submitted to Helnet-97

Robust Model Assessment for Neural Networks

Stefan M. Rüger
Department of Computing, Imperial College
180 Queen's Gate, London SW7 2BZ, England
<s.rueger@doc.ic.ac.uk>

Arnfried Ossen
Informatik, Technische Universität Berlin
Sekr. FR 5-9, Franklinstr. 28/29, D-10 587 Berlin, Germany
<ao@cs.tu-berlin.de>

July 24, 1997

Abstract

We present a robust model assessment method for neural-network models and demonstrate the approach for feedforward networks. In order to assess a neural-network model, the location of weight vectors after repeated learning experiments is analysed in the effective weight space. We argue that appropriate or underdetermined models clearly show a cluster structure in the effective weight space whereas too complex models do not exhibit a proper cluster structure. We achieve robust model assessment by restricting estimates of network performance to clusters.

1 Introduction

Methods to select the structure and complexity of neural networks come in many flavours [3, 1], but most of them rely on estimating the network performance using test set patterns not employed during training. Different candidate networks are trained and the respective performances on the test set are compared. The network with the lowest estimate is then chosen.

However, these estimates can be unreliable because the test set may contain high-leverage outliers or may be unrepresentative for the true population. More importantly, standard learning algorithms may fail to converge to the optimal weight vector. This means that the test set error estimates the performance (or generalisation error) of a particular learned weight vector, rather than the performance of a network model.

In order to assess the *model* performance it seems necessary to run repeated learning trials, e. g. with different initial weight vectors. Indeed, learning runs where only the initial weight vector is varied can end up in different local minima,

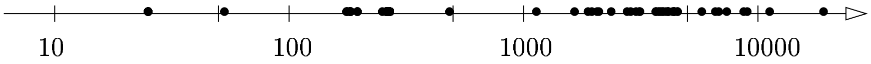


Figure 1: 100 test errors of different learning runs for a 4-9-2 network, where only the initial weight vector was randomly varied

resulting in a considerable test error variability (see Figure 1), and consequently in unreliable model assessment. Straightforward remedies, such as using the average test set error will still be inaccurate, as we will demonstrate in Section 2. We find that the introduction of an effective weight space, together with a canonical metric, allows us to remove the influence of other local minima and outliers, i. e. to estimate robustly.

The rest of this paper is organised as follows: In Section 3, we introduce the technique of weight vector clustering, which is applied to the experiments described in Section 2. The tree-like graphical representation of the resulting clusters is studied in Section 4, where we present robust performance statistics for feedforward networks. We then demonstrate that these robust statistics can be used for model assessment. Finally, we discuss a number of extensions to our work.

Throughout this article, the nodes of a network are addressed by symbols, 0 being the symbol for the bias node. All nonbias nodes are divided into one input layer, $k \geq 1$ hidden layers, and one output layer. Every layer is fully connected to the next layer, and there is a weight w_{ab} for every pair (a, b) of nodes, a and b being in successive layers. Hidden nodes and output nodes have a bias weight termed w_{oa}.

Hidden-layer nodes of feedforward networks use tanh as their activation function, nodes in the output layer, the identity. All weights of the network form a weight vector $w \in \mathbb{R}^E$ that parameterises a network function out_w. The term *i-h-o* addresses the network model with one hidden layer by describing the number of nodes of each layer. Approximation tasks may already be accomplished using networks with one hidden layer: the set of their network functions is dense in the set of all continuous functions with compact domain [5].

2 Network Performance

To demonstrate the characteristics of the test error, we chose a network that is not only small enough for concise presentation but also relevant for applications, e. g., from control engineering or time-series modeling. We generated 2400 data pairs (x^i, y^i) using a 4-9-2 feedforward network with a fixed weight vector $w^\star$. The inputs cover the $[-1, 1]^4$ cube, and the targets y^i may or may not contain small random perturbations ε^i:

$$y^i = \mathrm{out}_{w^\star}(x^i) + \varepsilon^i$$

The data pairs were randomly split into a training and a test set of 1200 pairs each.

Fitting $m = 100$ 4-9-2 networks with different random initial weights resulted in the test errors shown in Figure 1. The test error cannot reach zero because of the small amount of added noise. If the large variance were due to outliers, more robust estimators like the median could be used. The minimum should be an even better estimator, because outliers of the test error are expected to be too large.

Table 1 shows the above test error statistics for different network models, where the number of hidden nodes varies. For each network the same data were fitted 100 times using random initial weight vectors.

Model selection based on the lowest value of the test error statistic will result in too complex networks for the mean (4-12-2), median (4-11-2), and minimum (4-13-2). These test error statistics are not truly robust estimators of the generalisation error.

Table 1: Some statistics of the test error

Network	Mean ± SDV	Median	Minimum
4-4-2	20100 ± 38%	20400	12020
4-5-2	14200 ± 37%	11600	8376
4-6-2	10900 ± 51%	8730	4990
4-7-2	7880 ± 53%	7180	2058
4-8-2	5000 ± 110%	3860	269.1
⋆ 4-9-2	1870 ± 160%	268	25.06
4-10-2	1230 ± 290%	173	25.05
4-11-2	643 ± 310%	25.2	25.02
4-12-2	346 ± 590%	25.3	25.06
4-13-2	607 ± 530%	25.6	24.97

A more sophisticated use of the data, e. g., leave-k-out cross-validation or bootstrap, cannot improve the situation. Instead, we propose a robust statistic based on a geometric approach that enables us to restrict the performance estimation to a single local minimum, i. e. to remove the influence of other local minima and outliers.

3 Clustering in Weight Space

The basic idea of clustering in weight space is that an experiment, e. g., the approximation of a data set, is repeated with small changes like different initial weight vectors, different artificial noise in the data set (jitter), etc. Then, the resulting weight vectors are grouped by a suitable cluster algorithm in order to assess the typical results of the experiment.

Initially, clustering in weight space is hampered by symmetries $s\colon \mathbb{R}^E \to \mathbb{R}^E$ which leave the network function invariant: $\mathrm{out}_w = \mathrm{out}_{s(w)}$, e. g., by relabeling nodes within a hidden layer. Consider the subvector of weights that start or end at a certain hidden node. Changing their signs is another invariance, owing to the symmetry of the tanh activation function. Both transformations are bijective and linear; they form a certain symmetry group S that acts on the weight space. The problem is that networks with the same network function may have numerous different weight vectors. This may be overcome by applying a sign-change operation to all subvectors of hidden nodes whose bias weight is negative, and by relabeling the hidden nodes such that the bias weights are nondescending in every hidden layer. Thus every weight vector w (except for a set with zero Lebesgue measure) has a unique representative $r(w)$ in an effective weight space $\overline{W} := r(\mathbb{R}^E) \subset \mathbb{R}^E$ that is reduced in size by a factor of $h! \cdot 2^h$ for an i-h-o network.

The next problem is that r is discontinuous: consider two weight vectors $v \notin \overline{W}$ and $w \in \overline{W}$ that are very close to each other; their representative vectors $r(v)$ and $w = r(w)$ may nevertheless be separated. Thus $\overline{W}$ should be endowed with a metric that respects the symmetry group S. Let $d_E(v, w)$ be the metric in $\mathbb{R}^E$ that is induced by the maximum norm. We propose using

$$d(v, w) := \min_{s \in S} d_E(s(v), w)$$

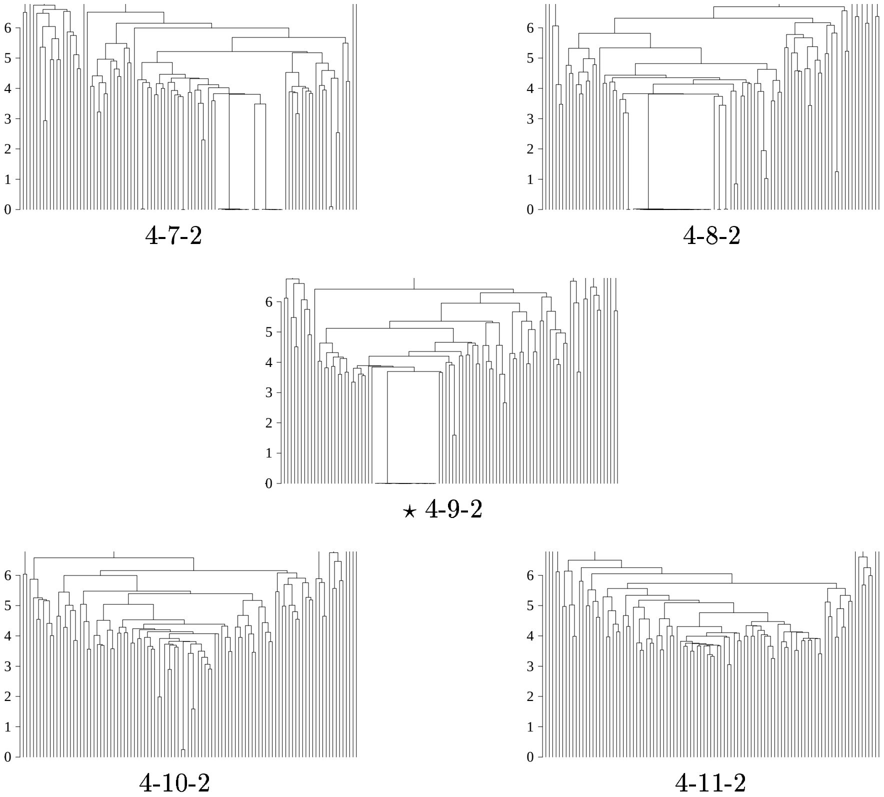
as a canonical metric in $\overline{W}$; then $r\colon (\mathbb{R}^E, d_E) \to (\overline{W}, d)$ is a continuous mapping. This enables us to work in the effective weight space knowing that similar network functions out_v and out_w should have a small distance $d(v, w)$. Although the calculation of $d(v, w)$ turns out to be intractable in the number of hidden nodes [4], we were able to implement an approximation based on efficient and sufficiently precise solutions of the Traveling-Salesman Problem.

Weight vectors $w^1, \dots, w^m$ resulting from repeated runs of an experiment can now be clustered. A good starting point for doing this is hierarchical clustering, where an arbitrary distance matrix D, e. g., $D_{ij} = d(r(w^i), r(w^j))$, can be supplied without the need to specify a number of clusters or cluster centers. Initially, each substructure contains a single vector. At each stage, the two closest substructures are combined to form one bigger substructure, the distance between substructures being defined as the maximal distance between its elements (a. k. a. complete linkage method). After $m - 1$ steps, all m weight vectors have been merged into one big structure. Dendrograms like in Figure 2 display this process as a tree, where the junction height indicates the distance between substructures.

Clusters, in our sense, are now represented by subtrees with a small height, and we require that a cluster have at least $\sqrt{m}$ elements in order to be statistically significant.

4 Robust Estimators

Usually, dendrograms reveal a great deal about the nature of the underlying learning algorithm. The presence of more than one cluster may indicate local optima. Singleton weight vectors with large distances to clusters can even indicate other types of deficiencies. Since cluster representatives are the typical results of an experiment, it is natural to compute the statistics of the test error *per cluster.* The deviation of the test error is negligible within a cluster. Hence, the mean, median, and minimum statistics almost coincide.



4-7-2

4-8-2

⋆ 4-9-2

4-10-2

4-11-2

Figure 2: Dendrograms

The absence of clusters suggests an arbitrariness in the weight vector estimates, allowing discretionary components that are compensated by other components — in other words, the presence of overfitting.

We generated dendrograms for the weight vectors of the respective experiments (see Figure 2). Dendrograms 4-7-2 and 4-8-2 display compact clusters. The cluster structure begins to decay in dendrogram 4-9-2, and has completely disappeared in 4-10-2 and the following dendrogram.

Table 2: Per-cluster statistic

Network	Mean	N/m	$N\overline{d}$
4-4-2	12000	13%	0.117
4-5-2	9000	16%	0.498
4-6-2	7690	9%	0.374
4-7-2	4380	10%	0.0603
4-8-2	269	24%	0.0683
⋆ 4-9-2	25.1	19%	0.0394

Table 2 shows the average test error per cluster, the relative cluster size N/m, and the geometrical property $N\overline{d}$ of cluster size N times the average distance $\overline{d}$ within the cluster.

Given the per-cluster statistic, we would clearly prefer the model 4-9-2.

5 Discussion

The proposed method is able to eliminate weight vectors that arise from deficiencies of neural-network learning procedures. We have used it to improve statistics that form the basis of model selection. It is, however, a general technique for improving the robustness of estimators. We have, e. g. successfully used it for improving confidence interval estimates for network outputs [2].

It is not necessarily restricted to feedforward networks. The same model selection method can, e. g. be applied to Boltzmann machines.

In our opinion, if a series of weight vectors *does not* contain a single cluster of similar solutions, the experiment setup (data set, learning algorithm, or network model) is likely to be inadequate. We conjecture that a purely geometrical approach might be sufficient to select adequate models (see Table 2). Defining a suitable measure for the presence and compactness of clusters might eventually lead to an algorithm that autofocuses to an appropriate network structure. One advantage would then be that the full data set could be used for training. These issues and others require further research.

References

[1] Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[2] Arnfried Ossen and Stefan M. Rüger. Weight space analysis and forecast uncertainty. *Journal of Forecasting*, in press.

[3] Brian D. Ripley. *Pattern Recognition and Neural Networks.* Cambridge University Press, 1996.

[4] Stefan M. Rüger and Arnfried Ossen. The metric structure of weight space. *Neural Procssing Letters*, 5(2):63–71, 1997.

[5] Halbert White. *Artificial Neural Networks — Approximation & Learning Theory.* Blackwell, Oxford, Cambridge, 1992.