are possible. These include the addition of skip connections across time, “leaky units” that integrate signals with different time constants, and the removal of some of the connections