Techniques for Dealing with Missing Values in Feedforward Networks

Peter Vamplew, David Clark, Anthony Adams, Jason Muench*

Artificial Neural Networks Research Group, Department of Computer Science, University of Tasmania

* permanent address: Department of Information Science and Engineering, University of Canberra

Abstract: Missing or incomplete data, a common reality, causes problems for artificial neural networks. In this paper we investigate several methods for dealing with missing values in feedforward networks. Reduced networks, substitution, estimation and expanded networks are applied to three data sets. We find that data sets vary in their sensitivity to missing values, and that reduced networks and estimation are the most effective ways of dealing with them.

Introduction

Artificial neural networks trained using backpropagation have been used for a wide variety of classification problems. In many real world problems, however, some of the data may be missing or incomplete. This causes particular problems for artificial neural networks as the distributed nature of the processing makes it very difficult to isolate effects due to one variable. The aim of this study is to compare different techniques for dealing with missing data. We assume a complete set of training data is available and a single hidden layer back-propagation network. This work builds on and extends earlier work of Vamplew and Adams[1].

Terminology

In this paper we will adopt the following terminology: V = number of variables; N, H, M = number of input, hidden, output nodes; and μ, σ = mean, standard deviation (of the training data)

Description of the data

We have applied the techniques to three data sets, the Iris, the Weedseed and the Handshape data set. All data are normalised to the range ±1.

The Iris data set[2] has been used widely as a testbed for statistical analysis techniques. The sepal length, sepal width, petal length, and petal width were measured on 50 iris specimens from each of 3 species, Iris setosa, Iris versicolor, and Iris virginica. There are thus 150 pieces of data, of which 100 are used for training and 50 for testing. The proportion of data which is correctly classified is high.

In the Weedseed data set[3], weed seeds are classified into one of ten types, based on seven measurements of dimensions of the seeds. The seed data consists of measurements of 398 different seeds, giving 40 examples of each seed type, apart from two types which have only 39 examples. A test set consisting of ten examples of each seed type is extracted from these data, and the remaining 298 examples are used as the training set. The proportion of data which is correctly classified is only moderate.

The Handshape data set[4] was developed by measuring the values returned by 16 sensors on various hand and finger joints when the hand was positioned in 20 different hand shapes used in the Auslan sign language. A training set of 2000 examples and a test set of 200 examples were created by adding noise to the 20 original examples. This data set contains many redundant inputs and a high proportion of the data can be classified correctly even with high levels of added noise. Thus for the purposes of this study, we used only 7 variables for each data point.

Experimental details

The networks are trained using the standard pattern presentation backpropagation algorithm which updates the weights after each

(randomly-chosen) presentation of the data. We use a fixed number of presentations and a fixed number of hidden nodes for each data set. In each trial, there are 10 runs, each with different starting weights. For all training the training rate was 0.1 and the momentum 0.

Data set	Presentations	Architecture (N H M)	# training points	# testing points
Iris	50,000	4 3 3	100	50
Weedseed	100,000	7 8 10	298	100
Handshape	100,000	7 9 20	2000	200

Methods

We investigated four general approaches for dealing with missing values: *Reduced network* - A separate network is trained for each missing value, each having V-1 nodes in the input layer; *Substitution* - Another value is substituted for the missing value; *Estimation* - A value for the missing value is estimated from the remaining data. This is substituted for the missing value; *Expanded network* - The standard network architecture is modified by the addition of extra input nodes in such a manner as to allow the network to distinguish between missing and complete data.

Reduced networks

V networks are trained, each with one variable absent (N = V-1). As each network is trained on all of the data available, and is of the right size for the data, it is expected to perform as well as possible. It should give an upper bound on what is possible and provide a baseline for judging other methods. A major disadvantage of this approach is that it requires N+1 networks to cater for all possible cases of one missing value. If it is to be applied to data points with two missing values, it will require an additional $N^2/2$ networks.

Substitution

Here a constant value for substitution is found for each variable that might be missing. This is then substituted in the complete network (ie. the network trained on complete data). We report on substituting the mean, median and zero (the mid-range). Other substitutions including the minimum, the maximum, "committeeing" the results from a series of values, and a random value were tried, but all gave poorer results and are not further reported.

The mean is calculated for each variable over all examples in the training set. It is fixed for all data points, that is the mean of the variable in the training set is used whenever that variable is missing. It is a very simple approach, easy to understand and requires no extra networks or training. The median and mid-ranges are treated similarly.

Estimation

The best estimate is derived by using a separate network to estimate the missing value from the values that are present. Thus there is only one classification network, supported by V estimating networks (each with N = V-1). Unlike the classifier networks, the supporting networks have a single analogue output. This approach relies on the input values being correlated, which is true for most real data sets.

Expanded networks

The aim here is to try to let the network know whether data is complete or has missing values. This is achieved by using two input nodes for each input variable. Three types of expanded networks are investigated, flagged, high/low and shadow weight networks.

In flagged networks, each data value is associated with a "value" node and a "flag" node. If the value is known, it is input to the value node and 0 is input to the flag node. If the value is missing, that attribute's mean is input to the value node and 1 is input to the flag node.

In high/low networks, each data value is associated with a "high" node and a "low" node. If the value is known, it is input to both the low and high nodes. If the value is missing, predetermined constants are input to the low and high nodes. Thus this technique essentially treats missing data as "fuzzy". Three sets of constants are tried: low = minimum value, high = maximum value; low = $\mu - \sigma$, high = $\mu + \sigma$; and low = $\mu - 2\sigma$, high = $\mu + 2\sigma$.

In the shadow weight networks, each data value is associated with a "standard" node and a "shadow" node. If the value is known, it is input to the standard node and 0 is input to the shadow node. If the value is missing, 0 is input to the standard node, and 1 is input to the shadow node. The main difference between the shadowed network and other expanded networks is in the training. This occurs in two phases. Firstly, a complete network (with no weights from shadow nodes to hidden layer nodes) is trained using complete data. Then the weights of the links between the shadow nodes and nodes in the hidden layer are trained. The aim is to create a network in which the performance of complete data is not degraded. The performance improves when several 'shadow' hidden nodes are added to the hidden layer. These nodes are completely connected to the input layer, but are only trained during the second phase of training and only used when the input data has missing values.

All of the expanded networks require extra training data. The training data is augmented with data where there have been substitutions for (single) values (that is, one value per data point). This means that there will be (V+1) times as much training data.

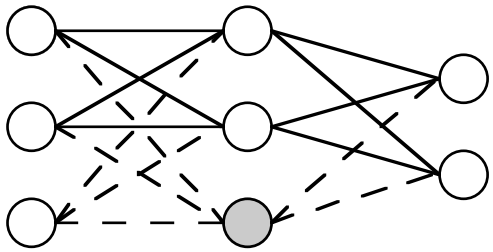

Figure 1: A shadow weights network. The bold lines indicate standard weights, the dashed lines and shaded circle represent shadow weights and node (in this case, the lowest input value is unknown)

Results

Results are set out in the tables below, shaded cells indicating a significant reduction in performance (using the reduced networks as a baseline). They show that only reduced networks and estimation are effective on all data sets.

The results also show a significant difference between data sets. For the Iris data, all techniques work well. For the Weedseed data, only reduced networks and estimation work well. For the Handshape data, all techniques worked well except substitution. A tentative conclusion here is that data which is hard to classify is more susceptible to missing values and needs more work if data points with missing values are to be useful.

Iris data

Variable missing	none	1	2	3	4
Reduced network	94.0	93.6	92.8	88.2	83.6
Substitute mean	94.0	93.8	95.8	73.4	85.2
Substitute median	94.0	93.8	94.0	72.0	88.0
Substitute zero	94.0	94.0	96.0	76.4	88.0
Estimation	94.0	94.0	96.0	94.0	93.6

Flagged	93.2	94.0	89.0	93.6	82.6
High / low (min max)	92.6	94.0	88.6	92.6	85.6
High / low ($\mu \pm \sigma$)	91.2	94.0	89.2	92.0	84.4
High / low ($\mu \pm 2\sigma$)	90.0	93.2	88.2	93.8	82.0
Shadow weights	93.2	93.0	92.8	93.2	84.8

Weedseed data

Variable missing	none	1	2	3	4	5	6	7
Reduced network	65.6	63.5	62.0	59.8	59.7	66.9	64.0	64.4
Substitute (mean)	65.6	51.4	44.9	49.0	38.9	48.0	41.5	56.9
Substitute (median)	65.6	53.4	46.8	48.9	39.5	45.5	42.4	62.0
Substitute (zero)	65.6	40.1	41.2	29.4	35.2	38.3	42.1	43.2
Estimation	65.6	63.1	63.4	59.5	59.0	63.4	62.5	66.2
Flagged	64.9	47	43	42.9	38.1	44.3	36.4	55.6
High / low (min max)	67.3	36.5	36.4	32.9	31.1	33.2	32.4	43.4
High / low ($\mu \pm \sigma$)	67.3	45.8	39.7	44.1	34.1	48.5	35.4	61.3
High / low ($\mu \pm 2\sigma$)	67.3	44.9	38.0	41.3	33.4	42.7	34.6	57.7
Shadowed weights	65.9	54.8	60.8	47.7	52.4	51.7	53.8	62.5

Handshape data

Variable missing	none	1	2	3	4	5	6	7
Reduced network	99.6	94.5	79.8	79.9	94.3	89.5	89.8	94.3
Substitute (mean)	99.6	79.9	78.7	70.5	86.4	81.2	88.1	82.4
Substitute (median)	99.6	80.2	76.9	70.4	83.1	76.6	87.6	76.0
Substitute (zero)	99.6	78.1	73.7	68.3	85.3	79.1	81.7	82.5
Estimation	99.6	94.9	79.7	78.2	95.0	89.3	89.1	93.6
Flagged	99.5	93.8	79.6	79.5	93.9	86.9	88.7	90.6

High / low (min max)	98.8	94.0	78.9	79.7	93.5	88.7	89.4	92.3
High / low ($\mu \pm \sigma$)	99.0	93.8	79.3	78.9	93.5	88.7	89.4	92.3
High / low ($\mu \pm 2\sigma$)	98.7	93.7	78.7	79.9	93.2	88.8	89.5	92.3
Shadowed weights	99.7	94.9	79.8	78.0	94.5	88.7	90.8	93.4

Equivalence of networks

Several of the networks described above are topographically equivalent, a simple transformation turning a network into another of different type which will give exactly the same output for equivalent data. There exist 1-1 mappings between all expanded networks, except for the shadow network with additional hidden layer nodes. Moreover, there is a mapping (not 1-1) from the complete network to each expanded network. Thus all expanded networks should perform equivalently, and they should all perform at least as well as substitution. The results are in broad agreement with this analysis.

Multiple Missing Values

The different missing values techniques vary in terms of the ease with which they can be scaled to handle examples with multiple missing inputs. Substitution and expanded networks cope readily with this situation as all that is required is to input the appropriate value(s) for each missing input value.

The main issue is whether these networks can generalise from the examples with single missing values seen during training, or whether it is necessary to also train the networks on examples with multiple missing values.

Estimation and reduced networks do not scale well to multiple missing values as they suffer from a combinatorial expansion in the number of networks to be trained. It is possible to adapt estimation to the case of multiple missing values however, by combining it with the substitution techniques. The estimate networks are trained and used to produce an estimate of each missing input as before, with the exception that substitution is performed for any missing values which are required as inputs to these estimate networks. The estimates produced in this way can then be used as inputs for the main classification network.

Vamplew[5] has shown that expanded networks and substitution generalise poorly with multiple missing values if trained using single missing values. He has also shown that the combines substitution/estimation technique is effective even when training is performed on single missing values.

Conclusions

1. All techniques perform well on complete data. Thus techniques such as shadow networks which are designed specifically to ensure that the performance on complete data does not degrade are not necessary.
2. Techniques involving fixed valued substitution for missing values do not perform as well as ones which estimate the missing values from the non-missing values for that data point.
3. Data sets vary in their sensitivity to missing values. If the data set is insensitive to missing values, all techniques are effective, but if it is sensitive to missing values, the choice of technique is important.

References

1 Vamplew, P and Adams, T., Missing Values in a Backpropogation Neural Net, in Leong, S and Jabri, M (eds)

Proceedings of the Third Australian Conference on Neural Networks, Sydney, Feb, 1992, 64-67.

2 Fisher, R.A., The use of multiple measurements in Taxonomic Problems, *Annals of Eugenics*, **7**, 1936, 179-188.

3 *Weedseed data*, obtained from University of Stirling, in private communication, Collier, P.A., Department of Computer Science, University of Tasmamia.

4 Vamplew, P. and Adams, A, The SLARTI System: Applying Artificial Neural Networks to Sign Language Recognition in *Proceedings of the Conference on Technology and Persons with Disabilities*, California State University, Northridge, 18-21 March, 1992.

5 Vamplew, P., *Computer Recognition of Hand Gestures*, Ph. D. thesis, Department of Computer Science, University of Tasmania, forthcoming.