ABSTRACT

Backpropagation has shown to be an efficient learning rule for graded perceptrons. However, as initially introduced, it was limited to feedforward structures. Extension of backpropagation to systems with feedback was done by this author, in [4]. In this paper, this extension is presented, and the error propagation circuit is interpreted as the transpose of the linearized perceptron network. The error propagation network is shown to always be stable during training, and a sufficient condition for the stability of the perceptron network is derived. Finally, potentially useful relationships with Hopfield networks and Boltzmann machines are discussed.

1. INTRODUCTION

Backpropagation has been independently introduced by several authors (including at least Parker, Le Cun, and Rummelhart, Hinton and Williams), as a learning rule for feedforward multilayer graded perceptron networks [1]. Its power is by now well demonstrated (see [2] for an example). It is based on the minimization of the squared error of the actual output, relative to a desired output, this minimization being performed through a gradient descent technique. Its extension to a special class of perceptrons with feedback was made in [1], following a suggestion by Minsky and Papert [3]. This class of perceptrons is characterized by the (implicit) assumption of the existence of a sample-and-hold operation at the output of each unit, all sample-and-holds being triggered synchronously. Under this assumption, the perceptron with feedback can be "unfolded" in time, into an equivalent feedforward one, and can therefore be trained using backpropagation. An important limitation of backpropagation in this context, however, is
that it demands the existence of an essentially unlimited amount of memory in each unit.

In this paper, we will be concerned with a different class of feedback perceptrons: they will be assumed not to have any sample-and-hold; instead, for each input pattern, the outputs of the units will change continuously in time until a stable state is reached. The outputs of the perceptron are observed only in the stable state, and are then compared to the desired outputs. Training, i.e., weight update, is performed with the system in the stable state. The input-output mapping to be learned by the perceptron is assumed to be combinatorial, i.e., the desired outputs depend only on present inputs, not on past ones.

The extension of backpropagation to this class of perceptrons was first made by this author, in [4]. Here, we will review its derivation, and we will briefly discuss the problem of stability. We will then proceed to discuss the relationships between feedback perceptrons and Hopfield networks and Boltzmann machines.

2. BACKPROPAGATION IN FEEDBACK PERCEPTRONS

Consider a graded perceptron network, and designate by $x_k$ the external inputs ($k = 1, ..., K$), by $y_i$ the outputs of the units ($i = 1, ..., N$), by $s_i$ the result of the sum performed at the input of unit $i$, and by $o_p$ the external outputs ($p \in O$, where $O$ is the set of units producing external outputs). The static equations of the perceptron network are

$$s_i = \sum_{n=1}^{N} a_{ni} y_n + \sum_{k=1}^{K} b_{ki} x_k + c_i \quad i = 1, ..., N \quad (1)$$

$$y_i = S_i(s_i) \quad i = 1, ..., N \quad (2)$$

$$o_p = y_p \quad p \in O \quad (3)$$

where $a_{ni}$ and $b_{ki}$ are weights, $c_i$ is a bias term, and $S_i$ is the nonlinear function in unit $i$ (usually a sigmoid). In a feedforward perceptron, the units can be numbered in such a way that the array $[a_{ni}]$ is lower triangular, with zeros in the main diagonal. Note that in the nomenclature used in this paper, we do not consider external inputs as units.