T-61.5030 Advanced course in neural computing Karhunen, Peltonen

Exercise 11, Nov. 30, 2006

1. Show that a recurrent MLP having two hidden layers (see Fig. 15.4 in Haykin) can be represented by the state-space model

$$\begin{aligned}\mathbf{x}(n+1) &= \mathbf{f}(\mathbf{x}(n), \mathbf{u}(n)) \\ \mathbf{y}(n) &= \mathbf{g}(\mathbf{x}(n), \mathbf{u}(n)) \ ,\end{aligned}$$

   where $\mathbf{u}$ is the input, $\mathbf{y}$ is the output, $\mathbf{x}$ is the state, and $\mathbf{f}$ and $\mathbf{g}$ are vector-valued nonlinear functions.

2. Is it possible to transform any state-space model into a NARX model (in which the output is fed back to the inputs through a delay line)? How about vice versa?

3. Construct an example of a network that is observable and of a network that is not. Do the same after replacing observability with controllability.

4. We have discussed two learning methods for recurrent MLPs, real-time recurrent learning (RTRL) and back-propagation through time (BPTT). In BPTT, the structure of the recurrent network is "unfolded in time" so that the time dimension is transformed into additional layers in the network. In RTRL, the gradient of the state of the network is updated recursively, and it is then used for adjusting the weights of the network with instantaneous gradients of the error function (cf. Haykin's book if this explanation is too concise). Discuss the relative merits of the algorithms from the point of view of applicability (you may use mathematics but you do not need to). Pay special attention to the following problems:

   (a) What are the crucial differences between RTRL and BPTT?

   (b) What problems does RTRL have?

   (c) How to solve the problems?

5. Construct the simple finite-state automaton in Haykin's Figure P15.17 (more information in Haykin's exercise 15.17) using a second order recurrent network. Would the network be able to learn the automaton? If so, how?