Computational Complexity Problems in Neural Associative Memories

Patrik Floréen

Ph.D. dissertation

Department of Computer Science, University of Helsinki

Report A-1992-5, Helsinki 1992, 128 + 8 pages

ISSN 0781-6855 ISBN 951-45-6296-8

Abstract

Neural associative memories are memory constructs consisting of a weighted network with a large number of simple nodes, whose computations collectively result in retrieving stored information when partial or distorted information is given as input. We study two types of neural associative memories: the Hopfield and the Hamming memories.

For the Hopfield memory, we first study worst-case bounds on the convergence times in both asynchronous and synchronous updating. However, the main theme in our study of Hopfield networks is the computational complexity of assessing the memory quality of a given network. We show that (assuming $P \neq NP$) it is hard to count the number of stable vectors, to compute the size of attraction domains (in synchronous updating), and to compute the attraction radii (in synchronous updating). It is hard even to approximate the attraction radii, unless $P = NP$.

For the Hamming memory, we study which conditions the parameters of the network must satisfy in order to ensure convergence and we obtain a tight bound on the convergence time. With an optimal choice of parameter values, the worst-case convergence time is $\Theta(p \log(pn))$, where p is the memory capacity and n is the vector length. Restricting the instances to those with a unique nearest stored vector to the input vector, the worst-case convergence time is $\Theta(p \log n)$. For random vectors, the probability for such instances to occur approaches 1 as n grows, if the number of stored vectors grows subexponentially in n. By dynamically changing the connection weights and by taking powers of the activity levels, the convergence is speeded up considerably. Finally, we present a new memory model based on the Hamming memory, but with a feed-forward network structure made possible by the use of more complicated node functions. In simulations, we show that the new memory model behaves as expected.

Throughout the study, we also explore the consequences of allowing "don't know" elements in both the input and the output. Experiments show that "don't know" elements are effective especially in the input.

Computing Reviews (1991) Categories and Subject Descriptors:

B.3.2 [Memory Structures]:
Design Styles — *Associative Memories*

C.1.3 [Processor Architectures]:
Other Architecture Styles — *Neural Nets*

F.1.1 [Computation by Abstract Devices]:
Models of Computation — *Self-modifying machines (e.g., neural networks)*

F.1.3 [Computation by Abstract Devices]:
Complexity Classes — *Reducibility and Completeness*

General Terms:
Algorithms, Theory.

Additional Key Words and Phrases:
Associative memory, neural networks, computational complexity.

Acknowledgements

I am greatly indebted to my advisor, Doctor Pekka Orponen, for his guidance of my studies and for promoting my scientific career in form of productive research cooperation, good financial arrangements and deep interest in my work. Without his support, this thesis would never have been written. I am grateful to Professor Esko Ukkonen for his valuable comments on my work and for his continued support during my graduate studies. I thank Doctor Taneli Huuskonen for useful and productive discussions on one of my problems.

I am grateful to Professor Martti Tienari for providing me with good working conditions at the Department of Computer Science. During the period of this research, I have been financially supported by the Academy of Finland, the University of Helsinki, and Svenska Vetenskapliga Centralrådet.

Finally, I wish to express my gratitude to all those who have enriched my life during studies and research.