A Study of Experimental Evaluations of Neural Network Learning Algorithms: Current Research Practice

Lutz Prechelt (prechelt@ira.uka.de)
Fakultät für Informatik
Universität Karlsruhe
76128 Karlsruhe, Germany
+49/721/608-4068, Fax: +49/721/694092

Technical Report 19/94

August 24, 1994

Abstract

113 articles about neural network learning algorithms published in 1993 and 1994 are examined for the amount of experimental evaluation they contain. Every third of them does employ not even a single realistic or real learning problem. Only 6% of all articles present results for more than one problem using real world data. Furthermore, one third of all articles does not present any quantitative comparison with a previously known algorithm. These results indicate that the quality of research in the area of neural network learning algorithms needs improvement. The publication standards should be raised and easily accessible collections of example problems be built.

Contents

1 Introduction

A large body of research in artificial neural networks is concerned with finding good learning algorithms to solve practical application problems. Such work tries to improve for instance the quality of found solutions (generalization), the probability of convergence, the ease of use, the learning speed, or some combination thereof. Currently, there exists no theory that quantitatively predicts the behavior of a new algorithm compared to other algorithms for any of these criteria. Consequently, experimental evaluation[1] is needed to validate any claims of improvement made for a new algorithm or to characterize under which circumstances improvements can be expected.

It seems that such evaluation is often not performed thoroughly enough, even in articles published by leading journals. Motivated by this impression, I decided to investigate this hypothesis by studying the current research practice empirically. In a recent study of experimental evaluation in computer science publications, the journal Neural Computation produced quite good results, far above average [1]. However, the only measure used in that work was the fraction of article space devoted to the evaluation and the articles considered were not only those about learning algorithms. The approach taken in the present study is more concrete at assessing the quality of an evaluation. I review the set of all articles presenting learning algorithms for practical problems that appeared in two renown neural network journals in 1993 and the first half of 1994. In each article, the number of problems used in the algorithm evaluation and the number of other algorithms used for comparison were counted. While high numbers resulting from such counting cannot prove that the evaluation has high quality, low numbers prove that the quality is low.

The articles under consideration are from the two oldest journals dedicated to neural network research, namely *Neural Networks* (NN), the official journal of the International Neural Network Society, published by Elsevier, and *Neural Computation* (NC), published by MIT Press. From Neural Networks, all articles of volume 6 (1993) and all articles from numbers 1 to 5 of volume 7 (1994) were used. From Neural Computation, all articles of volume 5 (1993) and all articles from numbers 1 to 4 of volume 6 (1994) were used.

The subsequent sections present the methodology and limitations of the study, the obtained results, and the conclusions drawn. The raw data collected for the study is listed in appendix A.

2 Methodology

2.1 Approach

The objective of the present study is to determine the quality of current algorithm evaluations. As a measure of quality we use the number of problems and compared algorithms used in an evaluation. The exact criteria are described in the next section. We consider the quality of the evaluation to be low when these numbers are low. If the numbers are high, no statement of quality can be made with this method.

The rationale of this approach can be outlined as follows. Since this study is meant to be a scientific one, its results must be reproducible. However, human judgement is subjective, making it difficult to obtain reproducible results. There are two ways to improve reproducibility of human judgement.

1. Let a large set of competent human subjects repeat the judgement and apply statistical techniques to compute an aggregate judgement.
2. Find judgement rules that avoid subjectivity.

Since the first approach is very resource-intensive, the second approach was taken in the present study. The criteria to be applied for counting can be formulated in a way that reduces subjectivity to a negligible level.

2.2 Method

These steps were taken to obtain the raw data for the study.

1. Each article from the before-mentioned range of publications was classified into one of the following categories.

[1] In this report, I will use the term *evaluation* to mean *experimental evaluation*.

Theory. Articles belong to the "Theory" category if and only if the major contributions made by the paper are formally proven propositions.

Modeling. Articles predominantly concerned with the formal modeling of some aspects of natural neural networks, or with discussing the properties of such models, or with other aspects of biological plausibility belong to the "Modeling" category.

Algorithm. Articles whose main contribution is the design of a new learning algorithm to be applied to practical problems form the "Algorithm" category[2]. Empirical studies comparing several known algorithms and application papers presenting architectures for applying known algorithms to a particular problem field are also included here, since they are quite rare (only 5% of the category).

Other. All articles that do not fit into any of the above categories are put into the "Other" category. This includes surveys and papers on electronic neural network hardware.

"If in doubt, leave it out." In all borderline cases, papers were *not* classified as Algorithm in order to avoid a negative bias in the data due to papers that were not meant to make an algorithm contribution and, thus, lack proper evaluation. In particular, any paper appearing in Neural Computation that was marked to be a "Note" and that would have been an Algorithm paper by its topic was classified as Other in order to avoid a negative bias in the data due to papers that were simply too short to contain proper evaluation.

2. After the category of each article was determined, only the articles from the Algorithm category were used in the study. Each Algorithm article was reviewed to determine the two key metrics used in the study, namely

- the number of different learning problems (data sets) used in the evaluation and
- the number of known algorithms a proposed algorithm is compared to.

For a more meaningful discussion, each learning problem is classified to be either an artificial, a realistic, or a real problem.

[2]The word Algorithm, with capital A, will be used throughout this report to refer to the category.

Artificial problems are those whose data is generated synthetically based on some simple logic or arithmetic formula.

Realistic problems also consist of synthetic data, but are generated by a model with properties similar to what can be found in real problems. Only the following three types of data generation procedures yield what is considered realistic problems: firstly, data generation using a complex and realistic mathematical model of a physical system such as a cart/pole system or robot kinematics; secondly, data generation by chaotic mathematical processes, such as the Mackey-Glass equation; and thirdly, data generation by stochastic processes, such as mixtures of Gaussian random variables. Realistic problems are useful to assess the behavior of an algorithm on problems with known properties; they provide the best way to *characterize* the kinds of problems for which an algorithm will yield good results.

Real problems consist of data that represents actual observations of phenomena in the physical world. Such data tends to contain some amount of errors and noise. Most importantly and in contrast to realistic artificial data, real data usually has characteristics that are not completely known (surprising features). We want learning algorithms to cope well with problems whose characteristics are partially unknown; how well they do can best be tested with real data.

Synthetic variations of the same problem count as a separate problem only if it is plausible to expect that two algorithms may compare very different on the variation than on the original problem. In many cases, two variations of a problem were found: one with and one without noise in the data. A very different problem representation is another kind of problem variation that counts as a separate problem. What exactly "very different" means cannot be quantified, but I did my best to apply constant criteria throughout the study.

To *use* a problem in an evaluation means to report any kind of quantitative data about the behavior of the proposed algorithm on this problem, for instance learning speed, convergence probability, training set error, or test set error.

The algorithms used for comparison were originally discriminated to be either neural network algorithm or other algorithms. Since that discriminiation is fuzzy, however, the separation is dropped in the discussion of the results. The count includes all algorithms not introduced in the article in question; algorithms that are newly proposed in an article are not counted. Articles presenting comparative empirical studies of known algorithms had all algorithms counted. When an article introduces several new algorithms at once, all algorithms used for a comparison to *any* of the new ones are counted, i.e., an algorithm used for comparison is counted even if it is not compared to all of the new algorithms.

2.3 Limitations

The method described above does not allow for a quantitative judgement of the overall quality of an evaluation. Even if many problems and compared algorithms are used, the relevance of the results may still be low due to irrelevant performance measures, irrelevant or biased problems, improper description of the setup, or other methodological errors. The assumption used in the approach is *not* that a large number of problems and compared algorithms in an article implies high evaluation quality, but instead that a small number implies low evaluation quality. The counting criteria themselves are biased towards finding large numbers.

An absolute quality measure is not required, since all this study is meant to do is investigating the hypothesis that algorithm evaluations are often of low quality. No attempt will be made to quantify what low quality means, because any such quantification would necessarily be arbitrary. Instead, we will reject the hypothesis unless we find subjectively overwhelming evidence for it. Hence, the approach of the study is quite conservative.

Nevertheless, a few remarks must be made on possible objections against the approach.

1. *An algorithm proposed for a narrow application domain does not allow for a wide variety of test problems.* This is true, but is not the issue debated here. Even for a very specialized algorithm, a number of different incarnations of problems from its domain can be found and should be investigated. For instance, variations of a problem obtained by significantly changing a major parameter such as the resolution of the data would be counted as separate problems. Only the number of problems is judged, not their variety.

2. *Often no algorithms can be found to be compared to an algorithm proposed for a narrow application domain.* Maybe no other *specialized* algorithms can be found. But it is nevertheless interesting to see how much improvement the new algorithm represents compared to known general purpose algorithms. Thus, such algorithms should be used for comparison.

3. *Algorithms solving a problem for which no solution was previously known cannot be compared to others.* This is true, but it hardly ever applies; I did not observe any instance of such an algorithm in the whole sample investigated in this study, although arguably there are a few borderline cases.

4. *Totally new approaches to a problem do not allow for comparison.* Why not? If the approach was made for its assumed utility, a comparison is the best means to assess it. Otherwise the article should not claim utility and would then be classified as Modeling in this study.

5. *Often a thorough evaluation is simply too much work.* The result of scientific work should be knowledge. An algorithm about whose behavior too little knowledge is available is no proper scientific contribution. Experimental evaluation may be a lot of work, but it needs to be done.

6. *I believe that your data contains many errors.* Probably there is a considerable number of errors in my data. See [1] for a discussion and estimation of the precision to be expected of the observations made; that discussion should also apply to the present study. Most importantly, however, as we will see below, the conclusions from this study do not change even if a large margin of error is assumed.

3 Results and Discussion

The raw data obtained during the study is presented in appendix A. In this section, I will present only the most prominent findings. Since the differences between Neural Networks and Neural Computation are quite small in most respects covered here, I will discuss the set of all Algorithm articles studied as a whole.

Let us first have a look at the total number of problems used in the evaluation. This is depicted in figure 1.

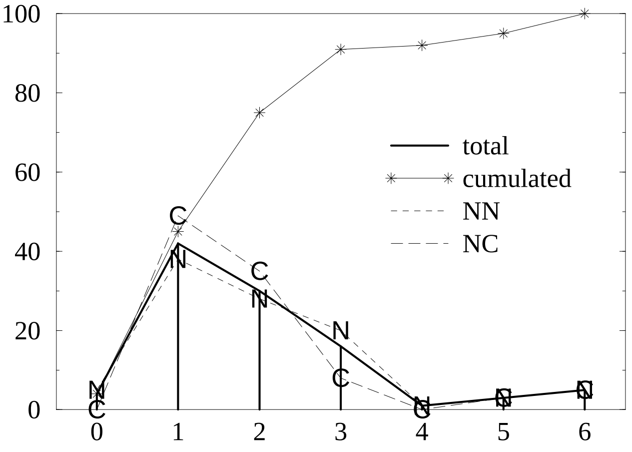

Figure 1: Percentage y of Algorithm articles that use a total of x different problems for the evaluation.

The figure is to be read as follows. On the abscissa (x-axis), we find the article classes from "0 problems used" up to "5 problems used". The last point, $x = 6$, stands for "6 or more problems used". The ordinate value (y-value) indicates the percentage of articles belonging to the class. The curve drawn as a thick line indicates the value for the total of all Algorithm articles found, while the dashed lines show the corresponding data for Neural Networks (NN) and Neural Computation (NC), respectively, alone. The starred line is the accumulation of the values on the thick line from left to right; it can be used to read quantiles. All other figures have the same structure.

As we see, 4% of all articles do not have any experimental evaluation and only 25% use more than two problems for the evaluation. While it is surprising enough that any Algorithm article without experimental evaluation can be published in a renown journal, it is even more staggering how few articles use a broad set of problems. Only 9% of all articles use more than three problems.

Now let us differentiate this data by problems being either artificial, realistic, or real as defined in section 2.2. Figure 2 shows the number of artificial problems used. No special remark is to be made here, since artificial problems should only serve for the illustration (as opposed to the evaluation) of an algorithm; a large number of artificial problems in an article is neither good nor bad. 20 articles (18%) employed the "grandfather" of all neural network problems, the XOR or n-bit parity[3].

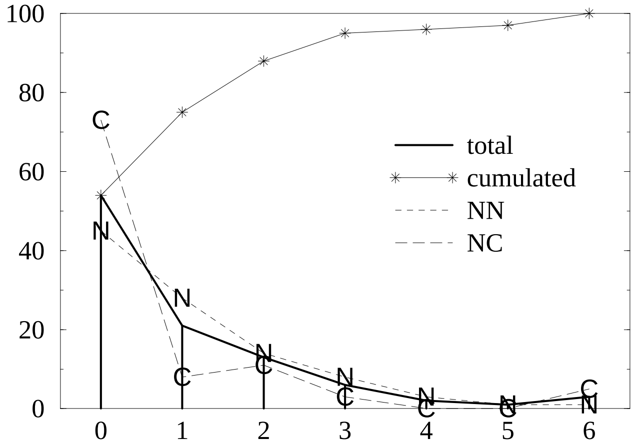

Figure 2: Percentage y of Algorithm articles that use x different artificial problems for the evaluation.

Figure 3 shows the number of realistic problems used per article. As mentioned before, such problems are useful to explore an algorithm on data whose properties are realistic, yet exactly known. Despite that usefulness, 59% of all articles do not use any realistic problem, only 4% use more than two, and 3% more than three. As we see, an experimental exploration of the question "For which kinds of problems is this algorithm best suited?" is hardly ever done.

Figure 4 shows the number of real problems used per article. *The Zen master says: The only way to test how an algorithm performs on a real problem is to test how it performs on a real problem.* Of course, nobody can say how results on one real problem (or, for that matter, 15 real problems)

[3] This number cannot be found in the raw data listing in appendix A

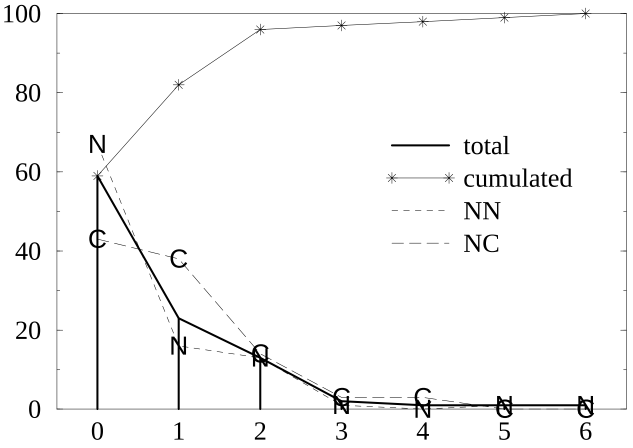

Figure 3: Percentage y of Algorithm articles that use x different realistic problems for the evaluation.

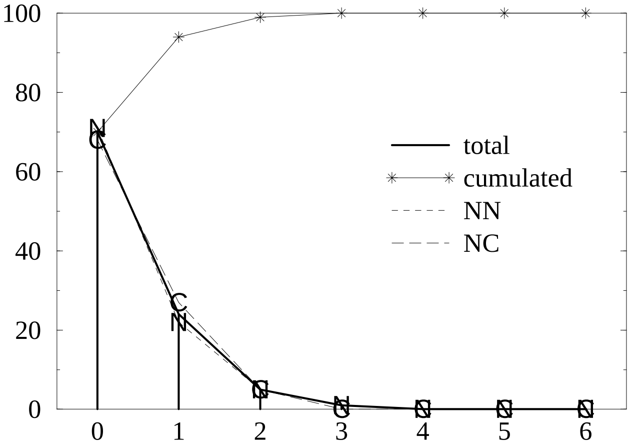

Figure 4: Percentage y of Algorithm articles that use x different real problems for the evaluation.

generalize to other problems, but it is also impossible to say exactly how the performance on realistic problems will generalize to real problems. Thus, it should at least be verified that an algorithm performs well for *some* real problems, as real problems are the only tests of a learning algorithm that are *guaranteed* to have at least some practical relevance (namely for the exact problem tested). Another reason is that real data tends to have some totally unexpected features that artificially generated data, even if otherwise realistic, lacks. However, the use of real problems in the articles of the study is deplorably rare. 70% of all articles do not use any real problem, only 1% use more than two, and not a single one was found using more than three.

Even when summing the number of realistic and real problems used in each article, as depicted in figure 5, a huge fraction of all articles is devoid of a reasonable number of test problems. 34% of all

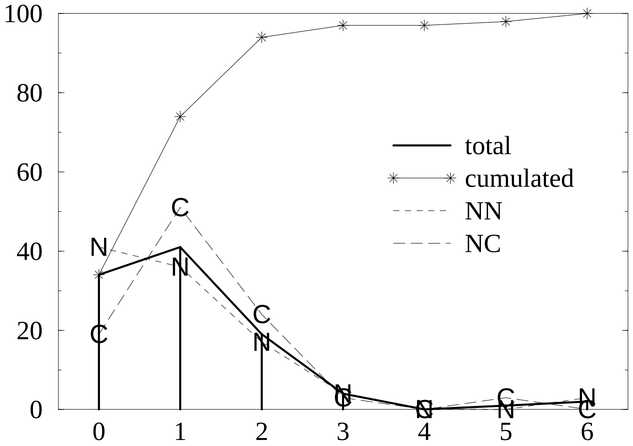

Figure 5: Percentage y of Algorithm articles that use x different realistic or real problems for the evaluation.

articles use zero realistic *and* zero real problems, 6% use of more than two and a mere 3% use more than three.

The situation does not look much better when one considers the number of other algorithms used for comparison, as shown in figure 6. As much as

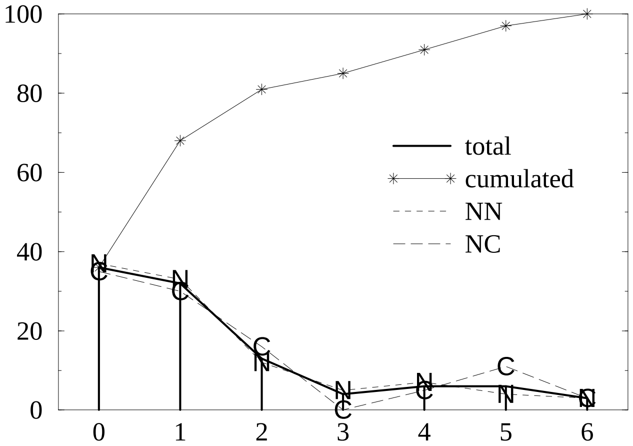

Figure 6: Percentage y of Algorithm articles that use x different known algorithms for comparison.

34% of all articles feature no comparison with other algorithms at all; only 19% compare to more than two known algorithms. This would not be a problem if everybody used standardized problems in standardized setups, but for the realistic and real problems this is not the case — it is very rare today that two different articles publish directly comparable results for the same problem. Without such comparability, however, the above number means that for one out of every three articles the evaluation performed would better be called a naval inspection.

4 Conclusion

Let us finally make a short mental experiment. Assume that we set the following very modest standard. *An algorithm evaluation is called acceptable if it uses a minimum of two real or realistic problems and compares the results to those of at least one alternative algorithm.*
Now assume that somebody had asked you before you read this report "What fraction of Algorithm articles published in NN and NC do you guess does *not* meet this standard?".
What had your answer been? The correct answer for the sample of articles investigated here is 82%. Sad, but true.

This result indicates that today new neural network learning algorithms are often published in a form that does not represent useful and validated knowledge. These articles present an idea of the kind "This is a way to tackle certain learning problems.", but they do not tell us what we have to expect if we really try that idea. Instead, each article presenting a new algorithm should give at least a preliminary answer to the questions "For what kinds of problems does the new algorithm work well or not well?" and "Under what conditions should we prefer the new algorithm over previously known ones?". This information is essential if the publication of the algorithm is meant to be a scientific progress.

I believe the following steps should be taken to improve on the current situation.

1. Editors and reviewers must set significantly higher standards for the experimental evaluation of a new learning algorithm. Articles that do not meet these standards should usually be rejected.
2. Researchers must reserve enough resources for thorough experimental evaluation of their algorithms.
3. The research community must prepare and use public collections of example problems from all relevant fields in order to simplify algorithm evaluations. Re-use of example problems is also a prerequisite for broad comparisons of algorithms. Only a few fields such as speech recognition and optical character recognition do already have such collections.
4. Standard experimental setups and standard result presentation formats should be developed to improve comparability and reproducibility of evaluation results.

Without these improvements, progress in the learning algorithm field will be significantly slower than it could be.

A Collected data

This section contains the raw data collected during the study. For each article that was considered an Algorithm article, the following lists contain the page number (**p**) of its first page, the number of artificial (**A**), realistic (**r**), and real (**R**) data sets used, the number of neural network methods (**N**) compared to the proposed ones, and the number of other, i.e., non-neural, methods (**o**) compared to the proposed ones. For the Theory, Modeling, and Other categories, only the page numbers of the respective articles are given.

The study comprises 271 articles, 150 from Neural Networks and 121 from Neural Computation. Of these, 113 (41%), were Algorithm articles, 76 (67%) of them from Neural Networks and 37 (33%) of them from Neural Computation; only 6 (5%) of all Algorithm articles were empirical studies.

Below is the raw data for each of the journal volumes.

A.1 Neural Computation 1993

Total number of articles: 72

Algorithms, 28 articles (39%):

p	A	r	R	N	o	p	A	r	R	N	o
45	0	2	0	0	0	61	0	1	0	0	0

75	0	2	0	0	1	89	0	1	1	0	1
105	0	1	0	4	1	115	3	0	0	0	0
242	2	0	0	1	0	267	0	1	0	0	0
289	0	0	1	0	0	317	2	1	0	0	1
331	0	1	0	0	4	402	0	0	1	0	0
419	0	0	1	0	0	430	0	0	2	0	0
443	0	0	2	1	0	456	0	1	0	1	0
463	1	0	0	0	0	473	0	0	1	1	0
483	0	1	0	2	0	492	0	0	1	4	0
625	0	2	0	1	0	719	1	0	0	0	0
750	6	0	0	2	0	795	0	3	0	0	0
885	0	0	1	0	0	928	0	1	1	1	0
939	0	2	0	6	0	954	1	1	0	2	0

Modelling, 13 articles (18%):
1, 32, 213, 228, 341, 374, 597, 636, 695, 736, 823, 843, 869,

Theory, 17 articles (24%):
18, 21, 132, 140, 165, 205, 278, 305, 359, 371, 392, 550, 767, 783, 812, 893, 910,

Other, 14 articles (19%):
154, 200, 210, 260, 363, 367, 505, 570, 587, 613, 649, 665, 681, 856,

A.2 Neural Computation 1994, 1–4

Total number of articles: 49

Algorithms, 9 articles (18%):

p	A	r	R	N	o	p	A	r	R	N	o
161	0	1	0	0	2	181	0	1	0	1	4
270	0	0	1	0	0	441	2	0	0	1	0
469	12	1	0	3	2	509	0	4	1	2	0
521	0	2	0	4	1	718	0	1	0	1	1
748	2	0	0	1	0						

Modelling, 24 articles (49%):
1, 14, 19, 29, 38, 56, 69, 85, 127, 225, 242, 296, 307, 357, 390, 405, 559, 602, 615, 622, 642, 668, 679, 696,

Theory, 15 articles (31%):
100, 147, 220, 256, 285, 319, 334, 341, 420, 459, 491, 543, 658, 739, 761,

Other, 1 article (2%):
215,

A.3 Neural Networks 1993

Total number of articles: 86

Algorithms, 43 articles (50%):

p	A	r	R	N	o	p	A	r	R	N	o
33	0	0	1	0	0	69	0	0	1	1	0
99	1	2	0	0	0	115	1	0	0	1	0
127	1	0	0	1	0	133	3	0	0	0	0
217	0	0	0	0	0	223	3	0	0	0	0
275	0	0	1	2	2	285	0	0	1	1	0
351	2	1	0	1	0	365	1	1	0	1	0
385	3	0	0	1	2	397	0	2	0	1	0
409	1	0	0	0	0	423	0	0	0	0	0
441	2	0	0	0	0	485	1	0	0	1	0
517	1	0	0	0	1	525	2	0	0	2	1
535	0	0	3	3	1	559	0	0	1	0	0
583	2	0	0	0	0	613	1	0	1	1	0
719	2	0	1	1	0	729	0	0	1	1	0
771	0	6	0	12	0	785	1	0	0	0	1
791	0	0	1	0	0	807	0	1	0	1	0
835	6	0	0	0	0	869	0	0	1	1	1
907	1	0	0	0	0	919	3	0	0	1	1
933	0	2	0	0	1	963	1	0	0	0	1
997	1	0	0	2	0	1011	1	0	0	0	0
1023	0	0	2	4	0	1105	4	0	2	2	3
1117	1	1	0	1	5	1159	0	0	0	0	0
1169	2	0	0	1	0						

Modelling, 10 articles (12%):
7, 43, 245, 313, 463, 499, 753, 895, 947, 1045,

Theory, 31 articles (36%):
79, 161, 167, 177, 189, 393, 429, 547, 573, 627, 649, 655, 667, 677, 689, 705, 709, 801, 823, 845, 855, 861, 975, 991, 1019, 1061, 1069, 1073, 1089, 1095, 1131,

Other, 2 articles (2%):
327, 1147,

A.4 Neural Networks 1994, 1–5

Total number of articles: 64

Algorithms, 33 articles (52%):

p	A	r	R	N	o	p	A	r	R	N	o
1	3	0	0	3	2	41	0	0	1	1	0
89	2	2	0	1	0	113	1	1	0	0	2
129	2	1	0	1	0	159	0	0	0	0	0
169	1	0	0	0	0	175	1	1	0	1	0
183	0	2	0	1	0	195	1	0	0	0	0
279	2	0	0	0	0	291	0	5	1	0	0
301	3	0	0	0	0	341	5	0	0	2	0
353	4	0	1	1	1	375	0	0	1	1	1
389	0	0	2	0	0	397	1	0	0	0	0
539	0	2	1	3	0	553	0	2	0	0	0
565	0	2	0	0	3	643	0	1	0	0	0
661	2	1	0	1	0	671	1	0	0	0	1
681	0	2	0	1	1	691	0	0	1	2	0
709	0	1	0	3	1	767	0	3	0	0	0
777	0	0	2	5	0	797	2	0	1	4	0
809	1	1	0	1	0	819	0	1	0	0	0
833	0	2	0	0	0						

Modelling, 8 articles (13%):
13, 49, 65, 79, 419, 441, 463, 589,

Theory, 23 articles (36%):
101, 141, 147, 219, 233, 251, 261, 271, 313, 321, 331, 449, 455, 477, 491, 495, 507, 523, 609, 629, 737, 753, 783,

Other, 0 articles (0%):

References

[1] Paul Lukowicz, Ernst A. Heinz, Lutz Prechelt, and Walter F. Tichy. Experimental evaluation in computer science: A quantitative study. Technical Report 17/94, Fakultät für Informatik, Universität Karlsruhe, 76137 Karlsruhe, Germany, August 1994. anonymous ftp: /pub/papers/techreports/1994/1994-17.ps.Z on ftp.ira.uka.de.