Chapter 4
Long Short-Term Memory

As discussed in the previous chapter, an important benefit of recurrent neural networks is their ability to use contextual information when mapping between input and output sequences. Unfortunately, for standard RNN architectures, the range of context that can be in practice accessed is quite limited. The problem is that the influence of a given input on the hidden layer, and therefore on the network output, either decays or blows up exponentially as it cycles around the network's recurrent connections. This effect is often referred to in the literature as the *vanishing gradient problem* (Hochreiter, 1991; Hochreiter et al., 2001a; Bengio et al., 1994). The vanishing gradient problem is illustrated schematically in Figure 4.1

Numerous attempts were made in the 1990s to address the problem of vanishing gradients for RNNs. These included non-gradient based training algorithms, such as simulated annealing and discrete error propagation (Bengio et al., 1994), explicitly introduced time delays (Lang et al., 1990; Lin et al., 1996; Plate, 1993) or time constants (Mozer, 1992), and hierarchical sequence compression (Schmidhuber, 1992). The approach favoured by this book is the *Long Short-Term Memory* (LSTM) architecture (Hochreiter and Schmidhuber, 1997).

This chapter reviews the background material for LSTM. Section 4.1 describes the basic structure of LSTM and explains how it tackles the vanishing gradient problem. Section 4.3 discusses an approximate and an exact algorithm for calculating the LSTM error gradient. Section 4.4 describes some enhancements to the basic LSTM architecture. Section 4.2 discusses the effect of preprocessing on long range dependencies. Section 4.6 provides all the equations required to train and apply LSTM networks.

4.1 Network Architecture

The LSTM architecture consists of a set of recurrently connected subnets, known as memory blocks. These blocks can be thought of as a differentiable version of the memory chips in a digital computer. Each block contains one or more self-connected memory cells and three multiplicative units—the input,

A. Graves: Supervised Sequence Labell. with Recur. Neur. Networks, SCI 385, pp. 37–45.
springerlink.com
© Springer-Verlag Berlin Heidelberg 2012

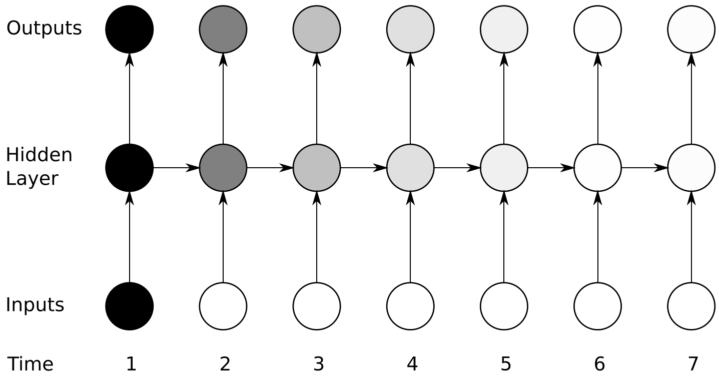

Fig. 4.1 The vanishing gradient problem for RNNs. The shading of the nodes in the unfolded network indicates their sensitivity to the inputs at time one (the darker the shade, the greater the sensitivity). The sensitivity decays over time as new inputs overwrite the activations of the hidden layer, and the network 'forgets' the first inputs.

output and forget gates—that provide continuous analogues of write, read and reset operations for the cells.

Figure 4.2 provides an illustration of an LSTM memory block with a single cell. An LSTM network is the same as a standard RNN, except that the summation units in the hidden layer are replaced by memory blocks, as illustrated in Fig. 4.3. LSTM blocks can also be mixed with ordinary summation units, although this is typically not necessary. The same output layers can be used for LSTM networks as for standard RNNs.

The multiplicative gates allow LSTM memory cells to store and access information over long periods of time, thereby mitigating the vanishing gradient problem. For example, as long as the input gate remains closed (i.e. has an activation near 0), the activation of the cell will not be overwritten by the new inputs arriving in the network, and can therefore be made available to the net much later in the sequence, by opening the output gate. The preservation over time of gradient information by LSTM is illustrated in Figure 4.4.

Over the past decade, LSTM has proved successful at a range of synthetic tasks requiring long range memory, including learning context free languages (Gers and Schmidhuber, 2001), recalling high precision real numbers over extended noisy sequences (Hochreiter and Schmidhuber, 1997) and various tasks requiring precise timing and counting (Gers et al., 2002). In particular, it has solved several artificial problems that remain impossible with any other RNN architecture.

Additionally, LSTM has been applied to various real-world problems, such as protein secondary structure prediction (Hochreiter et al., 2007; Chen and Chaudhari, 2005), music generation (Eck and Schmidhuber, 2002),