

Neural Networks

Methodology and Applications

Bearbeitet von
Gérard Dreyfus

1. Auflage 2005. Buch. xviii, 498 S. Hardcover

ISBN 978 3 540 22980 3

Format (B x L): 15,5 x 23,5 cm

Gewicht: 1980 g

[Weitere Fachgebiete > Mathematik > Numerik und Wissenschaftliches Rechnen > Angewandte Mathematik, Mathematische Modelle](#)

schnell und portofrei erhältlich bei


DIE FACHBUCHHANDLUNG

Die Online-Fachbuchhandlung beck-shop.de ist spezialisiert auf Fachbücher, insbesondere Recht, Steuern und Wirtschaft. Im Sortiment finden Sie alle Medien (Bücher, Zeitschriften, CDs, eBooks, etc.) aller Verlage. Ergänzt wird das Programm durch Services wie Neuerscheinungsdienst oder Zusammenstellungen von Büchern zu Sonderpreisen. Der Shop führt mehr als 8 Millionen Produkte.

Contents

<i>List of Contributors</i>	xvii
1 Neural Networks: An Overview	
<i>G. Dreyfus</i>	1
1.1 Neural Networks: Definitions and Properties	2
1.1.1 Neural Networks	3
1.1.2 The Training of Neural Networks	12
1.1.3 The Fundamental Property of Neural Networks with Supervised Training: Parsimonious Approximation	13
1.1.4 Feedforward Neural Networks with Supervised Training for Static Modeling and Discrimination (Classification)	15
1.1.5 Feedforward Neural Networks with Unsupervised Training for Data Analysis and Visualization	21
1.1.6 Recurrent Neural Networks for Black-Box Modeling, Gray-Box Modeling, and Control	22
1.1.7 Recurrent Neural Networks Without Training for Combinatorial Optimization	23
1.2 When and How to Use Neural Networks with Supervised Training ..	24
1.2.1 When to Use Neural Networks?	24
1.2.2 How to Design Neural Networks?	25
1.3 Feedforward Neural Networks and Discrimination (Classification) ..	32
1.3.1 What Is a Classification Problem?	33
1.3.2 When Is a Statistical Classifier such as a Neural Network Appropriate?	33
1.3.3 Probabilistic Classification and Bayes Formula	36
1.3.4 Bayes Decision Rule	41
1.3.5 Classification and Regression	43
1.4 Some Applications of Neural Networks to Various Areas of Engineering	50
1.4.1 Introduction	50

1.4.2	An Application in Pattern Recognition: The Automatic Reading of Zip Codes	51
1.4.3	An Application in Nondestructive Testing: Defect Detection by Eddy Currents	55
1.4.4	An Application in Forecasting: The Estimation of the Probability of Election to the French Parliament	56
1.4.5	An Application in Data Mining: Information Filtering	57
1.4.6	An Application in Bioengineering: Quantitative Structure-Relation Activity Prediction for Organic Molecules	62
1.4.7	An Application in Formulation: The Prediction of the Liquidus Temperatures of Industrial Glasses	64
1.4.8	An Application to the Modeling of an Industrial Process: The Modeling of Spot Welding	65
1.4.9	An Application in Robotics: The Modeling of the Hydraulic Actuator of a Robot Arm	68
1.4.10	An Application of Semiphsical Modeling to a Manufacturing Process	70
1.4.11	Two Applications in Environment Control: Ozone Pollution and Urban Hydrology	71
1.4.12	An Application in Mobile Robotics	75
1.5	Conclusion	76
1.6	Additional Material	77
1.6.1	Some Usual Neurons	77
1.6.2	The Ho and Kashyap Algorithm	79
References		80

2 Modeling with Neural Networks: Principles and Model Design Methodology

<i>G. Dreyfus</i>	85
2.1 What Is a Model?	85
2.1.1 From Black-Box Models to Knowledge-Based Models	85
2.1.2 Static vs. Dynamic Models	86
2.1.3 How to Deal With Uncertainty? The Statistical Context of Modeling and Machine Learning	86
2.2 Elementary Concepts and Vocabulary of Statistics	87
2.2.1 What is a Random Variable?	87
2.2.2 Expectation Value of a Random Variable	89
2.2.3 Unbiased Estimator of a Parameter of a Distribution	89
2.2.4 Variance of a Random Variable	90
2.2.5 Confidence Interval	92
2.2.6 Hypothesis Testing	92
2.3 Static Black-Box Modeling	92
2.3.1 Regression	93
2.3.2 Introduction to the Design Methodology	94
2.4 Input Selection for a Static Black-Box Model	95

2.4.1	Reduction of the Dimension of Representation Space	95
2.4.2	Choice of Relevant Variables	96
2.4.3	Conclusion on Variable Selection	103
2.5	Estimation of the Parameters (Training) of a Static Model	103
2.5.1	Training Models that are Linear with Respect to Their Parameters: The Least Squares Method for Linear Regression	106
2.5.2	Nonadaptive (Batch) Training of Static Models that Are Not Linear with Respect to Their Parameters	110
2.5.3	Adaptive (On-Line) Training of Models that Are Nonlinear with Respect to Their Parameters	121
2.5.4	Training with Regularization	121
2.5.5	Conclusion on the Training of Static Models	130
2.6	Model Selection	131
2.6.1	Preliminary Step: Discarding Overfitted Model by Computing the Rank of the Jacobian Matrix	133
2.6.2	A Global Approach to Model Selection: Cross-Validation and Leave-One-Out	134
2.6.3	Local Least Squares: Effect of Withdrawing an Example from the Training Set, and Virtual Leave-One-Out	137
2.6.4	Model Selection Methodology by Combination of the Local and Global Approaches	142
2.7	Dynamic Black-Box Modeling	149
2.7.1	State-Space Representation and Input-Output Representation	150
2.7.2	Assumptions on Noise and Their Consequences on the Structure, the Training and the Operation of the Model	151
2.7.3	Nonadaptive Training of Dynamic Models in Canonical Form	162
2.7.4	What to Do in Practice? A Real Example of Dynamic Black-Box Modeling	168
2.7.5	Casting Dynamic Models into a Canonical Form	171
2.8	Dynamic Semiphenical (Gray Box) Modeling	175
2.8.1	Principles of Semiphenical Modeling	175
2.9	Conclusion: What Tools?	186
2.10	Additional Material	187
2.10.1	Confidence Intervals: Design and Example	187
2.10.2	Hypothesis Testing: An Example	189
2.10.3	Pearson, Student and Fisher Distributions	189
2.10.4	Input Selection: Fisher's Test; Computation of the Cumulative Distribution Function of the Rank of the Probe Feature	190
2.10.5	Optimization Methods: Levenberg-Marquardt and BFGS	193
2.10.6	Line Search Methods for the Training Rate	195
2.10.7	Kullback-Leibler Divergence Between two Gaussians	196
2.10.8	Computation of the Leverages	197
	References	199

3 Modeling Methodology: Dimension Reduction and Resampling Methods

<i>J.-M. Martinez</i>	203
3.1 Introduction	203
3.2 Preprocessing	204
3.2.1 Preprocessing of Inputs	204
3.2.2 Preprocessing Outputs for Supervised Classification	205
3.2.3 Preprocessing Outputs for Regression	206
3.3 Input Dimension Reduction	207
3.4 Principal Component Analysis	207
3.4.1 Principle of PCA	207
3.5 Curvilinear Component Analysis	211
3.5.1 Formal Presentation of Curvilinear Component Analysis ...	213
3.5.2 Curvilinear Component Analysis Algorithm	215
3.5.3 Implementation of Curvilinear Component Analysis	216
3.5.4 Quality of the Projection	217
3.5.5 Difficulties of Curvilinear Component Analysis	218
3.5.6 Applied to Spectrometry	219
3.6 The Bootstrap and Neural Networks	220
3.6.1 Principle of the Bootstrap	222
3.6.2 Bootstrap Estimation of the Standard Deviation	223
3.6.3 The Generalization Error Estimated by the Bootstrap	224
3.6.4 The NeMo Method	225
3.6.5 Testing the NeMo Method	227
3.6.6 Conclusions	229
References	230

4 Neural Identification of Controlled Dynamical Systems and Recurrent Networks

<i>M. Samuelides</i>	231
4.1 Formal Definition and Examples of Discrete-Time Controlled Dynamical Systems	232
4.1.1 Formal Definition of a Controlled Dynamical System by State Equation	232
4.1.2 An Example of Discrete Dynamical System	233
4.1.3 Example: The Linear Oscillator	234
4.1.4 Example: The Inverted Pendulum	235
4.1.5 Example of Nonlinear Oscillator: The Van Der Pol Oscillator	236
4.1.6 Markov Chain as a Model for Discrete-Time Dynamical Systems with Noise	236
4.1.7 Linear Gaussian Model as an Example of a Continuous-State Dynamical System with Noise	239
4.1.8 Auto-Regressive Models	240
4.1.9 Limits of Modeling Uncertainties Using State Noise	242
4.2 Regression Modeling of Controlled Dynamical Systems	242

4.2.1	Linear Regression for Controlled Dynamical Systems	242
4.2.2	Nonlinear Identification Using Feedforward Neural Networks	246
4.3	On-Line Adaptive Identification and Recursive Prediction Error Method	250
4.3.1	Recursive Estimation of Empirical Mean	250
4.3.2	Recursive Estimation of Linear Regression	252
4.3.3	Recursive Identification of an AR Model	253
4.3.4	General Recursive Prediction Error Method (RPEM)	255
4.3.5	Application to the Linear Identification of a Controlled Dynamical System	256
4.4	Innovation Filtering in a State Model	258
4.4.1	Introduction of a Measurement Equation	258
4.4.2	Kalman Filtering	261
4.4.3	Extension of the Kalman Filter	265
4.5	Recurrent Neural Networks	270
4.5.1	Neural Simulator of an Open-Loop Controlled Dynamical System	270
4.5.2	Neural Simulator of a Closed Loop Controlled Dynamical System	270
4.5.3	Classical Recurrent Network Examples	272
4.5.4	Canonical Form for Recurrent Networks	275
4.6	Learning for Recurrent Networks	276
4.6.1	Teacher-Forced Learning	277
4.6.2	Unfolding of the Canonical Form and Backpropagation Through Time (BPTT)	277
4.6.3	Real-Time Learning Algorithms for Recurrent Network (RTRL)	281
4.6.4	Application of Recurrent Networks to Measured Controlled Dynamical System Identification	282
4.7	Appendix (Algorithms and Theoretical Developments)	283
4.7.1	Computation of the Kalman Gain and Covariance Propagation	283
4.7.2	The Delay Distribution Is Crucial for Recurrent Network Dynamics	285
	References	287
 5 Closed-Loop Control Learning		
	<i>M. Samuelides</i>	289
5.1	Generic Issues in Closed-Loop Control of Nonlinear Systems	290
5.1.1	Basic Model of Closed-Loop Control	290
5.1.2	Controllability	291
5.1.3	Stability of Controlled Dynamical Systems	292
5.2	Design of a Neural Control with an Inverse Model	294
5.2.1	Straightforward Inversion	294
5.2.2	Model Reference Adaptive Control	297

5.2.3	Internal Model Based Control	299
5.2.4	Using Recurrent Neural Networks	301
5.3	Dynamic Programming and Optimal Control	303
5.3.1	Example of a Deterministic Problem in a Discrete State Space	303
5.3.2	Example of a Markov Decision Problem	305
5.3.3	Definition of a Decision Markov Problem	307
5.3.4	Finite Horizon Dynamic Programming	310
5.3.5	Infinite-Horizon Dynamic Programming with Discounted Cost	312
5.3.6	Partially Observed Markov Decision Problems	314
5.4	Reinforcement Learning and Neuro-Dynamic Programming	314
5.4.1	Policy Evaluation Using Monte Carlo Method and Reinforcement Learning	314
5.4.2	TD Algorithm of Policy Evaluation	316
5.4.3	Reinforcement Learning: Q-Learning Method	319
5.4.4	Reinforcement Learning and Neuronal Approximation	322
	References	325
6 Discrimination		
	<i>M. B. Gordon</i>	329
6.1	Training for Pattern Discrimination	330
6.1.1	Training and Generalization Errors	331
6.1.2	Discriminant Surfaces	332
6.2	Linear Separation: The Perceptron	334
6.3	The Geometry of Classification	336
6.3.1	Separating Hyperplane	336
6.3.2	Aligned Field	337
6.3.3	Stability of an Example	338
6.4	Training Algorithms for the Perceptron	339
6.4.1	Perceptron Algorithm	339
6.4.2	Convergence Theorem for the Perceptron Algorithm	341
6.4.3	Training by Minimization of a Cost Function	342
6.4.4	Cost Functions for the Perceptron	344
6.4.5	Example of Application: The Classification of Sonar Signals	351
6.4.6	Adaptive (On-Line) Training Algorithms	353
6.4.7	An Interpretation of Training in Terms of Forces	353
6.5	Beyond Linear Separation	355
6.5.1	Spherical Perceptron	355
6.5.2	Constructive Heuristics	356
6.5.3	Support Vector Machines (SVM)	359
6.6	Problems with More than two Classes	362
6.7	Theoretical Questions	364
6.7.1	The Probabilistic Framework	364

6.7.2	A Probabilistic Interpretation of the Perceptron Cost Functions	366
6.7.3	The Optimal Bayesian Classifier	368
6.7.4	Vapnik's Statistical Learning Theory	369
6.7.5	Prediction of the Typical Behavior	372
6.8	Additional Theoretical Material	374
6.8.1	Bounds to the Number of Iterations of the Perceptron Algorithm	374
6.8.2	Number of Linearly Separable Dichotomies	375
	References	376

7 Self-Organizing Maps and Unsupervised Classification

	<i>F. Badran, M. Yacoub, and S. Thiria</i>	379
7.1	Notations and Definitions	381
7.2	The k -Means Algorithm	383
7.2.1	Outline of the k -Means Algorithm	383
7.2.2	Stochastic Version of k -Means	386
7.2.3	Probabilistic Interpretation of k -Means	388
7.3	Self-Organizing Topological Maps	392
7.3.1	Self-Organizing Maps	392
7.3.2	The Batch Optimization Algorithm for Topological Maps ..	397
7.3.3	Kohonen's Algorithm	404
7.3.4	Discussion	406
7.3.5	Neural Architecture and Topological Maps	406
7.3.6	Architecture and Adaptive Topological Maps	408
7.3.7	Interpretation of Topological Self-Organization	409
7.3.8	Probabilistic Topological Map	412
7.4	Classification and Topological Maps	415
7.4.1	Labeling the Map Using Expert Data	416
7.4.2	Searching a Partition that Is Appropriate to the Classes ..	417
7.4.3	Labeling and Classification	420
7.5	Applications	421
7.5.1	A Satellite Remote Sensing Application	422
7.5.2	Classification and PRSOM	430
7.5.3	Topological Map and Documentary Research	439
	References	441

8 Neural Networks without Training for Optimization

	<i>L. Hérault</i>	443
8.1	Modelling an Optimisation Problem	443
8.1.1	Examples	444
8.1.2	The Travelling Salesman Problem (TSP)	445
8.2	Complexity of an Optimization Problem	446
8.2.1	Example	447
8.3	Classical Approaches to Combinatorial Problems	447

8.4	Introduction to Metaheuristics	448
8.5	Techniques Derived from Statistical Physics.....	449
8.5.1	Canonical Analysis	450
8.5.2	Microcanonical Analysis	456
8.5.3	Example: Travelling Salesman Problem	457
8.6	Neural Approaches	463
8.6.1	Formal Neural Networks for Optimization	463
8.6.2	Architectures of Neural Networks for Optimisation	465
8.6.3	Energy Functions for Combinatorial Optimisation	466
8.6.4	Recurrent Hopfield Neural Networks	467
8.6.5	Improvements of Hopfield Neural Networks	475
8.7	Tabu Search	484
8.8	Genetic Algorithms	484
8.9	Towards Hybrid Approaches	485
8.10	Conclusion	485
8.10.1	The Choice of a Technique	485
	References	486
	<i>About the Authors</i>	491
	Index	493