LEARNING RELIABILITY:

a study on indecisiveness in sample selection

CIP–GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Barakova, Emilia Ivanova

Learning Reliability:
a study on indecisiveness in sample selection – / E.I. Barakova
Proefschrift Rijksuniversiteit Groningen

Keywords: information technology, neural networks, reliability, symmetry, knowledge engineering.

ISBN 90–367–0987–3

PrintPartners Ipskamp B. V.
Capitool 25 Business & Science Park
Postbus 333
7500 AH Enschede

Beoordelingscommissie: Prof.dr.ir. W.M.G. van Bokhoven
Prof.dr.–ing. A. Kistner
Prof.dr.ir. J. Nerbonne

This thesis is based on the following publications:

1. (Barakova, E.I., and Spaanenburg, L.) *Learning and reproducing*, in: (Spaanenburg, L. et al.) V−Annals II (Shanker Publ., Maastricht), 1999.

2. (Barakova, E.I., and Spaanenburg, L.) *Windowed Active Sampling for Reliable Neural Learning*, Journal of Systems Architecture **44**, No. 8 (Elsevier Scientific Publ., Amsterdam, 1998) pp. 635−650.

3. (Barakova, E.I. and Spaanenburg, L.) *Symmetry: Between Indecision and Equality of Choice*, pp. 903−912, in: (Mira, J., Moreno−Diaz, R., Cabestany, J.) *Biological and Artificial Computation: From Neuroscience to Technology*, Lecture Notes in Computer Science **1240** (Springer Verlag, Berlin) 1997.

4. (Barakova, E.I., and Spaanenburg, L.) *Selective Sampling for High Reliability in Neural Signal Approximation*, Proceedings NICROSP'96 (Venice, August 1996) pp. 183−193.

5. (Barakova, E.I., Spaanenburg, L., and Zaprjanov, J.) *Neural fault diagnosis of a turbogenerator by vibroacoustic data*, Int. Conf. on Signal Processing, Applications & Technology ICSPAT'95 (Boston, MA, USA, 24−26 October 1995) pp. 1454−1458.

Preface.

The design of a product is based on the assumption of how it will be used. Conversely, the product is usually good for only such usage as was assumed during its conception. In a classical sense, the implicit assumption brings an explicit specification from which the design is derived. More often than not, the specification is therefore the starting point of a hopefully structured and well–behaved, but eventually mechanical design effort. Where the customer tends to learn from the design and mandates to change and/or augment the specification during the process, the project planning gets invalidated. Current practice is therefore to fix the specification in advance, for instance by contract.

The interest in Artificial Neural Networks (ANN) is founded on their ability to learn from examples, as derived from the environment in which the product will operate, instead of being designed from an hypothesis about the operation. It is commonly agreed that learning is based on *memorization* (associating or mapping a set of questions to their answers) and *generalization* (the ability to answer new questions about the same problem). As such, ANNs promise a perfect fit to their intended usage. But circumstantial evidence still does not equal a witness observation. Despite its historic fame, an Artificial Neural Network will not learn all, let alone under all circumstances. This is probably the most striking difference with a designed product: there will never be a proof by construction!

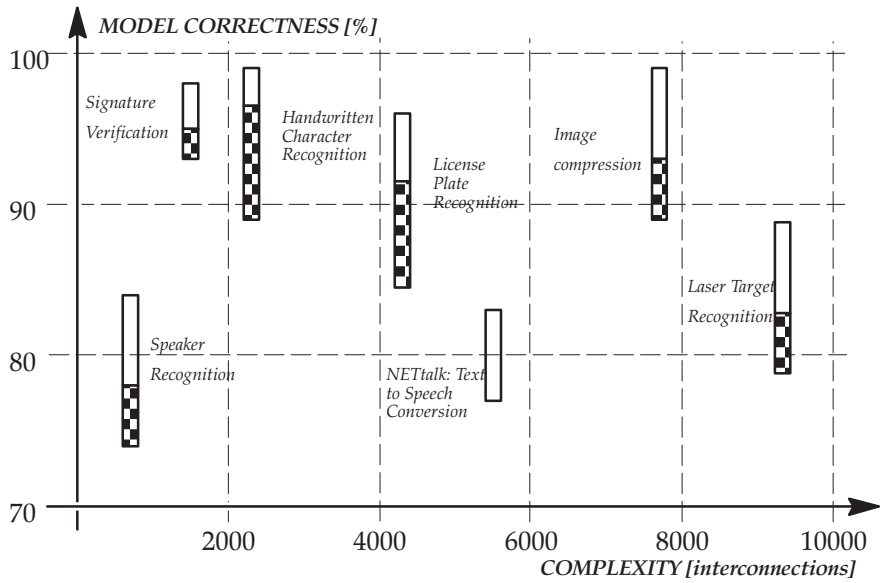

Figure 1: *Overview of real–world neural applications.*

With the coming of age of neural technology, an impressive number of neural products have found their way to the market place [88]. Some popular applications are indicated in figure 1, which position them in the area spanned by computational complexity and

model correctness. The bars indicate the achieved performance: the patterns within the bar indicate the widely achieved results, since the white part stands for the best results in the area. Clearly, none of them achieves a 100% correct functionality. It appears, that for each application a bottom level of functionality can be reached almost without any effort. However, to go beyond requires special attention and has therefore spurred a lot of research to develop new algorithms, to construct alternative architectures, to provide different settings of input parameters or to preprocess input data.

To achieve a product of ultimate performance, two methods can be devised: (a) its function is based on a provably correct algorithm, and (b) an effective redundancy is to be incorporated in the underlying algorithm. As far as ANNs are constructed from analysis of noisy data, they can entirely be considered as systems of the second type. Because statistics is concerned with data analysis as well, there is a considerable overlap between the fields of neural networks and statistics. To analyze learning and generalization of neural networks from noisy/randomized data, statistical inference can also be used.

Performance enhancement can be created by a kind of majority voting. This principle suggests that, instead of providing one neural network solution to a problem, a set of neural networks can be combined to form a neural net system which performs better than any of the networks on its own [116] [138]. The conclusion made in [112] is that mere redundancy does not necessarily increase reliability. Empirically it is common practice to train many different candidate networks to select the winner on basis of predefined criteria. A disadvantage of this method is that training of the losing networks does not help in a further development. Another weak point is that the criterion for choosing the best network is usually the performance on a validation set, which can not guarantee the modeling quality of the underlying data generator. But when the networks are incomplete versions of the same functionality, the combination might raise the functional correctness to a higher level (Figure 2).

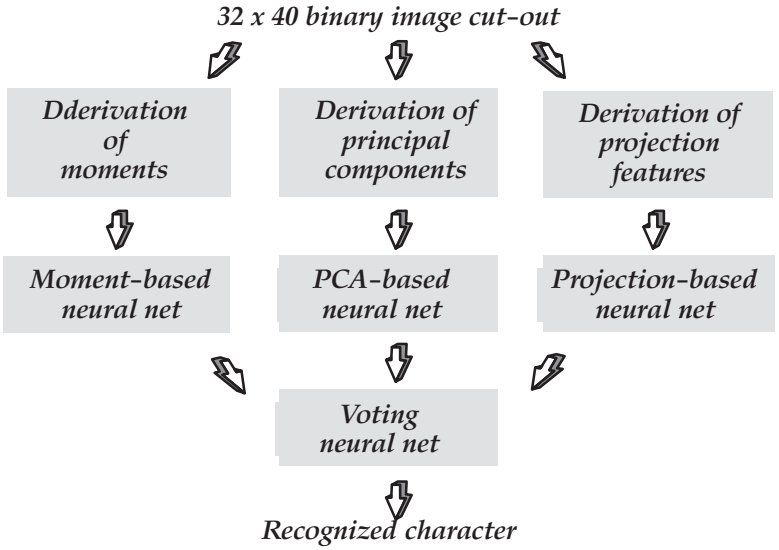

Figure 2: *A typical character recognizer (from [28]).*

The committee arrangement generalizes this idea. It can have significantly better predictions on new data at an acceptable increase of the computational complexity. The performance of the committee can be much better than the performance of each single network in isolation. The committee contains a set of a trained networks diversified in a distinct way. *Diversity* can appear in the number of hidden neurons, in the kind of network model, in the mixture of networks, in the optimization criteria, in the initial weight configuration, training parameters in the training samples, etc. The extent to which reliability can be improved by combining neural net solutions depends on the type of diversity, present in the set of nets.

All such techniques assume that the basic neural network is optimally trained. However, we have noticed that training algorithms are often slow and sometimes unable to converge, even though the underlying techniques often perform very well on other problems. In other words, even though an ANN can be trained to some functionality, there appears to be an underlying problem that causes unreliability in learning. This thesis will therefore be devoted to unravel such circumstances and to contribute ways in which reliable learning can be achieved. By large, the neural paradigm problem is represented as a stream of examples (data) and that guides the learning algorithm to adapt the network parameters until the network is "trained" to give the right answers to the posed questions. Thus the success and the reliability of this training depends to a large extent on the content and composition of this data stream.

Overall unreliable learning can be considered to result from the interaction between three factors: network, problem, and algorithm. In an attempt to answer questions like why and when the learning process will become unreliable and when a systematic failure can appear, backpropagation (still the algorithm with highest practical significance) has been used. The restricted class of architectures it is supposed to be used for and the feedforward architecture allow us to elaborate in more detail on the problem with respect to the chosen architecture and algorithm.

As we found that the conventional focus on network, problem and algorithm leaves much to be desired, we propose here to base the discussion rather on *symmetry*, *randomness* (as basic network design principles), and *knowledge* (the problem to be learned) as the basic ingredients of the universe of discourse. A high degree of symmetry in the initially designed network is historically viewed to favor the learning algorithm in providing an equal chance to move in several directions. However, this has also a drawback: the freedom of choice may lead to indecisiveness. Admittedly, randomness may in turn help the network escape from such a dilemma. But then again, randomness may wipe away the knowledge; hence a working balance should be found.

Symmetry can be dominant in the beginning of, but also at specific moments during, learning. Randomness (for instance as stochastic variable in the learning algorithm or as additional noise at the network input, output or internal parameters) is then required to force the presentation of examples to follow alternative itineraries. When the amount of randomness is not sufficient to counteract symmetry, learning will not be completed: instead of being adapted to ensure the right mapping between input/output data strings, the initial parameters will eventually become zero. If the noise (the randomness) of the system is dominant, learning will also be unsuccessful, because the network will rather learn the noise than the exemplified knowledge. The fundamental issue of learning is

therefore the creation of a functional balance between symmetry and randomness directed by the examples (the knowledge).

To bring this idea into tangible borders, the interaction between learning components is represented in the *error surface* paradigm. The network will be able to extract the necessary information by adapting itself to map the questions posed to the right answers. This adaptation is in fact an optimization procedure and is thus equivalent to finding the minimum energy state on an error landscape. The steps, that the learning algorithm takes on this landscape, are directed by the presented examples and form a *learning trajectory* on this surface. Directing this itinerary properly can help to escape some difficulties to pass surface areas, at which the learning algorithm normally spends a lot of time on or from which it can never escape. For finding an optimal trajectory on the error surface, the so–called regularisation methods have been used. An alternative effect has the introduction of extra noise during training. Our objection here is that the task complexity or the convergence accuracy may be changed in an unwanted direction. The investigation of the statistical long–run effects of example presentation when traveling on the difficult forms of the global error surface brings us to a constructive algorithm which helps in escaping them.

Therefore, the work in this thesis takes an alternative route to ensure reliable learning by focussing on *sample diversity* [116]. On basis of the instantaneous characteristics of the current training set we will conclude on learnability, reorder the set if necessary to establish the best sample sequence and train eventually a single network with success.

In conclusion, this thesis aims to give directions on how learning can be guaranteed so that its duration will be short and stable and its success unquestionable from the outset. In this respect, we aim to contribute to move neural technology from the realm of “Learning by Examples” to “Design by Examples”.

Groningen, march 1999.

Contents.

1 Introduction. 1

1.1 Neural networks. 2
1.1.1 The network structure. 3
1.1.2 The network operation. 4
1.1.3 Successful applications. 6

1.2 Deriving the neural model. 7
1.2.1 Formalization of neural learning. 7
1.2.2 Computational drawbacks of ANNs. 10
1.2.3 Network optimization. 11

1.3 Creating an intelligent system 12
1.3.1 Learning from reactivity 12
1.3.2 Data or samples? 13
1.3.3 This thesis 13

2 Learning Reliability. 15

2.1 Reliability in neural learning. 16
2.1.1 Notions and definitions. 17
2.1.1.1 Sensitivity. 18
2.1.1.2 Tolerance. 18
2.1.1.3 Redundancy. 19
2.1.2 Neural system reliability. 19
2.1.2.1 Performance aspects. 19
2.1.2.2 Reliability assessment. 20
2.1.3 Fault tolerance in neural networks. 21
2.1.3.1 Fault models. 21
2.1.3.2 Failure models 23
2.1.3.3 Suitability of reliability analysis. 24

2.2 Reliability as optimal trajectory. 24
2.2.1 The error landscape paradigm. 25
2.2.1.1 Reliable learning trajectory. 26
2.2.1.2 Influences on neural reliability 27
2.2.1.3 Correspondence with learning factors. 28
2.2.2 Different views on the error relief. 31
2.2.2.1 Learning process is a trajectory. 32

2.2.2.2 Effects of randomness 34
2.2.2.3 Distributed representation 35
2.3 Reliability enhancement 35
2.3.1 Functional redundancy for enhanced generalization. 35
2.3.1.1 Generalization diversity and committees. 36
2.3.1.2 Voting network. 37
2.3.2 Regularization methods for reliability enhancement. 38
2.3.3 Adaptable learning trajectory. 40
2.3.3.1 Criteria for reliability estimation 42
2.3.3.2 Towards an optimal learning trajectory 44

3 Symmetry and indecision. 45
3.1 Initial symmetry. 46
3.1.1 Structural symmetries. 46
3.1.1.1 Permutation transformation. 46
3.1.1.2 Sign transformation. 47
3.1.1.3 Repeatedness. 48
3.1.2 Symmetries in the network parameters. 48
3.1.2.1 Overparametrization. 49
3.1.2.2 Range symmetries. 50
3.1.3 Statistical mechanics view on symmetry breaking. 51
3.1.3.1 Spin−glass model. 51
3.1.3.2 Soft committee machine. 52
3.1.3.3 Shortcomings. 52
3.2 Flatness by subsequent learning 53
3.2.1 Looking at adaptation 53
3.2.1.1 Weight adaptation in time 54
3.2.1.2 The role of transfer 54
3.2.2 Incomplete adaptation 56
3.2.2.1 Saturation effects. 56
3.2.2.2 Numerical influences. 57
3.2.2.3 Badly balanced training set. 57
3.3 Learning scenarios, causing degradation. 58
3.3.1 Symmetries in the patterns 59
3.3.1.1 Spatial symmetries in patterns. 59
3.3.1.2 Temporal symmetries in patterns. 61
3.3.2 Symmetry in the error surface 61
3.3.2.1 Extrema 61
3.3.2.2 The saddle point. 62
3.3.3 Symmetrical signals on problematic regions 64

3.3.3.1 Training two identical networks. 67
3.3.3.2 Error landscape symmetries and degradations. . 70
3.4 Knowledge, Symmetry and Randomness. 72
3.4.1 How things came to bear. 72
3.4.1.1 Taking a different view. 73
3.4.1.2 Learning stages and reliability. 74
3.4.1.3 Towards a KRS measure. 77
3.4.2 The KRS model. 77
3.4.2.1 Looking into the mirror. 78
3.4.2.2 The role of the qualifier. 79

4 Example selection 81

4.1 The basics of sampling 83
4.1.1 Sampling techniques 83
4.1.1.1 Random sampling 84
4.1.1.2 Alternative sampling schemes 85
4.1.2 Neural active learning 86
4.1.2.1 Active selection (Informative learning) 86
4.1.2.2 Active sampling (Progressive learning) 87
4.1.2.3 Active learning implementation principles 89
4.2 Alternative example selection schemes. 90
4.2.1 Example presentation order and learning success. 90
4.2.1.1 Impact on the learning success. 91
4.2.1.2 Resampling schemes. 92
4.2.1.3 Bootstrap resampling 93
4.2.2 Example stream features 93
4.2.2.1 Definition of a cancelation training set 94
4.2.2.2 Symmetrical signals and cancelation 97
4.2.2.3 Cancelation criterion 99
4.2.3 Reliability of cancelation signals 103
4.2.3.1 Periodicity 104
4.2.3.2 Mean and variance of training duration 105
4.3 Sampling strategy. 107
4.3.1 Windowed sampling strategy 109
4.3.1.1 Formalization of sample selection techniques . . 110
4.3.1.2 Experiments with sample selection orderings . . 112
4.3.2 Summary and further suggestions. 115

5 Algorithms for windowed sample selection 119

5.1 Cancelation Signal Groups 120

5.1.1 Second−order problems 120
5.1.1.1 Some elementary observations. 120
5.1.1.2 Influence of sampling 122
5.1.2 Cancelation and Periodicity. 123
5.1.2.1 Least squares optimization for periodic signals. 123
5.1.2.2 Cancelation effects by the periodical signals . . . 124
5.2 Improved learnability. 127
5.2.1 Improved learnability for second−order problems. 127
5.2.1.1 An algorithm for active training 128
5.2.1.2 Algorithm 1. 130
5.2.2 Coping with periodicity and general cancelation. 131
5.2.2.1 Algorithm 2. 132
5.2.2.2 Experiments with interval size 134
5.2.2.3 Constructing input streams. 137
5.3 Two real−life problems 140
5.3.1 Diagnosis of turbo−generator 141
5.3.2 QRS detection. 143
5.3.2.1 Physiological facts about ECG 143
5.3.2.2 Artificial sample generation 144
5.3.2.3 Towards a solution 145
5.3.2.4 Approximation by windowed sample selection . 146
5.4 Discussion. 148

6 Closing remarks. 151
6.1 Justification. 151
6.2 Contributions of this thesis. 153
6.3 Suggestions for future research 156

References. 159

List of Tables. 171

List of Figures. 173

List of Symbols. 175

List of Abbreviations. 176

Appendices 177
Appendix A: KRS−experiments 178

A.1 Signal No 1 178
A.2 Signal No 2 179
A.3 Signal No 3 179
A.4 Signal No 4 180
A.5 Signal No 5 180
A.6 Signal No 6 181
A.7 Signal No 7 181
A.8 Signal No 8 182
A.9 Signal No 9 182
Appendix B: The Random walk 183
B.1 The random walk in one direction 183
B.2 Generalizing the random walk 184
B.3 Interpretation 184
Appendix C: Software 185
C.1 The Permutation procedure. 185

Index of Terms. 187

Samenvatting 190

Acknowledgements 194