Limited Fan-in Random Wired Cascade-Correlation

Henrik Klagges and Michael Soegtrop
IBM Research Division
Physics Group Munich
Schellingstrasse 4/III
D–8000 Muenchen 40
Federal Republic of Germany
E-mail: henrik@robots.ox.ac.uk, uh311ae@sun1.lrz-muenchen.de

1 Abstract

The success of new learning algorithms like Cascade Correlation (CC) lies partly in topology construction strategies which are difficult to map onto SIMD-parallel neurocomputers. A CC variation that limits the connection fan-in and random-wires the neurons was invented to ease the SIMD-implementation. Surprisingly, the method produced superior and very compact networks with improved generalization. In particular, solutions of the 2-spirals problem improved from 133+−27 total weights for standard CC down to 60+−10 with 75% less connection crossings. Performance increased with candidate pool size and was correlated with a reduction of artefacts in the receptive field visualizations. We argue that, for general neural network learning, construction algorithms are as important as weight adaption rules. This requires sparse matrix support from neurocomputer hardware.

2 Introduction

Current special-purpose neurocomputers are typically tailored to train regular multi-layer perceptrons with the backpropagation learning rule. However, new algorithms like CC running on a standard workstation can reach comparable or better learning performance on selected problems, not least of which is 2-spirals [1]. We focus on one of CC's key features, its incremental network building strategy. Starting with a minimal inputs-outputs network, additional hidden units get selected out of a pool of competing candidates, which are each connected to all previously installed units. They are then – one at a time – added to the network, and made static (frozen). One way to parallelize this algorithm is to distribute the training of each candidate onto a separate processing node, thus keeping as many physical nodes busy as there are candidates. However, our experiments suggest it is not clear that significant speed-ups result from large candidate numbers. All candidates receive the same inputs and thus the same raw information, and differ only in the random initialization of their weights. This limits the usefulness of many candidates or multiple (n-best) candidate installation because of their redundancy.

3 Limited Fan-In Random-Wired CC

A second obstacle to parallelization is the highly sequential nature of the deep neuron cascade created by CC, because no unit can finish its netinput calculation until all units lower in the cascade have. This is equivalent to the n-th hidden unit having a connection fan-in of $O(n)$. Therefore, the classical inner loop of neural network calculations – the loop over the input and weights of each unit – would be of different length for each unit, resulting in serious load unbalancing in a SIMD implementation. However, this can be attacked by limiting the fan-in of hidden units to a constant c. In this case we are left with the question to *which* c of the n already installed units the new unit/candidates should be connected (if $n > c$). A simple heuristic to connect new hidden units is to random-wire them. Thus a candidate pool now contains candidates that have different, random connection patterns. This constant unit fan-in removes unit-dependent memory layout, loop control, and other obstacles for a SIMD-implementation. Moreover, running very large candidate pools gets useful. Candidates with different connection combinations are less redundant, permitting larger searches and n-best candidate installation.

3.1 Simulation results

N-Limited fan-in random-wired CC (n-LFCC), invented only to simplify parallel implementation, showed good performance compared to standard CC on the 2-spirals and 3-discs [2] problems [1].

algorithm	training set performance	test set performance	total no. of weights	total no. of hidden units	connection crossings
2-spirals:					
2-LFCC	100%	$(97+-1)\%$	$60+-11$	$20.2+-3.7$	$(1.0+-.2)e8$
CC	100%	$(94+-2)\%$	$133+-27$	$13.0+-1.6$	$(4.0+-.7)e8$
3-discs:					
2-LFCC	100%	$(87+-7)\%$	$221+-16$	$73+-5.3$	$(3.4+-.3)e9$
CC	100%	$(75+-9)\%$	$1125+-132$	$45+-3.55$	n/a

2-LFCC produced compact[2], well generalizing networks quickly (see figure). Smaller networks showed better generalization, hinting that the higher the information compression ratio, the better the learning. A large candidate pool (64-256) was helpful, otherwise inferior candidates were frequently frozen and installed into the network. This phenomenon affects CC badly and low fan-in LFCC even more. It is interesting to note that bad frozen units caused patchy or jagged artefacts in the receptive field visualizations. It is also plausible that an algorithm that installs *permanent* feature detectors needs to install *good* ones - and a small pool decreases the odds for this considerably.

3.2 Future Work

The number of possible m-fan-in combinations out of n neurons increases as nCm. Thus for constant candidate pool size the fraction of combination space that can be searched by the random-wired method decreases quickly. This is a typical characteristic of random-based optimizers and results in bad scaling behaviour. A speculative solution to this is a genetic algorithm (GA), which is a proven heuristic to navigate through large search spaces. The candidate pool is treated as a population of individuals with their connection lists as genomes and and their correlation as fitness values. Thus, the GA operations reduce to only the recombination and mutation stages, with almost no computation and memory overhead over LFCC. However, our initial experiments suggest no advantage for the GA-based topology constructor, at least for small problems.

4 Conclusion

We argue that LFCC is an interesting variant of CC. Furthermore, we think that topology construction is of equal importance for neural network learning as weight adaption. Therefore, future neurocomputers should include support for *dynamically changing random-wired topologies*. Interfacing topology construction strategies to advanced searchers opens up a new frontier of research.

Acknowledgments & references:
We based our code on Scott Crowder's (CMU) public domain CC-code. Scott Fahlman (CMU) was always friendly and discussed loads of questions.
(1) Fahlman, S. and Lebiere, C. (1990). "The Cascade-Correlation Learning Architecture", in Touretzky, D.S. (ed.), NIPS 2, Morgan Kaufman
(2) Sjogaard, S. (1991). "A Conceptual Approach to Generalization in Dynamic Neural Networks", Technical report of the Computer Science Department of the University of Aarhus, Denmark (available via anonymous ftp from ftp.cis.ohio-state.edu, /pub/neuroprose)

[1] In the case of the 3-disc/CC solution, longint arithmetic overflow invalidated the connection crossings counter.
[2] The $60+-11$ weights for 2-spirals are the best published numbers we are aware of, with CC itself as the next best solution.