The Howl Effect in Dynamic-Network Learning

Chris Thornton
Cognitive and Computing Sciences
University of Sussex
Brighton
BN1 9QH
UK

Email: Christopher.Thornton@firenet.uk.com
Tel: (44)1273 678856

May 21, 2003

Abstract

The paper is concerned with those variants of the back-propagation learning scheme in which new hidden units are introduced one at a time (eg. the Tiling algorithm [1], the Upstart algorithm [2], and the Cascade-Correlation algorithm [3]). It suggests that such incremental schemes are necessarily misled in situations where the task requires the existence of several units acting in concert (eg. in the case where the hidden units must provide an optimal binary encoding of the input units). The paper proposes that, as a generic solution, incremental schemes should be capable of introducing subnets of variable size.

Introduction

In the traditional neural-network scenario, the architecture of the network is fixed and learning only affects the strengths of weights on connections. Recently, however, various researchers have described schemes for *dynamic-network learning*. In these schemes the topology of the network is altered or augmented during learning. Examples include the Tiling algorithm [1], the Upstart algorithm [2], and the Cascade-Correlation algorithm [3]. These dynamic schemes are all based on the well-known back-propagation learning mechanism. Back-propagation requires that the network is pre-wired in an appropriate way [4]. The dynamic schemes attempt to alleviate this requirement by enabling an appropriate architecture to be developed automatically as a part of learning.

There are various reasons for wanting the development of the network architecture to be under the control of learning rather than under the control of a "programmer". In particular there is the pragmatic reason that in many cases it may be very hard to pre-specify an appropriate architecture for a given problem. In addition, Fahlman and Lebiere have recently identified some theoretical problems which may beset back-propagation learning in those cases where the complete network exists from the beginning of learning [3].

They note that in back-propagation each hidden unit is attempting to learn a correct set of weights at the same time as all the other hidden units. The computational context in which the hidden unit operates is therefore constantly changing: each unit thus attempts to reach a 'moving target' and may indulge in unfruitful oscillations. Fahlman and Lebiere also describe the *herd effect*. This takes place during the earlier phases of learning when the network is still relatively undifferentiated. The desired mapping is assumed to involve several subtasks which must be performed by relatively distinct subnets. All subnets initially receive similar error information and may therefore try to take on the same task at the same time; ie. the subnets may respond as a herd. Again, unfruitful oscillations result.

The moving target problem and the herd effect provide additional arguments in favour of using dynamic-network learning schemes. However, it is important to note that dynamic-network learning brings with it its own problems and effects. In particular the *incremental* dynamic-network learning schemes mentioned above (ie. schemes in which units are added one at a time) are associated with what is here called the *howl effect*. The name derives from the well-known story of the electronics engineer who, having caused a radio to emit a high-pitched howl by removing an arbitrary component, concluded (wrongly) that the component must have been a 'howl-suppressor'. The moral of the story is that the functional role of a component may be quite different depending on whether it is treated as an independent entity or as a single component of a complex system. In incremental learning schemes where units are introduced one at a time we necessarily treat each unit as an independent entity. If the nature of the task is such that a set of units are required to work in concert then an incremental, one-at-a-time scheme is unlikely to be effective.

Example

A simple illustrative example involves the 8-unit encoder problem [5]]. In this problem there are 8 input units and 8 output units. The task is to reproduce the pattern of input activation at the output units given the constraint that only one input unit is turned on at a time. In previous work this problem has been studied in the context of a static network with a fixed number of hidden units (3 being the minimum required for a perfect solution). However, consider the case where we attempt to derive a network for this problem using an incremental scheme. Initially we will add and seek to train up a single hidden unit. Because of the simplicity of the problem we know that in an optimal solution this unit must serve to represent one bit in a binary encoding of the input vector. But it is most unlikely that a learning scheme applied to this single unit will lead to this result since a single bit of a 3-bit binary encoding serves as an extremely poor indication of what the input was.

In this case, then, it is likely that a simple-minded incremental scheme would not lead to a good solution to the problem. We know that in a good solution several units must act in concert to produce a particular type of encoding of the input. If we attempt to add and train hidden units one at a time we will miss the good solution. An obvious response to this problem is to investigate schemes which allow the algorithm to introduce units in small groups, ie. to introduce complete subnets into the network. However, this really only solves half the problem since the question still remains of how we should train each of the units in the introduced subnetwork.

The problem really is to do with the fact that current network learning algorithms — even the newer constructive ones — have a '1-unit fixation'. It is an intrinsic part of their approach that units are treated as independent entities. They are thus unlikely to play a role in any genuine solution to the howl effect. Whether this is likely to be a severe restriction in practice is hard to decide. At present it is difficult to turn up a decent example which demonstrates the howl effect actually having a deleterious impact on network learning. (The lack of an example in [3] to illustrate the herd effect suggests that examples of that phenomenon are hard to come by too.) But this may simply be a result of the fact that databases tend to be produced in a way that maximizes attribute independence. As a point of note, I have not been able to find a single database in the UCI repository of machine learning databases which features non-independent attributes.

Concluding comments

The observation that some learning problem may require subnets of hidden units to act in concert leads us to conclude that constructive algorithms should, ideally, have the capacity to introduce groups of units and to train them collectively rather than individually. However, at present, none of the better known constructive algorithms can actually do this. Experience suggests, however, that the sorts of learning problems which are conventionally used in empirical work do not actually necessitate collective training of units. Until the field gets more firmly to grips with the problem of learning from real environments, taking as input low-level, noisy data, it is hard to draw firm conclusions.

References

[1] Mezard, M. and Nadal, J. (1989). Learning in feedforward layered networks: the tiling algorithm. *J. Phys. A: Math. Gen.*, *22* (pp. 2191-2203).

[2] Frean, M. (1989). *The Upstart Algorithm: A Method for Constructing and Training Feed-Forward Neural Networks*. Edinburgh Physics Preprint 89/479, Dept. of Physics, University of Edinburgh.

[3] Fahlman, S. and Lebiere, C. (1990). The cascade-correlation learning architecture. In D.S. Touretzky (Ed.), *Advances in Neural Information Processing Systems 2* (pp. 524-532.). Morgan Kaufmann Publishers, Los Altos CA.

[4] Hinton, G. (1989). Connectionist learning procedures. *Artificial Intelligence, 40* (pp. 185-234).

[5] Hinton, G., Sejnowski, T. and Ackley, D. (1984). *Boltzmann Machines: Constraint Satisfaction Networks that Learn.* Technical Report CMU-CS-84-119, Carnegie-Mellon University.