

Contents

1	Abstract	4
2	Motivation	5
2.1	Historical Background and Bibliographical Notes	5
3	Neural Networks	6
3.1	The Perceptron	6
3.2	Activation Functions	7
3.3	Layered Networks	8
3.4	Feed-Forward Networks	8
3.5	Multilayer Perceptrons	9
3.6	Expressive Power	9
4	Network Training	11
4.1	Error Measures	11
4.2	Training Approaches	12
4.3	Parameter Optimization	12
4.3.1	Linear Units	12
4.3.2	Weight Initialization	13
4.3.3	Gradient Descent	13
4.3.4	Momentum	13
4.3.5	Enhanced Gradient Descent	14
4.3.6	Newton's Method	14
4.4	Error Backpropagation	15
4.4.1	Efficiency	17
4.5	The Hessian	17
4.5.1	Efficiency	19
4.6	Regularization	19
4.6.1	L2-Regularization	19
4.6.2	Early Stopping	20
5	Pattern Classification	21
5.1	Statistical Background	21
5.2	Bayes' Decision Rule	21
5.3	Maximum Likelihood Estimation	21
5.3.1	Derivation of Cross-Entropy	22
5.4	Application: Recognizing Handwritten Digits	23
5.4.1	Matrix Notation	23
5.4.2	Implementation and Results	24
6	Conclusion	25
	Appendices	26

<i>LIST OF TABLES</i>	3
-----------------------	---

A MatLab Implementation	26
A.1 Logistic Sigmoid and its Derivative	26
A.2 Training Procedure	26
A.3 Validation Procedure	28
Literature	30

List of Tables

List of Figures

1	Single processing units and its components.	6
2	Network graph of a perceptron with D input units and C output units. . .	6
3	The logistic sigmoid as activation function.	7
4	Network graph for a two-layer perceptron with C input units, D output units and m hidden units.	9
5	Single-layer perceptron for modeling boolean AND.	10
6	The learning rate and its influence on the rate of convergence.	14
7	Backpropagation of errors through the network.	16
8	Exact evaluation of the hessian.	17
9	Early stopping based on a validation set.	20
10	Error on the training set during training.	23
11	Results of training a two-layer perceptron using the MNIST dataset.	24