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

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

(Received 30 September 1994; accepted 14 September 1995)

Appeared in "Neural Networks" Vol. 9, 1996

Abstract

190 articles about neural network learning algorithms published in 1993 and 1994 are examined for the amount of experimental evaluation they contain. 29% of them employ not even a single realistic or real learning problem. Only 8% of the articles present results for more than one problem using real world data. Furthermore, one third of all articles do not present any quantitative comparison with a previously known algorithm. These results suggest that we should strive for better assessment practices in neural network learning algorithm research. For the long-term benefit of the field, the publication standards should be raised in this respect and easily accessible collections of benchmark problems should be built.

Keywords: algorithm evaluation, science, experiment

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 solutions found (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.

I often felt that such evaluation is frequently 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 had quite good results, far above average (Tichy, Lukowicz, Prechelt and Heinz, 1995). 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 a large set of articles presenting learning algorithms for practical problems that appeared in four renowned neural network journals in 1993 and 1994. In each article, the number of problems used in the algorithm evaluation and the number of previously known algorithms used for comparison were counted. Although high numbers resulting from such counting cannot prove that the evaluation has high quality, low numbers strongly suggest that the quality is insufficient.

The articles under consideration are from four of the oldest and most well-known journals dedicated

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

to neural network research, namely

1. *Neural Networks* (NN), published by Pergamon Press; all articles of Volume 6 (1993) and all articles from numbers 1 to 5 of Volume 7 (1994).
2. *Neural Computation* (NC), published by The MIT Press; all articles of Volume 5 (1993) and all articles from numbers 1 to 4 of Volume 6 (1994).
3. *Neurocomputing* (NE), published by Elsevier Science; all articles of Volume 5 (1993) and Volume 6 (1994).
4. *IEEE Transactions on Neural Networks* (TN), published by the IEEE; all articles of Volume 5 (1994).

Altogether, 414 articles are in the sample.

The subsequent sections present the methodology and limitations of the study, the results obtained, and the conclusions drawn.

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 if these numbers are low. If the numbers are high, no statement of quality can be made with this method.

The rationale of this approach is to have criteria that involve only a minimal amount of subjectivity, so that the results of the study are reliable and repeatable. The criteria to be applied for counting can be formulated in a way that reduces subjectivity to a negligible level.

2.2 Method

Two 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.

	A	E	M	T	H	O	Tot.
NC	34 28%	3 2%	37 31%	32 26%		15 12%	121
NE	23 56%	4 10%	2 5%	4 10%	5 12%	3 7%	41
NN	71 47%	5 3%	18 12%	54 36%	2 1%		150
TN	47 46%	3 3%		32 31%	9 9%	11 11%	102
Tot.	175 42%	15 4%	57 14%	122 29%	16 4%	29 7%	414

Table 1: Distribution of articles over classes Algorithm (A), Empirical (E), Modeling (M), Theory (T), Hardware (H), and Other (O) for the four journals. Empty fields are zero entries.

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 (less than 8% of the category).

Hardware. Articles whose main contributions are concerned with the design of circuits for electronic implementations of neural networks.

Other. All articles that do not fit into any of the above categories are put into the "Other" category. In particular, this includes surveys and reviews.

"If in doubt, leave it out": In 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, the short "Note" papers in Neural Computation and "Letter" papers in IEEE TNN, that would have been Algorithm papers by their topic were 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]The word Algorithm, with capital A, will be used throughout this report to refer to the category.

In Figure 1 you can see how many articles from which journals were classified in each of the categories described above. In the table, empirical studies are shown separately, although for the rest of the analysis they are treated as a part of the Algorithm class.

2. After the category of each article was determined, only the articles from the Algorithm category (A or E in the table) 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.

Artificial problems are those whose data is generated synthetically based on some simple logic or arithmetic formula, for example encoder/decoder, parity, sine wave etc.

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, or non-trivial differential equations; 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 and counted: 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 distinguished to be either neural network algorithms or other algorithms. Since this discrimination 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 with *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 only that a small number im-

plies 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 investigate the hypothesis that algorithm evaluations are often of low quality. We will reject the hypothesis unless we find subjectively overwhelming evidence for it — based on counting alone. 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 not a 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 are a number of errors in my data. No double-checking was performed to eliminate such errors, but the classification was done carefully to keep the error density low. Most importantly, the conclusions from this study do not change even if a rather large margin of error is assumed.

3 Results and Discussion

In the following, 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.

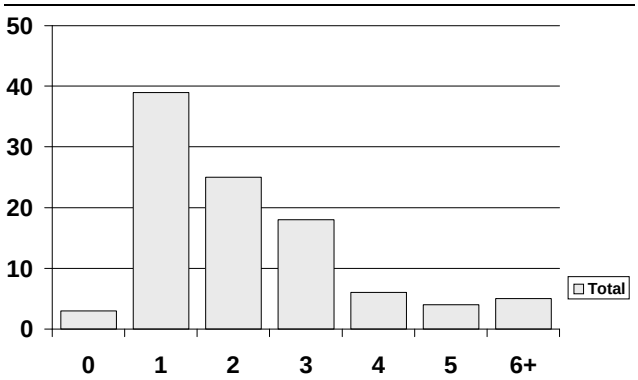

Figure 1: Percentage y of Algorithm articles that use a total of x different problems for the evaluation.

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. All other figures have the same structure.

As we see, 3% of all articles do not have any experimental evaluation and only 33% 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 renowned journal, it is even more surprising how few articles use a broad set of problems. Only 15% 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

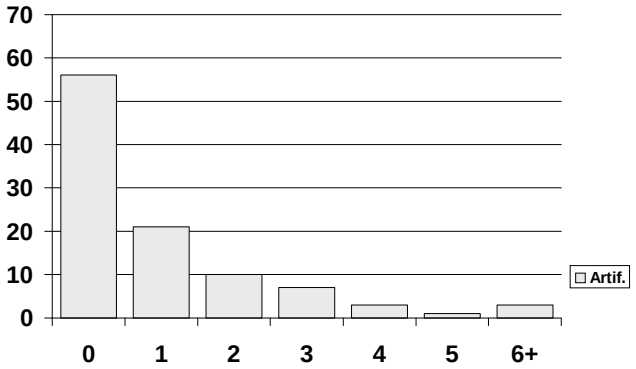

Figure 2: Percentage y of Algorithm articles that use x different artificial problems for the evaluation.

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.

Figure 3 shows the number of realistic problems used per article. As mentioned before, such prob-

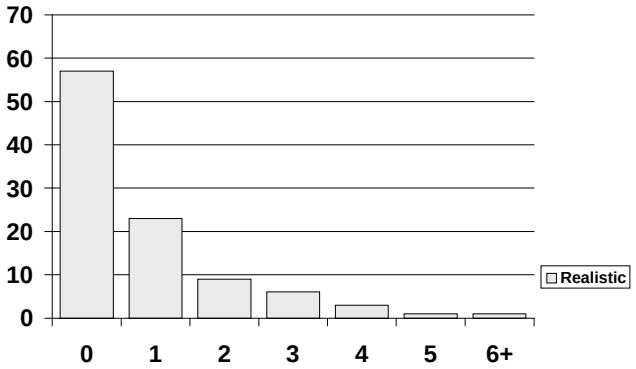

Figure 3: Percentage y of Algorithm articles that use x different realistic problems for the evaluation.

lems are useful to explore an algorithm on data whose properties are realistic, yet exactly known. Despite that usefulness, 57% of all articles do not use any realistic problem, only 11% use more than two, and 5% 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. Of course, nobody can say how results on one real problem (or, for that matter, 15 real problems) generalize to other problems, but

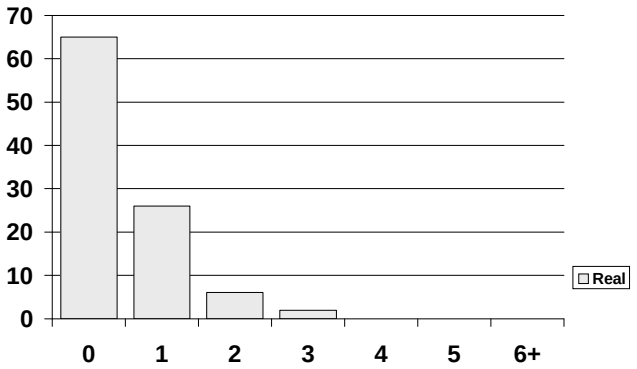

Figure 4: Percentage y of Algorithm articles that use x different real problems for the evaluation.

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 rare. 65% of all articles do not use any real problem, only 2% 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. 29%

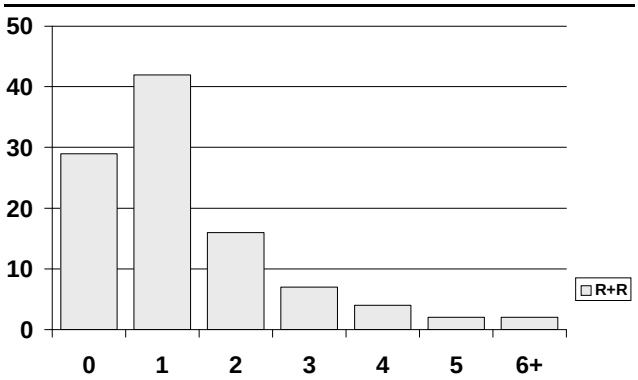

Figure 5: Percentage y of Algorithm articles that use x different realistic or real problems for the evaluation.

of all articles use zero realistic *and* zero real problems, that is, they are devoid of any meaningful empirical evaluation whatsoever! 14% use more than two problems and a mere 7% 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

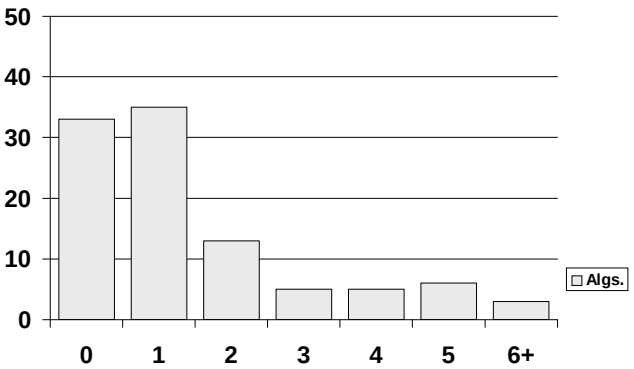

Figure 6: Percentage y of Algorithm articles that use x different known algorithms for comparison.

as 33% 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 quite 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 the top NN journals do you guess does *not* meet this standard?".
What had your answer been? The correct answer for the sample of articles investigated here is 78%. 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 should 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 should reserve enough resources for thorough experimental evaluation of their algorithms.
3. The research community should prepare and use public collections of example problems from all relevant areas in order to simplify algorithm evaluations. Re-use of example problems is also a prerequisite for broad comparisons of algorithms. Some related fields such as speech recognition, optical character recognition, image restoration, statistics, and machine learning do already have such collections and some efforts specifically for NN research are underway.
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.

References

Tichy, W.F., Lukowicz, P., Prechelt, L., and Heinz, E.A. (1995). Experimental evaluation in computer science: A quantitative study. *Journal of Systems and Software* 18(1), 9–18. Also as Technical Report 17/94 (August 1994),

Fakultät für Informatik, Universität Karlsruhe, Germany. Anonymous FTP: /pub/papers/tech-reports/1994/1994-17.ps.gz on ftp.ira.uka.de.