Chapter 7
APECS:
An Adaptively Parameterised Model of Associative Learning and Memory

I.P.L. McLaren
University of Exeter, UK

ABSTRACT

In this chapter the author will first give an overview of the ideas behind Adaptively Parameterised Error Correcting Learning (APECS) as introduced in McLaren (1993). It will take a somewhat historical perspective, tracing the development of this approach from its origins as a solution to the sequential learning problem identified by McCloskey and Cohen (1989) in the context of paired associate learning, to its more recent application as a model of human contingency learning.

BACKGROUND: THE SEQUENTIAL LEARNING PROBLEM

The development of novel connectionist algorithms (Rumelhart, Hinton, and Williams, 1986; Ackley, Hinton, and Sejnowski, 1985) capable of driving learning in multi-layer networks can be seen as one of the major developments in cognitive science in the nineteen-eighties. One of these algorithms, Back Propagation (Rumelhart, Hinton, and Williams, 1986) used gradient descent to learn input / output relationships, and was typically instantiated in feed-forward architectures. This otherwise successful approach, however, came up against the sequential learning problem identified by McCloskey and Cohen (1989) and further analysed by Ratcliff (1990). A general statement of this problem is that if a network employing Back Propagation is first taught one set of input / output relations, and then some other mapping is learnt whose input terms are similar to those first used in training, then a near complete loss of performance on the first mapping is observed on test. We can say that the new learning wipes out the old. This is not a necessary characteristic of the feed-forward architecture, because, if training alternates between the two mappings, repeatedly teaching first one and then the other, eventually a solution is reached that captures both sets of input / output relationships. Thus, this “catastrophic

DOI: 10.4018/978-1-60960-021-1.ch007

interference", when new learning erases old, is only seen if the two mappings are learnt in sequence. This does not mean that this property of the learning algorithm can be ignored, however, as learning (in humans and networks) often takes place within a sequential format (eg see Ratcliff, 1990; Hinton and Plaut, 1987; Sejnowski and Rosenberg, 1987).

As a simple example of this general type of problem, consider modelling a paired-associate experiment (based on Barnes and Underwood, 1959) in which human subjects are required to learn a list (list 1) of eight nonsense syllable - adjective pairs to a criterion of 100%. That is, after some number of training trials, the subject is able to provide the correct adjectival response to each nonsense syllable stimulus. After learning list 1, the subjects learn list 2, which employs the same nonsense syllables as the first, but new adjectives paired with them. Training continues until subjects are near perfect on this list (>90%). They are then asked to recall the original list 1 adjectival responses for each nonsense syllable. Performance drops to around 50% for this list, which is taken to be an instance of retroactive interference (control groups suggest that it is not simply the passage of time that is responsible for this decline in performance).

As McCloskey and Cohen (1989) showed, this task can be modelled in a feed-forward two layer network running Back Propagation. The list 'context' and the nonsense syllables (eg dax, teg) are the input, and the adjectives (e.g. regal, sleek) are the output (see Figure 1 which shows both the network in question and the experimental design).

After cycling through the list several times, activation of the input nodes representing list context in conjunction with a nonsense syllable results in the activation of the output nodes corresponding to the correct adjective via the set of connection strengths or weights developed by the network. During learning of the second list, nodes standing for the List 2 context are used in conjunction with the old nonsense syllable nodes, together with extra output nodes representing the new adjectives (keen, swift). Training proceeds until activation of nodes representing List 2 + dax (for example) results in activation of the 'keen' node. Now, List 1 recall can be tested by presenting List 1 + dax as input. The result produced by the network is – 'keen'. There is no sign of previously having learnt 'regal' to this input. McCloskey and Cohen were able to show that even minimal training on List 2 resulted in (at best!) nearly complete loss of List 1 on test, rather than the 50% loss shown in humans (at worst). This result does not depend on the local coding scheme employed here, as they obtained the same outcome using distributed representations of contexts, stimuli and responses.

Figure 2 gives simulation results for this sequential learning task employing a two item list and employing a modified version of Back Propagation that is used throughout this paper. Despite these minor differences, the results are the same as those reported by McCloskey and Cohen.

After training on List 1 until performance meets their "within 0.1" criterion on test, i.e. activation of an input pattern produces the correct response to within 0.1 of each node's target activation level, learning the List 2 items to the same criterion powerfully degrades List 1 performance. In fact, testing on List 1 now fails to meet a "best match" criterion which requires that the output be more similar to the target response than to any of the other possible responses in the lists. Analysis of these simulation results indicates that the difficulty facing the network is that the initial List 1 solution (i.e. the weights) is not one that can survive learning of List 2, because the List 1 responses to the nonsense syllables have to be suppressed in some fashion, and once this is done they cannot be recovered. Only when the lists are alternated during training can a List 1 solution that is protected from the effects of List 2 learning be developed (an example is shown in Figure 3). In fact, if the network was alternated on the two

Related Content