

Hopfield Neural Networks

BOOK

- I Neural Networks: Methodology and Applications
- II Elements of Artificial Neural Networks
- III Handbook of Brain Theory and Neural Networks
- IV Principles of artificial neural networks
- V Neural Networks and Statistical Learning

- VI An introduction to Neural Networks
- Introduction to the Theory of Neural Computation

Hopfield Neural Networks

- I CHP8 Neural Networks without Training for Optimization
- II CHP 6-7
- IV CHP 7
- V CHP 6-7

6	Hopfield Networks, Simulated Annealing, and Chaotic Neural Networks
6.1	Hopfield Model
6.2	Continuous-Time Hopfield Network
6.3	Simulated Annealing
6.4	Hopfield Networks for Optimization
6.4.1	Combinatorial Optimization Problems
6.4.2	Escaping Local Minima for Combinatorial Optimization Problems
6.4.3	Solving Other Optimization Problems
6.5	Chaos and Chaotic Neural Networks
6.5.1	Chaos, Bifurcation, and Fractals
6.5.2	Chaotic Neural Networks
6.6	Multistate Hopfield Networks
6.7	Cellular Neural Networks
	References



7	Associative Memory Networks	187
7.1	Introduction	187
7.2	Hopfield Model: Storage and Retrieval	189
7.2.1	Generalized Hebbian Rule	189
7.2.2	Pseudoinverse Rule	191
7.2.3	Perceptron-Type Learning Rule	191
7.2.4	Retrieval Stage	192
7.3	Storage Capability of the Hopfield Model	193
7.4	Increasing Storage Capacity	197
7.5	Multistate Hopfield Networks for Associative Memory	200
7.6	Multilayer Perceptrons as Associative Memories	201
7.7	Hamming Network	203

Applications

- Graph-Bisection
- linear programming
- TSP
- Associative Memory

Simple “Neural” Optimization
Networks: An A/D Converter,
Signal Decision Circuit,
and a Linear Programming Circuit

THE LINEAR PROGRAMMING NETWORK

IV. THE LINEAR PROGRAMMING NETWORK

The linear programming problem can be stated as the attempt to minimize a cost function

$$\pi = \vec{A} \cdot \vec{V} \quad (14)$$

where \vec{A} is an N -dimensional vector of coefficients for the N variables which are the components of \vec{V} . This minimization is to be accomplished subject to a set of M linear

$$\vec{D}_j \cdot \vec{V} \geq B_j, \quad j = 1, \dots, M$$

$$\vec{D}_j = \begin{bmatrix} D_{j1} \\ D_{j2} \\ \vdots \\ D_{jN} \end{bmatrix}$$

(15)

