PERGAMON

International Journal of Rock Mechanics and Mining Sciences 36 (1999) 29–39

International Journal of
Rock Mechanics
and Mining Sciences

Application of neural networks for the prediction of the unconfined compressive strength (UCS) from Equotip hardness

F. Meulenkamp, M. Alvarez Grima

Section Engineering Geology, Department of Applied Earth Sciences, Faculty of Civil Engineering, Delft University of Technology, Deflt, Netherlands

Accepted 10 September 1998

Abstract

This paper presents the application of a neural network for the prediction of the UCS from hardness tests on rock samples. To investigate the suitability of this approach, the results of the network are compared to predictions obtained by conventional statistical relations.

The network was trained to predict the UCS based on the hardness, porosity, density, grain size and rock type information of a rock sample. A dataset containing 194 rock sample records, ranging from weak sandstones to very strong granodiorites, was used to train the network with the Levenberg–Marquardt training algorithm. Two sets, each containing 17 rock samples, were used to validate the generalization and prediction capabilities of the network.

1. Introduction

We developed a computer program to predict the unconfined compressive strength (UCS) of rock samples using a back-propagation supervised neural network. A neural network is a computational system composed of nodes (called neurons) and connections between these nodes. When trained with a sufficient number of training records (i.e. rock samples) the network recognizes the different existing relations for the given problem.

Commonly, the UCS of rock samples is determined using either the laboratory UCS test or UCS correlated index tests. In the UCS test, the strain of a rock sample is measured while the unconfined compressive force is increased. The stress at which the sample fails is referred to as the peak strength. The second approach uses index tests to predict the UCS instead of measuring it. The major advantages of the use of index tests are the low costs involved and their flexibility.

In 1993 Verwaal and Mulder [1] investigated the possibility to predict the UCS from rock hardness information using the Equotip hardness tester [2]. On the basis of their findings Asef [3] concluded that the Equotip hardness tester can serve as an index test for rock strength properties. Both the research of Verwaal and Mulder [1], as well as the investigation of Asef [3] resulted in statistical UCS predicting models.

The major demerit of statistical relations (e.g. regression analysis) is the prediction of mean values only. Consequently, low UCS values are overestimated, while high UCS values are underestimated. A neural network does not force the predicted UCS value to be a mean value, thus preserving and using the existing variance of the measured data.

This paper presents a neural network capable of predicting the UCS of rock samples. The rock samples, used to train the network, were assembled in the Falset area (near Tarragona, NE) and are of Carboniferous to Cretaceous age. Each sample is described by six different rock characteristics. The rock characteristics which function as the training input of the network are: Equotip reading, porosity, density, grain size and rock type, while the corresponding UCS value functions as the training output of the network.

2. Rock sample characterization

Rock strength is known to be related to the texture and mineralogical composition of the rock. Most critical are: mineralogical composition (rock type), density, porosity and grainsize [4]. Therefore, the network pre-

0148-9062/99/$ - see front matter
PII: S0148-9062(98)00173-9

dicts the UCS while knowing the Equotip hardness reading, porosity, density, grain size and the type of rock of a sample. The latter two descriptors are specified in a linguistic manner. So, a conversion into numerical values is necessary in order to use this information in both the statistical and neural network models. Because most geologists are unfamiliar with the Equotip hardness tester, some background information of this device is given first.

2.1. The Equotip hardness tester

The hardness[1] of a rock is defined in this investigation as the response of the rock material to an impacting device, in this case the Equotip tester (Fig. 1).

The Equotip hardness tester [2] is composed of an impact device (ball of tungsten carbide) and an electronic rebound velocity measuring device. The impact ball is "fired" against the material surface, and when the ball rebounds into the device it generates an induced current in the coil. The measured voltage of this current indicates the rebound velocity. The ratio between the rebound and the impact velocity can be considered as the hardness value L [6]:

$$L = \frac{V_{\text{rebound}}}{V_{\text{impact}}} \times 1000. \quad (1)$$

The impact energy of the device is approximately 3 Nmm, yielded by the 3 mm diameter tip. Several types of impact testers (probes) are available for the Equotip. Due to the stronger correlation between the readings of probe C with the UCS values compared to other probes (e.g. probe D), probe C is used in this study to determine the hardness of all rock samples [3, 7].

2.2. Rock description conversion

Besides the measured porosity, density and UCS value, also a description of the texture and rock type is given for the individual rock samples in the data set. However, to make use of the linguistic descriptions in neural networks and statistics, those descriptions have to be converted into numerical values. It is important to be consistent in such a conversion (Table 1). Thus, the type of rock is classified according to its strength. The texture is rated from fine grained to conglomerate.

[1] Hardness is a loosely defined term, because the hardness determined is the result of the type of impact process (geometry and mechanical properties of the impacting device) and the rock petrography (orientation and size of the minerals, grain shape, presence of microcracks) [15].

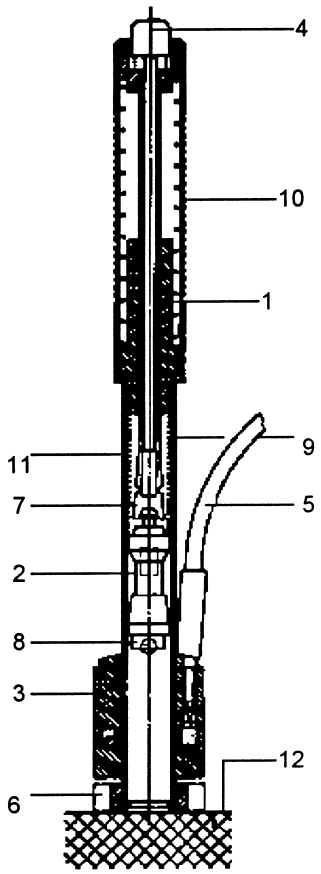

Fig. 1. Schematic design of the Equotip hardness tester: (1) loading tube, (2) guide tube, (3) coil with coil holder, (4) release button, (5) connection cable, (6) support ring, (7) impact body, (8) test tip, (9) impact spring, (10) loading spring, (11) catch chuck and (12) material to be tested.

2.3. Rock sample data

The data set used to develop the network model consists of sandstones, limestones, dolomitic limestones, dolomites, granites and granodiorites. Table 2 lists the descriptive statistics of the data. The minimum and maximum values given in Table 2 of the different rock sample properties function as the boundary conditions of the presented network model.

3. Neural network model

Neural networks are simplified models of the biological structure found in human brains. These models consist of elementary processing units, called neurons. It is the large amount of interconnections between these neurons and their capability to learn from data, which provide for a strong predicting and classification tool. Therefore, the neural network approach is selected to solve the complex relations between the

Table 1
Conversion values of the rock description compared to BS 5930 (1981) values. Mean UCS values in MPa

Grain size	BS 5930 (mm)	Num. Value	Rock type	Mean UCS	BS 5930 (1981)	Num. Value
fine grained	<0.06	1	granodiorite	257	extremely strong	1
medium	0.06–0.6	2	granite	230	extremely strong	2
coarse grained	0.6–2	3	dolomite	131	very strong	3
conglomerate	>2	4	dolomitic limestone	133	very strong	4
			limestone	120	very strong	5
			sandstone	35	moderately strong	6

Table 2
Descriptive statistics of the assembled rock sample data. Valid number of records is 226 (including training and test sets)

Property	Range	Minimum	Maximum	Mean	Standard Deviation	Variance
UCS (MPa)	274.4	4.8	279.3	118.5	64.3	4140.6
Equotip (–)	599.0	298.9	897.9	687.1	97.6	9542.3
Density (kN/m^3)	13.9	17.4	31.3	26.3	2.4	5.7
Porosity (%)	38.5	0.01	38.5	2.5	5.7	33.1
Grain size (–)	3.0	1.00	4.0	1.5	1.0	1.0

different rock properties, resulting in an UCS predicting model.

The network[2] model presented in this article is a supervised back-propagation neural network, making use of the Levenberg–Marquardt approximation. Before discussing the network architecture and its predicting capabilities a brief introduction to the training procedure is given first.

3.1. Training a network with Back-Propagation

A network needs first to be trained before interpreting new information. Several different algorithms are available when training neural networks. However, the back-propagation algorithm is the most popular, for it provides the most efficient learning procedure for multilayer neural networks. Also the fact that back-propagation algorithms are specially capable to solve predicting problems [9] make them so popular.

The Back-Propagation networks consist of an input layer, one or more hidden layers and an output layer (Fig. 2). Each layer is composed of different processing units (also called neurons), connected to the units of the next layer. A transfer function processes input data that reach the corresponding neuron. To differentiate between the different processing units, values called biases are introduced in the transfer functions. These biases are referred to as the temperature of a neuron [10].

During training of the network, data is processed through the network, until it reaches the output layer (*forward pass*). In this layer the output is compared to the measured UCS value (the "true" output). The difference or error between both is processed back through the network (*backward pass*) updating the individual weights of the connections and the biases of the individual neurons. The input and output data are mostly represented as vectors called training pairs. The process as mentioned above is repeated for all the training pairs in the data set, until the network error converged to a threshold minimum defined by a corresponding cost function; usually the root mean squared error (RMS) or summed squared error (SSE).

The algorithm used to train our network makes use of the Levenberg–Marquardt approximation. This al-

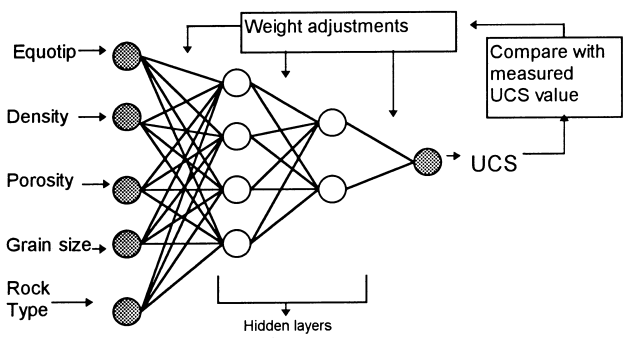

Fig. 2. Back-propagating neural network.

[2] The network is developed in Matlab 5.0, using also a neural network toolbox [8].

gorithm is more powerful than the common used gradient descent methods, because the Levenberg–Marquardt approximation makes training more accurate and faster near minima on the error surface [8]. The method is as follows:

$$\Delta \mathbf{W} = (\mathbf{J}^T\mathbf{J} + \mu\mathbf{I})^{-1}\mathbf{J}^T e. \tag{2}$$

In Eq. (2) the adjusted weight matrix $\Delta\mathbf{W}$ is calculated using a Jacobian matrix $\mathbf{J}$, a transposed Jacobian matrix $\mathbf{J}^T$, a constant multiplier μ, a unity matrix $\mathbf{I}$ and an error vector e. The Jacobian matrix contains the weights derivatives of the errors:

$$\mathbf{J} = \begin{vmatrix} \dfrac{\partial E}{\partial w_{ij}} & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \dfrac{\partial E}{\partial w_m} \end{vmatrix}. \tag{3}$$

If the scalar μ is very large, the Levenberg–Marquardt algorithm approximates the normal gradient descent method, while if it is small, the expression transforms into the Gauss–Newton method [8]. For more detailed information the readers are referred to Lines and Treitel [11].

After each successful step (lower errors) the constant μ is decreased, forcing the adjusted weight matrix to transform as quickly as possible to the Gauss–Newton solution. When after a step the errors increase the constant μ is increased subsequently. The weights of the adjusted weight matrix (Eq. (2)) are used in the *forward pass*. The mathematics of both the forward and *backward pass* are briefly explained in the following.

3.1.1. Forward pass

The net input (net_{pj}) of neuron j in a layer L and the output (o_{pj}) of the same neuron of the pth training pair (i.e. the inputs and the corresponding UCS value of a rock sample) are calculated by

$$\text{net}_{pj} = \sum_{n=1}^{\text{last}} w_{jn} o_{pn}, \tag{4}$$

where, the number of neurons in the previous layer $(L-1)$ are defined by $n = 1$ to last neuron and the weights between the neurons of layer L and $L-1$ by w_{jn}. The output (o_{pj}) is calculated using the *logarithmic sigmoid* transfer function:

$$O_{pj} = f_{pj}(\text{net}_{pj}) = \frac{1}{1 + e^{-(\text{net}_{pj} + \theta_j)}}, \tag{5}$$

where, θ_j is the bias of neuron j.

[3] The network overtrained remembers insignificant information of the training data. An underfitted network is unable to predict with acceptable accuracy.

The outputs of all neurons of layer L are calculated using Eq. (5) and processed again with Eq. (4) to generate new inputs for the next layer. This process continues until the output layer is reached.

3.1.2. Backward pass

In general the output vector, containing all o_{pj} of the neurons of the output layer, is not the same as the true output vector (i.e. the measured UCS value). This true output vector is composed of the summation of t_{pj}. The error between these vectors is the error made while processing the input–output vector pair and is calculated as follows:

$$E_p = \frac{1}{2}\sum (t_{pj} - o_{pj})^2. \tag{6}$$

When a network is trained with a database containing a substantial amount of input and output vector pairs the total error E, (sum of the training errors E_p) can be calculated.

$$E_t = \sum E_p. \tag{7}$$

To reduce the training error, the connection weights are changed during a completion of a forward and backward pass by adjustments (Δw) of all the connections weights w. Eqs. (2) and (3) calculate those adjustments. This process will continue until the training error reaches a predefined target threshold error (i.e. cost function).

3.2. Network architecture

Designing a network architecture requires more than selecting a certain number of neurons, followed by training only. Especially phenomena such as overfitting and underfitting[3] should be recognized and avoided in order to create a reliable network. Those two aspects — overfitting and underfitting — determine to a large extent the final configuration and training constraints of the network.

Mostly overfitting occurs when a network is trained using too many training epochs. Although the training error might reach its global minimum in this situation, the network is unable to generalize sufficiently, because the trained network remembers insignificant details of the training data. In fact, aiming at an absolute minimum in the training error is only a good approach when the training set contains many more cases than there are degrees of freedom (i.e. biases and weights) in the network [12].

Fig. 3(a) and (b) depict overfitting and underfitting, indicating that the lowest training error not necessarily results in good predictions of the UCS. There are two

underlying conditions resulting in an overfitting network:

- The network has too many neurons in relation to the amount of training cases.
- The network is trained with too many training epochs.

A strategy to avoid overfitting is to investigate the goodness of the predictions using different architectures. Afterwards, the configuration with the smallest root mean square error of the predictions is chosen.

The number of training epochs is kept constant during this procedure [7]. However, after the choice of the configuration the number of training epochs that will not result in an over- or underfitting network still has to be determined.

For this reason another method was applied to configure the network to omit overfitting. The strategy was as follows. First, an empirical formula is used to establish the number of neurons in the network, while keeping the configuration of the network unchanged and finally to settle the optimum number of training epochs [Fig. 4(a)]. The following equation was used to determine the number of hidden neurons [13]:

$$N_h = \frac{n}{k(m+p)}, \tag{8}$$

where, N_h is the number of hidden neurons, n is the number of cases in the training set, k is a noise factor[4], m and p are the amount of input and output neurons. N is 194, m is 5 (Equotip reading, density, porosity, grainsize and rock type as input nodes), p is 1 (UCS), k is 4 (considering measuring errors). Using these parameters in Eq. (8), around six hidden neurons seem to be acceptable for the network design.

Secondly, a suitable configuration of those neurons had to be chosen. Although different configurations are possible, a configuration having four neurons in the first hidden layer and two neurons in the second hidden layer, was chosen (networks with only one hidden layer generally give unsatisfactory predicting models [14]).

Finally, the determination of the most suitable number of training epochs was established. A test set was used to evaluate the predicting capabilities of the network after each epoch [Fig. 4(a) and (b)]. The weights and biases of the network were stored whenever the error in network predictions reached a minimum (this gave the optimum number of training epochs).

Another test set was processed through the configuration, having evidently the same initial weights and biases, to conclude whether the network can generalize. The network generalizes when the error of the UCS predictions of this set reaches its lowest possible value after the same number of training epochs as the first test.

Fig. 4(a) and (b) show indeed that for both test sets the lowest root mean squared error is reached after four training epochs. At first glance, this number of training epochs seems to be rather small to correspond to a global minimum, but this is in fact the effect of the Levenberg–Marquardt training algorithm. The Levenberg–Marquardt algorithm is a very powerful algorithm, able to train a network very quickly, up to 500 times faster than gradient decent algorithm, which is normally used in the back-propagation procedure

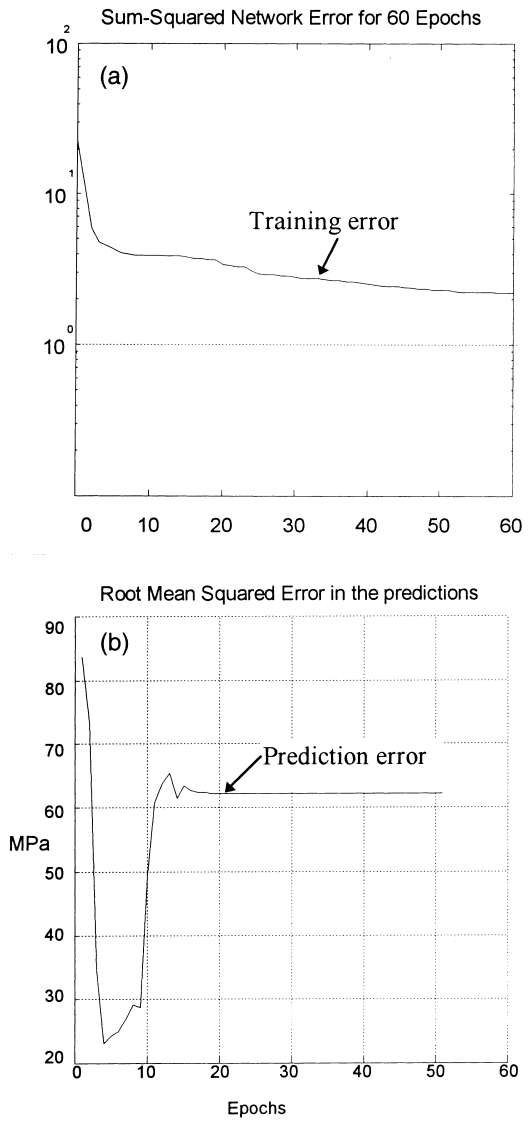

Fig. 3. (a) Training error (SSE) vs number of epochs. (b) Error in the predictions. The root mean square error (RMSE) of the network predictions vs the number of training epochs is presented. Note that the optimum number of training epochs does not necessarily coincides with a small training error [see (a)].

[4] $4 < k < 10$. k increases when there exists a substantial amount of noise in the data.

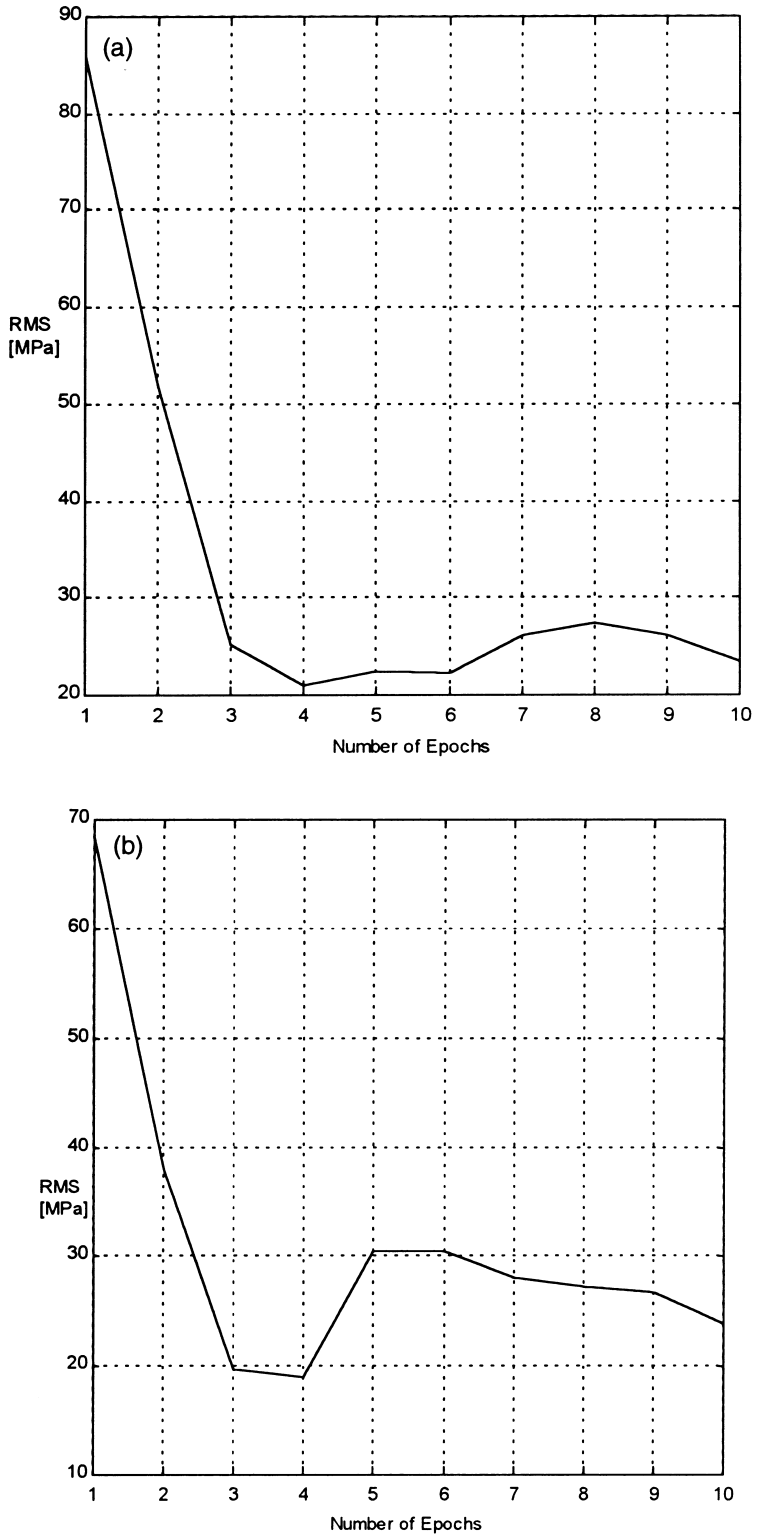

Fig. 4. (a) The root mean square error (RMSE) in MPa of the predictions vs the number of training epochs using test set A. (b) The root mean square error (RMSE) in MPa of the predictions vs the number of training epochs using test set B.

(Fig. 5). Other reasons for the quick convergence of the network are the small training set and the scaling[5] of the training data between 0 and 1 [15].

3.3. Relative strength estimations of the input parameters

To investigate the strength influences of the different inputs on the predicted UCS of a rock sample a relative strength estimation (RSE) algorithm, proposed by Yang and Zhang [16], was implemented in the model. Although the RSE algorithm is designed to investigate the hierarchical importance of input parameters used in a neural network, it also provides a strong tool to investigate the effects of the individual parameters on the estimated network output. Fig. 6 shows the relative strength estimations of the Equotip hardness reading, density, porosity, rock type and grainsize. From Fig. 6 it is possible to see that the network established a positive relation between both the Equotip reading and density and the predicted UCS value, while a negative relation is established for the other parameters. Meaning, for example, that when the value of rock type increases (e.g. a weaker rock type) the predicted UCS will decrease.

3.4. UCS predicting capabilities

In this section the predicting capability of the neural network is evaluated. Furthermore, a comparison with UCS predictions by statistical models is done in order to evaluate the results of the network. The results of the UCS predicting capabilities of both the statistical and the neural network models are discussed.

3.4.1. Statistical UCS predictions

Two different statistical UCS predicting models were used: a curve fitting relation, which proceeds on the findings of Verwaal and Mulder [1] and Asef [3], and a Multivariate Regression relation. The reason to evaluate the prediction results of the network with a multivariate regression relation is the possibility to introduce different independent variables in the model just as the network model does.

3.4.1.1. Curve fitting relation. For the generation of the curve fitting relation the statistical software package SPSS 7.5 [17] was used. An algorithm searches for a curve fit through the observed UCS values using a predefined function (e.g. power function), which explains the most of the variance of the observed data.

[5] Necessary for the transfer functions used in the neurons.

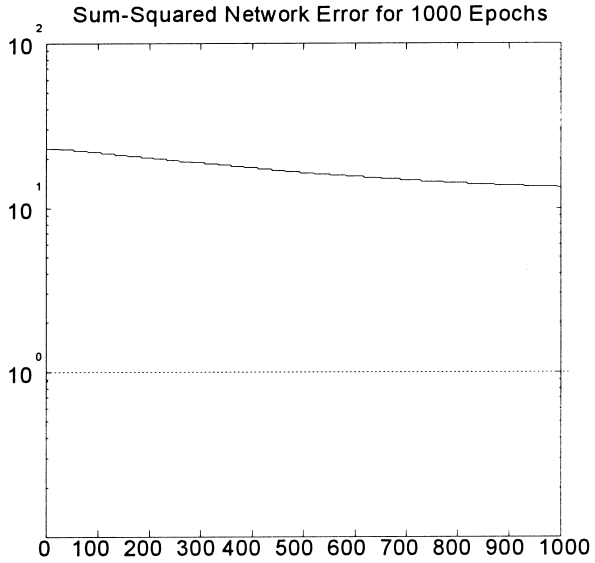

Fig. 5. Training error of the network using the gradient–descent algorithm. Note that the training error has still not converged after 1000 epochs near to the training error using the Levenberg–Marquardt algorithm (Fig. 3).

The following curve fit using a power function resulted in the most accurate UCS predicting relation [7]:

$$\mathrm{UCS} = 1.75 \cdot 10^{-9} L^{3.8}. \tag{11}$$

In Eq. (11) L refers to the Equotip value. The curve fitting relation (Eq. (11)) has a coefficient of determination (R^2) of 0.806, meaning that around 80% of the

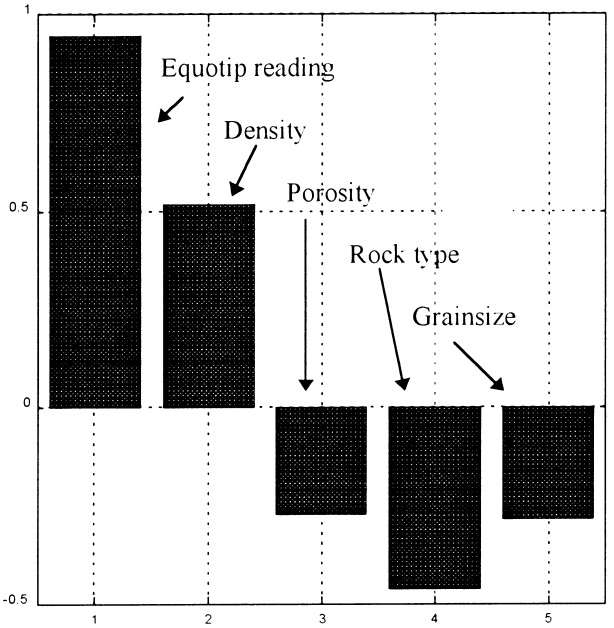

Fig. 6. Relative strength estimations (RSEs) of the different input parameters. A positive action on the output is given by a positive value on the Y-axis, while a negative action is given by a negative value.

Table 3
Predicted UCS values of test set A, using different predicting models

Number	Equotip	Rock type	Grain size	UCS (MPa) (measured)	UCS (MPa) (ANN)	UCS (MPa) (Eq. (11))	UCS (MPa) (Eq. (12))
(1)	827.4	granodiorite	fine	269.7	240.7	214.0	244.3
(2)	853.9	granite	fine	262.0	236.9	241.2	229.3
(3)	870.9	granodiorite	coarse	256.6	254.0	260.0	231.8
(4)	869.1	granite	fine	189.1	238.2	257.9	230.6
(5)	720.0	limestone	fine	168.8	135.0	126.2	127.7
(6)	750.8	limestone	fine	162.8	146.8	147.9	135.8
(7)	723.0	limestone	fine	162.3	139.4	128.2	129.9
(8)	724.0	limestone	medium	112.6	117.5	128.8	113.9
(9)	661.5	limestone	medium	99.0	97.1	91.4	98.8
(10)	646.7	dolomite	conglomerate	85.6	80.6	83.9	107.2
(11)	659.4	limestone	medium	73.4	95.9	90.3	98.1
(12)	648.9	sandstone	medium	72.7	72.8	85.0	68.1
(13)	564.4	limestone	fine	62.0	77.1	50.0	85.7
(14)	576.1	sandstone	medium	52.3	58.1	54.1	43.2
(15)	574.0	sandstone	medium	51.0	57.7	53.3	42.5
(16)	698.5	sandstone	conglomerate	43.8	61.4	112.4	44.2
(17)	591.3	sandstone	conglomerate	35.4	46.1	59.7	17.2

variance in the data is explained by the relation. The UCS prediction results of the curve fitting relation for test set A and B are given in Tables 3 and 4 and Figs. 7 and 8.

3.4.1.2. Multivariate regression relation. A multivariate relation was established using the same input variables as the neural network model: Equotip value (L), density, porosity, grainsize and rock type. This resulted in the following equation:

$$\text{UCS} = 0.25L + 28.14\ \text{density} - 0.75\ \text{porosity} - 15.47\ \text{grain size} - 21.55\ \text{rock type}. \quad (12)$$

The relation in Eq. (12) has a coefficient of determination (R^2) of 0.903 and an estimated standard error of 42.1 MPa. Figs. 7 and 8 and Tables 3 and 4 give the prediction results of the multivariate regression

Table 4
Predicted UCS value of test set B, using different predicting models

Number	Equotip	Rock type	Grain size	UCS (MPa) (measured)	UCS (MPa) (ANN)	UCS (MPa) (Eq. (11))	UCS (MPa) (Eq. (12))
(1)	862.1	granodiorite	coarse	274.6	253.9	250.1	230.0
(2)	855.5	granite	fine	206.3	235.8	242.9	227.8
(3)	798.4	granite	fine	186.8	217.7	186.8	213.5
(4)	739.0	limestone	fine	162.3	144.1	139.3	131.7
(5)	689.0	limestone	fine	142.3	131.2	106.7	121.1
(6)	633.7	limestone	fine	137.9	108.5	77.7	105.6
(7)	668.3	limestone	fine	136.2	137.2	95.0	124.5
(8)	652.9	limestone	fine	132.7	119.3	87.0	112.2
(9)	750.3	dolomitic lmst	fine	119.8	156.4	147.5	157.7
(10)	629.3	limestone	fine	119.4	114.6	75.6	110.1
(11)	668.0	limestone	medium	102.9	100.1	94.6	100.6
(12)	631.3	sandstone	fine	91.7	91.2	76.5	83.9
(13)	667.4	sandstone	medium	75.9	77.8	94.6	75.2
(14)	607.7	dolomite	fine	75.2	99.7	66.2	142.0
(15)	595.0	sandstone	conglomerate	38.0	42.9	61.6	14.8
(16)	400.5	mudstone	fine	6.6	12.5	13.6	25.4

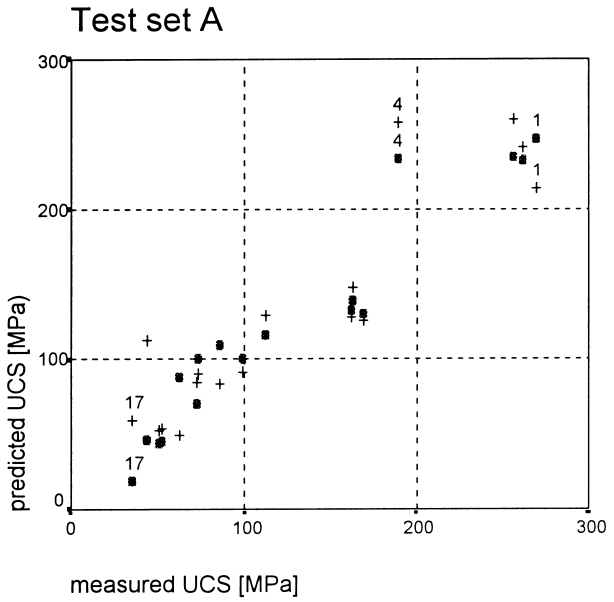

Fig. 7. Predicted UCS values vs measured UCS values for test set A, using the curve fitting relation (crosses) and the multivariate regression relation (dots).

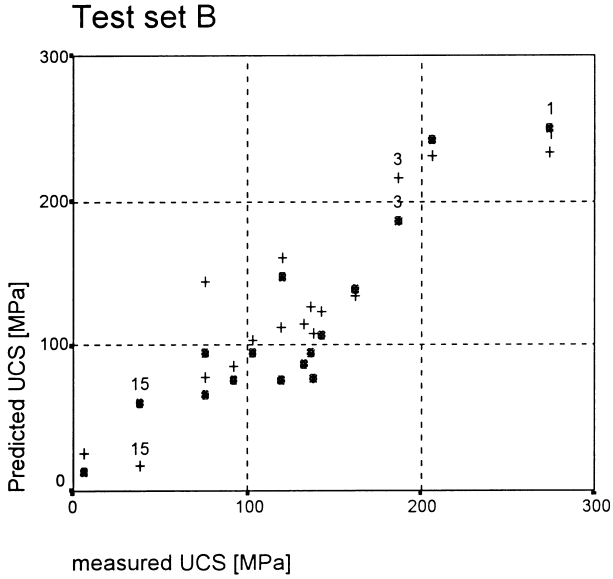

Fig. 8. Predicted UCS values vs measured UCS values for test set B, using the curve fitting relation (crosses) and the multivariate regression relation (dots).

Eq. (12). It should be noted, that the incorporated independent variables might introduce multicollinearity into the model. Although multicollinearity causes problems in interpreting the regression coefficients, it does not affect the usefulness of a regression equation for prediction of new observations [18].

3.4.2. Network UCS predictions

The UCS predictions of the network were done after the network was trained using 4 training epochs, resulting in a summed squared error (SSE) of the training of 3.92. This summed squared error might be confusing in the sense that its value seems to be high with respect to the relative small training set used. However, it should be noted that the summed squared error[6] of the training is the sum of the errors between the measured and predicted output for all training vector pairs.

It is important that the input values of the test set lie within the boundaries of the training data, because neural networks are not capable to extrapolate beyond the training data. Therefore, the minimum values provided in Table 2 serve as the boundary conditions of the input test values.

3.4.2.1. Predicting results of test set A. Fig. 9 depicts the predicting results of test set A. The correlation coefficient between the predicted UCS using the NN model and the measured UCS is 0.967. This compared with 0.957 of the predicted UCS values using multivariate regression analysis (Eq. (12)) and 0.910 of the predicted UCS values using the curve fitting relation (Eq. (11)).

From Fig. 9 and Table 3 it can be seen that a good correlation exists between the predicted UCS values; using the NN model and the measured UCS values. except the granite sample No. 4 (Table 3 and Fig. 9),

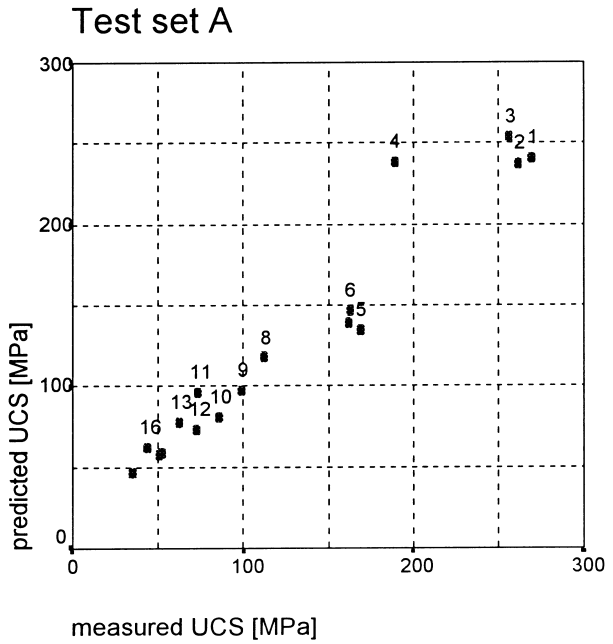

Fig. 9. Predicted UCS values vs the measured UCS values for test set A using the neural network model.

[6] This summed error should not be confused with the root mean squared (RMSE) error, which is often used to evaluate the training of a network, giving much smaller training errors.

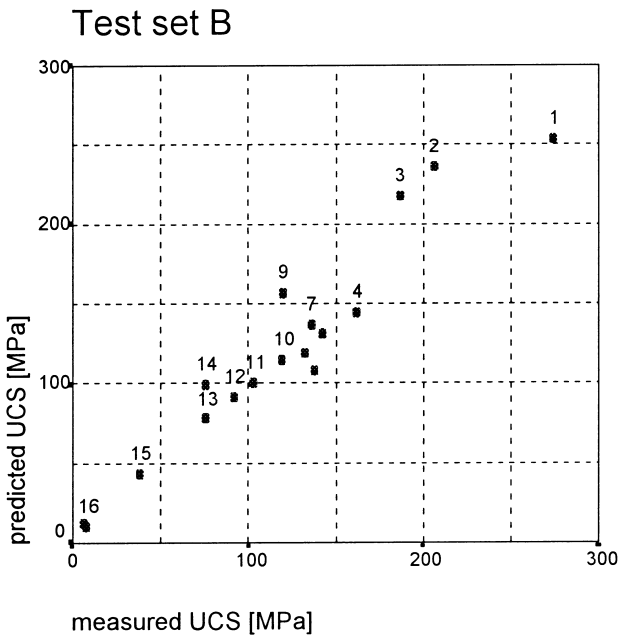

Fig. 10. Predicted UCS values vs the predicted UCS values for test set B using the nearest network model.

shows a strong overestimated UCS (238.2 MPa instead of the measured 189.1 MPa). The measured UCS of this granite rock sample is also overestimated severely using the curve fitting relation (257.9 MPa) and the multivariate equation (230.65 MPa). A possible explanation of the overestimation might be an incorrect measured Equotip value. This assumption is supported by the fact that all predicting models, especially the curve fitting relation which directly relates the UCS with the Equotip reading, do show high overestimated values. Another reason might be a bad failure during the UCS test of sample No. 4, resulting in a too low measured UCS compared to the corresponding Equotip reading.

3.4.2.2. Predicting results of test set B[7]. Fig. 10 depicts the UCS predictions of test set B. The correlation coefficient between the predicted UCS using the NN model and the measured UCS is 0.962. This compared with 0.906 of the predicted UCS values, using multivariate regression analysis (Eq. (11)) and 0.914 of the predicted UCS values, using the curve fitting relation (Eq. (12)).

Apart from rock sample No. 9, again a good correlation exists between the predicted and the measured UCS values. Rock sample No. 9 (a dolomitic limestone) shows an overestimation in the predicted UCS value (Fig. 10 and Table 4). Because also the curve fitting relation (Eq. (11)) and the multivariate regression relation (Eq. (12)) overestimate the UCS of this sample, the corresponding measured Equotip value might be considered as too high. Also the possibility of a premature failure during the UCS test can be a reason. Note also the too low predicted UCS values of limestone rock sample No. 4–8 (Table 4) using the curve fitting relation, compared with the UCS predictions, using the neural network. The predicted UCS values of these samples using the nearest network model are quite good, which indicates that other rock characteristics besides the hardness of the material (i.e. Equotip reading) influence the strength of these samples as well.

An indication that both the statistical relations and the neural network can generalize is given by the correlation coefficients of both test sets. Comparison of the correlation coefficients of the multivariate regression relation (Eq. (12)), the curve fitting relation (Eq. (11)) and the neural network, show that especially the accuracy of the UCS predictions using the multivariate regression equation is rather variable for the two test sets. For test set A, the multivariate regression relation reaches a correlation coefficient of 0.957, while the same relation reaches only 0.906 when using test set B. The correlation coefficients of the curve fitting relation and the neural network are more or less constant for both test sets. The curve fitting relation (Eq. (11)) has a correlation coefficient of 0.910 for test set A and 0.914 for test set B, while the correlation coefficients of the neural networks are 0.967 for test set A and 0.962 for test set B. This suggests that the generalization capability of both the network as well as the curve fitting relation is good. Examining the accuracy of the predictions of both the statistical methods and the neural network, shows that the latter gives the most accurate predictions in terms of the correlation coefficients.

4. Conclusions

Using neural networks to predict the UCS of rock samples does have some benefits compared to the statistical predictions. First, the UCS predictions are more accurate than the statistical predictions. Furthermore, and more important, the neural network is able to generalize much better than statistical models do. This is indicated by the consistency of the correlation coefficients for different test sets. The multivariate regression relation (Eq. (12)) shows inconsistent correlation values. This makes the applicability of neural networks for the predictions of the UCS more reliable than statistical models.

One of the major disadvantages of neural networks compared to statistical or fuzzy models is their opa-

[7] Test set B does not corresponds to the initial data set. So, it constitutes a real verification (or checking set).

queness. Therefore it is recommended to make a neural network model more transparent by verifying the established relations between the different input parameters and the output by using for example the RSE algorithm, as discussed in this paper.

Acknowledgements

The authors want to thank Professor Dr M. A. Reuter for his comments concerning neural networks. Furthermore, we want to thank Mr. Asef who provided the rock sample data necessary for this research and Dr P. N. W. Verhoef and W. Verwaal for their advice.

References

[1] Verwaal W, Mulder A. Estimating rock strength with the Equotip hardness tester: technical note.. International Journal of Rock Mechanics and Mining Sciences and Geomechanics 1993;30:659–62.
[2] Equotip. Operation Instructions, 5th ed. Proseq S.A., Zurich, Switzerland, 1977.
[3] Asef MR. Equotip as an index lestfor rock strength properties. M.Sc. thesis, ITC Delft, 1995.
[4] Hoek E, Brown ET. Underground Excavations in Rock. Institution of Mining and Metallurgy, London, 1980, pp. 527.
[5] Verhoef PNW. Wear of Rock Cutting Tools: Implications for the Site Investigation of Rock Dredging Projects. Rotterdam: Balkema, 1997, 327 pp.
[6] Leeb DH, Muller K, Wolfgang W. Harteprufung an Metallen und Kunststoffen. Ehningen, 1990.
[7] Meulenkamp F. Improving the prediction of the UCS, by Equotip readings using statistical and neural network models. Memoirs of the Centre for Engineering Geology in the Netherlands 1997;162:127.
[8] Demuth H, Beale M., Neural Nelivork ToolBox User's Guide. The Mathworks Inc., MA, 1994.
[9] Rogers SJ, Chen HC, Kopaska-Merkel DC, Fang JH. Predicting permeability from porosity using artificial neural networks. AAPG 1995;79(12):1786–91.
[10] Pandya AS, Macy RB. Pattern Recognition with Neural Networks in C + + . MEE Press, FL, 1996.
[11] Lines LR, Treitel S. A review of least-squares inversion and its application to geophysical problems. Geophysical Prospecting 1984;32:0–0.
[12] Murray AI. Applications of Neural Networks. Kluwer, Dordrecht, 1995.
[13] Aldrich C, Reuter MA, Deventer JSJ. The application of neural nets in the metallurgical industry. Minerals Engineering 1994;7:793–809.
[14] Sietsma J, Dow JF. Creating artificial neural networks that generalize. Neural Networks 1991;4:67–79.
[15] Singer DA, Kouda R. Application of a feedforward neural network in the search for Kuroko deposits in the Hokuroku district, Japan. Mathematical Geology 1996;28(8):1017–23.
[16] Yang Y, Zhang Q. A hierarchical analysis for rock engineering using artificial neural networks. Rock Mechanics and Rock Engineering 1997;30:207–22.
[17] Norusis MJ. SPSS Professional Statistics 7.5. Chigaco, 1997.
[18] Glantz SA, Slinker BK. Primer of Applied Regression and Analysis of Variance. McGraw-Hill, New York, 1992.