
http://researchrepository.murdoch.edu.au/16226/
A STUDY OF THE USE OF SELF-ORGANISING MAP FOR SPLITTING TRAINING AND VALIDATION SETS FOR BACKPROPAGATION NEURAL NETWORK

Kok Wai Wong, Chun Che Fung and Halit Eren

Curtin University of Technology
School of Electrical and Computer Engineering
Kent Street, Bentley, Western Australia
Phone (619) 351 2575 Fax (619) 351 2584
Email: tfungcc@cc.curtin.edu.au

ABSTRACT: Validation has been used for the estimation of generalisation error of the Backpropagation Networks. The simplest way is to divide the available data into training and validation data sets. In this paper, an approach using the Self-Organising Map is proposed for the selection of the training and validation data sets. The results obtained from this study has shown that the proposed method provides a quick and reliable selection criteria and the overall training time is also reduced by applying the split-sample early stopping approach.

1. INTRODUCTION

Backpropagation Neural Network (BPNN) has been used in applications such as pattern classification, function approximation and regression problems. In this paper, focus has been concentrated on the use of BPNN in function approximation. The most important features of BPNN is the ability to generalise. After the network has been trained with the available data, it is desired that the network is capable of providing reasonable performance for data input other than the training data set. In other words, the network is expected to have good generalisation ability.

However, poor generalisation may occur due to underfitting or overfitting. In the first case, the network is undertrained such that the system error remains high at the end of the training process. This may be due to inefficient iterations or the number of weights in the network configuration is too small. In these cases, the problem can be overcome with increased number of training iterations or to use an alternative network configuration. In the second case, the phenomenon of overfitting [1] occurs when the network tries to fit all the data which may include substantial noise signals imposed on the underlying function. There are a few approaches to avoid overfitting in order to obtain a good generalised BPNN. These are: (1) model selection, (2) jittering, (3) weight decay, (4) early stopping and (5) bayesian estimation. In this paper, the early stopping approach [2,3] is investigated.

In the early stopping approach, a straightforward technique is to divide the available data into training and validation data sets. This is known as the split-sample validation approach [4]. In this method, the error calculated on the validation set is used to determine when to stop the training process. In this case, the ability of providing a well generalised BPNN is very much dependent on the validation set. The usual way is to perform several independent splits, and then average the results to obtain an overall estimate of the network performance. This method of validation is widely accepted, but it does suffer from the disadvantage of long training time due to multiple training sessions on different splits. In addition to being sensitive to the specific way of splitting the data and the long training time, this method also requires large number of available data. This paper deals with the two problems of how to determine the training and validation data sets, and to reduce the overall training time. In this study, it is assumed that there are sufficient data.

This paper reports the results from an investigation of using Self-Organising Map (SOM) to split the available data into training and validation data sets. The problem of petrophysical properties prediction from well log data is used as a case study to illustrate the proposed SOM data-splitting approach. Initially, a number of normal splitting approaches have been used. The results generated are then compared to the results obtained from the data sets based on the SOM data-splitting approach. In this study, it has shown that the SOM data-splitting approach has consistently provided better results.
2. SPLIT-SAMPLE VALIDATION

Split-sample validation is the most commonly used method for estimating the generalisation capability of a BPNN [4]. In this approach, a set of validation data which is not used in the training process is used to calculate the validation error. The validation error is found in the same way as the average training system error of the BPNN:

\[ V_v = \frac{1}{2P} \sum (T_p - O_p)^2 \]  

(1)

where

\[ V_v = \text{average validation error} \]
\[ P = \text{no. of patterns} \]
\[ T_p = \text{target patterns} \]
\[ O_p = \text{output patterns} \]

The stopping point in this method is suggested to be the point when the validation error starts to rise. This point also indicates that the generalisation ability starts to degrade. Figure 1 shows a typical graphical plot of the training system error and the validation error. When training starts, the errors for both data set will normally reduce. After many training iterations, the validation error normally starts to rise although the training error may continue to fall. The BPNN training process can be stopped at this point, as further training will result in overfitting.

![Fig. 1: Typical plot of training and validation error](image)

Using the above approach, the generalisation ability of the BPNN is highly dependent on the set of validation data. Hence, the splitting methods used is important. However, there are no rules to suggest the best splitting methods. Nevertheless, the validation data set should demonstrate two characteristics: (1) the validation set should be statistically close to the training set, and (2), the validation error should indicate the generalisation ability of the final BPNN and it can be used as the stopping criteria for the training process.

3. SOM DATA-SPLITTING APPROACH

The Self-Organising Map (SOM) has the ability to classify and cluster a set of data [5]. It performs clustering based on the "winner-take-all" competitive learning technique. Through the unsupervised learning process, it will cluster the data into different classes according to their characteristics. It basically performs an estimation on the probability density function of the data. After the data have been classified, a quantization error (QE) corresponding to each data point is generated which suggests the proximity of the data in that specific class.

In data-splitting, SOM can first be applied to classify the available data. After classification, a quantization error corresponding to each data point is generated. A number of splitting approaches on this set of classified data can be adopted:

1. **Lowest QE**: Select the data in each class which has the lowest quantization error and forms the training set. The remaining data are used as the validation set.

2. **Low-High QE**: Select all the data with the lowest or highest quantization error in each class and form the training set. The remaining data form the validation set.

3. **Mean QE**: The training set comprises of data from each class with the mean quantization error. Similar to above, the remaining data form the validation set.

Using this approach, data from each class are selected for training or validation. This ensures the similarity between the two data sets and it fulfills the first characteristics discussed in previous section. From subsequent studies, it is observed that the training and validation errors consistently demonstrate the trends as shown in Figure 1. This illustrates the second characteristics of the validation set.

4. CASE STUDY

In this paper, the problem of predicting petrophysical properties [6] from well log data has been selected to test the proposed SOM data-splitting approach. In this problem, measurements from several log instruments are obtained from field exploration. In order to obtain the petrophysical properties of the well, samples from various depths are examined using extensive laboratory analysis. The process is lengthy and expensive. These data obtained from geologists in this field are commonly
known as core data. Based on these core data, log analysts will make use of plots from the well logs and try to derive mathematical models which fit the core data. The models will then be used to predict the petrophysical properties from other depths or other wells within the region. Traditionally, log analysts have used graphical techniques, mathematical and statistical approaches. The use of artificial neural network in this field have emerged and reported recently [7, 8, 9]. From literature, the BPNN is the most commonly used. Based on core data, a network is trained and then used for lithology classification or properties prediction. The results reported have been very promising. However, there are no reports on studies of the generalisation ability of the networks in this application.

In this study, core data from five wells within a particular region are used. It is assumed that all these wells exhibit similar petrophysical properties. Core data from four wells are used as training data. The set of core data in the fifth well are reserved as testing set to verify the accuracy of the trained BPNN. There are a total of 85 training core data and 32 test data. In this set of data, a total of 9 input logs are available and the target petrophysical property to be predicted is porosity. Although there are other properties such as permeability, volume of clay, fluid content, lithology and structures which are of great interest to the log analysts, they have not been considered in this paper. The network configuration selected for this study comprises of nine input nodes and one output node. One hidden layer is used and eight nodes in the hidden layer has found to give the best performance.

For comparison purposes, two splitting approaches have been used. The splitting approaches are:

1. **Select one skip one**: Select first core data as training data and the next one as validation data. Repeat this selection until the end of the set of core data.

2. **Block selection**: Select first half of the available data set as training and the second half as the validation set.

A BPNN is first trained and tested without the use of any validation set. The training process is stopped when the average system error is reduced to minimum. This test is called Test 1 and it is used to compare with the results obtained from subsequent networks trained with the data-splitting approaches.

Test 2A to Test 4B are based on the data-splitting methods without the use of SOM. Tests 2 and 3 are based on the Select one skip one approach described above. The difference between the two is the swapping of the test and validation data sets. Test 4 is based on the Block selection approach. In each test, two stopping criteria have been used. Test A means that the training is stopped when the minimum error or maximum number of iterations is reached. Test B means that the training is stopped when the validation error starts to rise. A summary of these tests are listed below.

Test 1: Train with all available core data and aims to reduce the average system error to minimum.

Test 2A: Use the Select one skip one approach and aims to reduce the system error to minimum.

Test 2B: Same as Test 2A, but stop training when validation error starts to rise.

Test 3A: Same as Test 2A but the training data and validation data sets are interchanged.

Test 3B: Same as Test 3A, but stop training when validation error starts to rise.

Test 4A: Using Block selection approach and aims to reduce the system error to minimum.

Test 4B: Same as Test 4A but stop training when validation error starts to rise.

The total number of the training and validation data used in the above tests are shown in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>Training</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>85</td>
<td>0</td>
</tr>
<tr>
<td>Test 2A &amp; 2B</td>
<td>43</td>
<td>42</td>
</tr>
<tr>
<td>Test 3A &amp; 3B</td>
<td>42</td>
<td>43</td>
</tr>
<tr>
<td>Test 4A &amp; 4B</td>
<td>43</td>
<td>42</td>
</tr>
</tbody>
</table>

**TABLE 1: Number of training and validation data for Test 1 to Test 4B.**

For SOM data-splitting, the 85 training core data are classified into predefined classes. The maps selected are 6-by-6 (36 classes), 7-by-7 (49 classes) and 8-by-8 (64 classes). These dimensions are chosen because it is intended to keep the number of training data between one-third to two-third of all the available data. After classification, quantization errors for each data are generated.

Based on the 6-by-6 output classes, several tests have been carried out and descriptions of these tests are given as follows:-

Test 5A: Select one data from each class and two from those classes that have more data as the training set. The purpose of this selection is to maintain the same number of training and validation data as in Test
2A and Test 2B. The BPNN is trained to the minimum system error.

Test 5B: Same as Test 5B but stop training when validation error starts to rise.

Test 6A: Use *Lowest QE* approach and reduce the system error to minimum.

Test 6B: Same as Test 6A but training is stopped when validation error starts to rise.

Test 7A: Use *Low-High QE* approach and reduce the system error to minimum.

Test 7B: Same as Test 7A but stop training when validation error starts to rise.

Test 8A: Use *Mean QE* approach and reduce the system error to minimum.

Test 8B: Same as Test 8A but stop training when validation error starts to rise.

As for the classifications based on the 7-by-7 and 8-by-8 maps, only the *Mean QE* approach is used. This is because the number of data in each class has been reduced and there is no need to use more than one data point from each class. Tests 9A and 9B are carried out on the 7-by-7 classes data. Finally, the Tests 10A and 10B are performed on the core data classified into the 8-by-8 map.

Test 9A: Use *Mean QE* approach and reduce the system error to minimum.

Test 9B: Same as Test 9A but stop training when validation error starts to rise.

Test 10A: Use *Mean QE* approach and reduce the system error to minimum.

Test 10B: Same as Test 10A but stop training when validation error starts to rise.

The total number of the training and validation data used in all these SOM data-splitting methods are shown in Table 2.

<table>
<thead>
<tr>
<th></th>
<th>Training</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 5A &amp; 5B</td>
<td>43</td>
<td>42</td>
</tr>
<tr>
<td>Test 6A &amp; 6B</td>
<td>30</td>
<td>55</td>
</tr>
<tr>
<td>Test 7A &amp; 7B</td>
<td>53</td>
<td>32</td>
</tr>
<tr>
<td>Test 8A &amp; 8B</td>
<td>30</td>
<td>55</td>
</tr>
<tr>
<td>Test 9A &amp; 9B</td>
<td>38</td>
<td>47</td>
</tr>
<tr>
<td>Test 10A &amp; 10B</td>
<td>42</td>
<td>43</td>
</tr>
</tbody>
</table>

**TABLE 2: Number of training and validation data used from Test 5A to Test 10B.**

5. RESULTS AND DISCUSSIONS

The tests performed in this study were carried out on a Pentium-90 computer. All the application software were developed under C environment. Having trained with data prepared from the tests mentioned in the previous section, the BPNNs were tested with the 32 core data in the fifth well for the prediction of porosity. During the training stage, all the tests were aimed to reduce the system error to 0.001 or stop after 50,000 iterations. The results obtained from these BPNNs are then compared with the core porosity values. Two statistical similarity and dissimilarity measures have been calculated for comparison purposes[10], they are:

**Percent similarity coefficient (PERCENT):**

\[
PSC_p = \frac{\sum_{i} \min(X_{a,i},X_{d,i})}{\sum_{i} (X_{a,i}+X_{d,i})}
\]

(2)

**Euclidean distance (EUCLID):**

\[
 ED_p = (\sum_{i} (X_{a,i} - X_{d,i})^2)^{1/2}
\]

(3)

where \(i\) and \(j\) represent the two data to be compared and \(k\) represents the pattern rows.

The results from Test 1 to Test 4B are shown in Table 3, and the results for various SOM data-splitting method are shown in Table 4. Test 4B could not be carried out because the validation error started to rise from the beginning of the training. This may suggest that the validation data and training data are statistically dissimilar.

<table>
<thead>
<tr>
<th>TEST</th>
<th>PERCENT</th>
<th>EUCLID</th>
<th>Training Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>91.865</td>
<td>0.8</td>
<td>25 min</td>
</tr>
<tr>
<td>2A</td>
<td>89.727</td>
<td>1.046</td>
<td>10 min</td>
</tr>
<tr>
<td>2B</td>
<td>93.199</td>
<td>0.627</td>
<td>13 sec</td>
</tr>
<tr>
<td>3A</td>
<td>92.842</td>
<td>0.67</td>
<td>2.8 min</td>
</tr>
<tr>
<td>3B</td>
<td>91.17</td>
<td>0.723</td>
<td>4 sec</td>
</tr>
<tr>
<td>4A</td>
<td>85.369</td>
<td>1.299</td>
<td>6.4 min</td>
</tr>
<tr>
<td>4B</td>
<td>NIL</td>
<td>NIL</td>
<td>NIL</td>
</tr>
</tbody>
</table>

**TABLE 3: Results and training time for Test 1 to Test 4B.**

From Table 3, Test 1 gives a relative good result of 91.9% percentage of similarity. However, the training time was close to half an hour and the number of training data used was 85. The system error did not reach 0.001 and the test stopped at the 50,000th iterations. In cases of using the data-splitting approach, Test 2B gives the best result and the training time was 13 seconds. Test 3B also gives a result which is compatible to Test 1 and only 4 seconds were used. However, it is not conclusive that the split-sample validation could guarantee a better result for early stopping. It is shown from the table that result from Test 3A is better than Test 3B and Test 1.
From practical viewpoint, this suggests that the user has to repeat the splitting process in order to find the best splitting arrangement and this approach is commonly adopted. This could be very time consuming especially when there are large amount of data. Some of the data may be statistically similar while others exhibit different characteristics. The results will be grossly incorrect if the selected training data are all of similar characteristics while other data with different characteristics are being left out. This phenomenon is suggested in the case of Test 4A and 4B.

<table>
<thead>
<tr>
<th>TEST</th>
<th>PERCENT</th>
<th>EUCLID</th>
<th>Training Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5A</td>
<td>86.748</td>
<td>1.261</td>
<td>23.8 min</td>
</tr>
<tr>
<td>5B</td>
<td>93.731</td>
<td>0.591</td>
<td>1.3 min</td>
</tr>
<tr>
<td>6A</td>
<td>88.236</td>
<td>1.034</td>
<td>4.8 min</td>
</tr>
<tr>
<td>6B</td>
<td>91.998</td>
<td>0.662</td>
<td>3 sec</td>
</tr>
<tr>
<td>7A</td>
<td>89.941</td>
<td>0.956</td>
<td>24.5 min</td>
</tr>
<tr>
<td>7B</td>
<td>92.572</td>
<td>0.653</td>
<td>48 sec</td>
</tr>
<tr>
<td>8A</td>
<td>90.935</td>
<td>0.849</td>
<td>23 min</td>
</tr>
<tr>
<td>8B</td>
<td>91.678</td>
<td>0.693</td>
<td>4 sec</td>
</tr>
<tr>
<td>9A</td>
<td>89.11</td>
<td>1.22</td>
<td>23.7 min</td>
</tr>
<tr>
<td>9B</td>
<td>91.74</td>
<td>0.762</td>
<td>34 sec</td>
</tr>
<tr>
<td>10A</td>
<td>90.917</td>
<td>0.735</td>
<td>5.5 min</td>
</tr>
<tr>
<td>10B</td>
<td>93.33</td>
<td>0.637</td>
<td>5 sec</td>
</tr>
</tbody>
</table>

### TABLE 4: Results and training time using SOM data-splitting

As from Test 5A to Test 10B, by using the SOM classification for data splitting, the early stopping approach performs better whichever ways the training data were selected from each class. The best result is obtained from Test 5B which gives a percentage of similarity of 93.7% and the training time was 1.3 minutes. In Tests 6B and 8B, the results were 92% and 91.7% respectively. It is worth to note that only 30 training data were used in both cases and the training time was less than 4 seconds. Test 10B used 42 training data and it gives a result of 93.3%. The value is again better than the result from Test 2B above.

Although their similarity coefficients do vary between different splitting methods, it is shown that the results from the SOM data-splitting and early stopping approach are always better. It is important that the training data must include all the essential characteristics, and the use of statistically similar validation data for verification of the network’s generalisation ability. Data obtained from the SOM approach fulfill these requirements. Another advantage of using the SOM data-splitting approach is that the overall training time is greatly reduced as it is not necessary to repeat and try different data-splitting processes.

In order to illustrate the generalisation capability of the networks, Figure 2 shows a graphical plot from Test 5 to compare the predicted porosity with the core porosity. Test 5A is the plot without early stopping while Test 5B was based on the SOM approach. It can be observed that the predicted porosity with validation gives a better result as compared to the one without validation. Figure 3(a) and 3(b) are cross-plots of the trained network outputs with respect to the core training data also from Test 5. Figure 3(a) shows that the output from Test 5A without any validation which gives better correlation between the training data and the network output. Figure 4(a) and 4(b) are cross-plots of the predicted outputs from Test 5A and 5B with respect to the testing core data in the fifth well. These data have not been presented to the network during the training or validation phases. It can be observed that overfitting has taken place in Test 5A as shown in Figure 4(a). Test 5A performed well in the training process as demonstrated in Figure 3(a) but failed to predict reasonably for data which were not included in the training process. On the other hand, the SOM data-splitting method provided better results as illustrated in Figure 4(b). Similar results were also observed from the other tests (Test 6 to Test 10).
6. CONCLUSION

This paper has investigated and reported the use of SOM as a data-splitting approach for the selection of training and validation data sets. These data sets are used to train BPNN based on split-sample validation early stopping method. The results have shown that the use of the SOM approach is consistent in providing a good generalised network and the training time is reduced while avoiding the overfitting problem. The SOM has also ensured that the training data set has enough information to include the underlying function, as well as the generation of a statistically similar validation set. This is useful in the application of split-sample validation and early stopping for BPNN training.

7. REFERENCE


162