Modelling of the hardenability of engineering steels using artificial neural networks.

P.J. van der Wolk

part 1: Text

master thesis

August 1995

Delft University of Technology
Department of Materials Science
Heat Treatment Science and Technology Group
Rotterdamseweg 137
2628 AL Delft

Supervisors:

Prof. dr. ir. S. van der Zwaag
Ir. W. G. Vermeulen

Contents

part 1

Summary

The hardenability of steels over a wide range of steel grades has been modelled from chemical composition and austenitising temperature in a statistical way using feedforward hierarchical artificial neural networks. The effect of the architecture of the network on the accuracy in the prediction was studied in detail. Both the number input parameters (which determine the number of nodes in the input layer) and the number of nodes in the hidden layer were varied, and the optimum architecture was established. Furthermore the minimum number of sets of data required to build an accurate model was determined. Using an optimised network it was shown that the neural network is indeed capable of predicting the Jominy hardness profile over a wide range of steel compositions with a very good accuracy (2 HRc). Over large areas of the compositional domain the error in the major part of the Jominy curve is comparable to the experimental error. Also a model for a specific steel grade, boron containing steels, was constructed.

The accuracy of the network models was compared to two types of other models: models from literature and models, based on the same database as the neural network, using the Partial Least Squares multilinear regression technique. Literature models were not very accurate for the database. The partial least squares models were reasonably accurate. The neural network model clearly outperforms all other models. In fact the quality of the neural network model is such that a significant further reduction in the error of prediction requires a reduction in the experimental error in the experimental data themselves.

1 Introduction

The hardenability of a steel is one of the most important parameters controlling the final properties of heat treatable steels. The hardenability of a steel is nowadays almost universally specified produced in terms of a hardness versus distance plot resulting from a standard test: the Jominy End Quenched test. This test characterises the hardenability of a steel completely, it is robust and relatively simple.

Since the start of this century it is known that the hardenability of a steel depends on its composition and the austenite grain size only. A lot of effort has indeed been spent on calculating the hardenability from these parameters. This resulted in a large number of publications describing empirical models for the calculation of the hardenability [1, 2]. Common in all models which are based on statistical data, rather than physical models is that they are valid over a limited compositional range only. A major reason for this shortcoming is that the literature models only use simple and predefined dependencies between composition and hardness values at specified positions.

Artificial neural networks modelling, a relatively new statistical technique, may have some significant advantages over the earlier models as it does not require a priori assumptions on functional dependencies. It is therefore well capable of modelling both linear and non-linear dependencies and handles interactions between input parameters very well.

As any statistical technique a neural network requires a large data set to develop an accurate model. In a laboratory or from literature data sets of this size can rarely be obtained . However such data sets are generated in the course of steel production in a steel plant. The data set used in this work provided by Nedstaal B.V. Alblasserdam and contained over 4000 sets of data.

Purpose of this research was to develop a model for the prediction of the Jominy hardness profile from chemical composition using artificial neural networks. In this work only feedforward hierarchical neural networks were used as they are most suitable for such types of data sets. The effect of the architecture of the network was studied in detail. Both the number of nodes in the input layer and the number of nodes in the hidden layer were varied, and the optimum architecture was established. Furthermore the minimum number of sets of data required to build an accurate model was determined.
Using an optimised network it is shown that the neural network is indeed capable of predicting the Jominy hardness profile over a wide range of steel compositions with very good accuracy.

Over large areas of the compositional domain the error in the major part of the Jominy curve is comparable to the experimental error. The error in the transitional region is larger than the experimental error.

The performance of this model is compared to general models from literature and three linear regression models trained using the same data set as used for the neural network modelling. The neural network model clearly outperforms the other models. In fact the quality of the neural network model is such that a significant further reduction in the error of prediction requires a reduction in the experimental error in the experimental data themselves.

2 The hardenability of engineering steels

2.1 Introduction

To characterise the hardenability of engineering steels the Jominy End-Quench test is widely used. In section 2.2 a brief description of this test is given. Several analytical models have been presented to describe the Jominy hardness profile as a function of the chemical composition and the austenite grain size. An overview of these models from literature is given in section 2.3.

2.2 Description of the Jominy End-Quench test

The Jominy End-Quench test [3] is used as a standard method to characterise the hardenability of steels. Hardenability is a measure of the depth to which a steel will harden when quenched from its austenizing temperature. The test consists of water quenching one end of a test specimen and measuring the hardening response as a function of the distance from the quenched end. The test configuration is shown in figure 2.1a. The cylindrical test specimen (25.4 mm (1.0 inch) in diameter and 76.2mm (3,0 inch) in length) is heated to a standard austenitising temperature which depends on the chemical composition of the steel being tested. It is held at austenizing temperature for 30 min. After austenitising the specimen is removed from the oven and one end of the specimen is quenched for at least 10 minutes by a standard stream of water, while the rest of the specimen is air cooled. As a result, a cooling rate gradient develops over the length of the specimen, with the highest cooling rate at the quenched end. The cooling rate gradient results in different microstructures and thus different mechanical properties over the length of the specimen. After quenching two flat surfaces are ground along the entire length of the bar to a minimum depth of 0.38 mm. Along these flat surfaces the hardness is determined using a hardness Rockwell C test at intervals of 1/16 inch beginning at the quenched end. Hardness values at the same distance from the quenched end from different flats are averaged, resulting in a hardness vs. depth profile or Jominy curve (Figure 1).

2.3 Literature models describing the hardenability from chemical composition

Since the start of this century it is known that the hardenability of steels is determined by its chemical composition and the austenite grain size. From then on numerous attempts have been made to characterise the hardenability from its chemical composition and austenite grain size. The chemical composition of the liquid metal can be measured easily during steelmaking.

Therefore an accurate model to calculate the hardenability in an early stage of the steel production could result in a tighter control of hardenability of the final product.

To characterise the hardenability Grossmann [4] formulated a generally applicable concept: the ideal critical diameter D_i. The ideal critical diameter is defined as the largest diameter of a cylindrical specimen which transforms into at least 50 percent of martensite when quenched with an infinitely large cooling rate at the surface. This diameter can either be determined metallographically or by determining the point of the Jominy curve where the hardness value drops below the value of a steel containing 50 percent martensite.

Until recently, attempts to calculate the hardenability of a steel have been focused mainly at the calculation of its critical diameter. For this purpose first the ideal critical diameter D_{ic} for a steel with identical carbon content to the steel of interest and no alloying elements is calculated from carbon content and austenite grain size only. For commonly used steel grades the effect of all other alloying elements is accounted for by element specific factors:

$$D_i = D_{iC} \cdot f_{Si} \cdot f_{Mn} \cdot f_{Cr} \cdots \qquad (2.1)$$

where: f_{Si}, f_{Mn}, f_{Cr} = specific factors for silicon, manganese, chromium respectively.

Every factor is a function of the amount of the corresponding element present and is usually given in tabulated form [4]. The Grossmann factors are listed in appendix A.
This Grossmann model does not take interaction between elements into account. Particularly for steels containing boron, this restriction is insuperable. Therefore for these steels the following so-called boron factor, B_f, containing the carbon content, is suggested:

$$B_f = 1+1.6\cdot(1.01- \%\ C) \qquad (2.2)$$

This boron factor is then used in the same way as the other element specific factors. Note that this boron factor still only depends on the carbon concentration, implying that the amount of boron is of no influence.

A large number of researchers determined hardenability factors for many steel grades. In time, preference for certain factors has evolved. For heat treatable steels with average carbon content (0.30-0.60%) the factors of Boyd and Field [5] or Kramer, Siegel and Brooks [6] are commonly used; for case hardening steels with max. 0.25 % carbon the Renata and Doane factors [7] ; and for hypereutectiod steels and carbonated surfaces of case hardening steels the factors of Jatczak [8].

The hardenability factors of these models are restricted to a limited domain, and show inconsistencies when compared to each other or to new data.
These limitations are due to the mathematical restrictions of the chosen method as interactions between alloying elements are not taken into account and linear dependencies are used. To make this method applicable over a wider range of steel grades exponential hardenability factors were proposed by Moser and Legat [9].

Determining the critical diameter gives no direct information on the development of the hardness along the depth profile. However, methods have been developed to calculate the complete Jominy curve from the surface hardness value and the critical diameter. The surface hardness is mainly determined by the carbon content. Field[10] developed a method to calculate the Jominy hardness value at distances (DH) from the quenched end:

$$DH = \frac{IH}{DF} \qquad (2.3)$$

where: DH= distance hardness
IH = initial hardness
DF = dividing factor

The dividing factor is a function of the critical diameter and the distance from the quenched end. It is usually given in tabulated form. Since then, a number of tables of DF values has been proposed [11, 12, 13]. Due to the fixed shape of the curve when using (3), this method may be inaccurate for certain steel grades. Therefore, separate tables for heat treatable, case hardening, and boron steel grades have been proposed.

Recently, Geary et. al. [14, 15] have described a so-called Database Method. This approach overcomes the problem of the fixed shape of the curve calculated with (2.3). The Database Method uses a large database containing the composition and the measured Jominy curves for a large number of steels. From the total database some ten steels are chosen with compositions closely matching composition of the steel of which the Jominy curve must be determined. The Jominy hardness profiles of these selected steels are then modified using the compositional mismatch and factors of ASTM-255 [16], an ASTM standard for calculating hardenability. This standard is essentially a mixture of the Grossmann and the Field method.

All methods described above have in common that the hardness at the quenched end and the critical diameter is calculated first, from which the Jominy profile is determined.

Alternatively, the hardness at a certain Jominy position can be calculated directly from the composition using multiple regression techniques.
Just [17,18] constructed a comprehensive model for a wide range of steel grades. The (additive) equation in its simplest (linear) from is:

$$J_x = b_0 + b_C \cdot \%C + b_{Si} \cdot \%Si + b_{Mn} \cdot \%Mn + \ldots \qquad (2.4)$$

where: J_x = the hardness value (HRc) at Jominy position x
b_b = coefficients determined by regression analysis

Using only linear functional dependencies, these formulas will not result in an accurate model for a wide range of steels. To improve the accuracy, other than linear dependencies and element interactions must be taken into account. Just formulated his so called comprehensive model adding $\sqrt{\%C}$ and x, the distance from the quenched end, as variables, resulting in just one equation for the hardness at positions $J_4 \ldots J_{19}$, where the x is given in $1/16$ of an inch. When interactions are expected to be present, these may be accounted for by adding interaction terms to equation (2.4).

Furthermore multiplicative models can also be constructed using multiple linear regression. In this case the general equation is given by:

$$J_x = b_0 \cdot (\%C)^{b_C} \cdot (1 + \%Si)^{b_{Si}} \cdot (1 + \%Mn)^{b_{Mn}} \ldots \qquad (2.5)$$

For implementation in multiple linear regression techniques, this equation is often rewritten as:

$$\log J_x = \log b_0 + b_C \cdot \log(\%C) + b_{Si} \cdot \log(1 + \%Si) \ldots \qquad (2.6)$$

However, solving (6) using multiple linear regression will not result in finding the optimum parameters of (5) [19]. These can only be found using multiple non-linear regression analysis. The additive models are reported to out-perform the multiplicative models implemented in multiple linear regression [18]. However this may be due to the use of multiple linear regression on non-linearly transformed variables.

All multiple regression models are only valid for a limited domain. For identical steel classes significantly different factors are reported [2]. Generally, any empirical model is expected to be accurate only for the data source it is based upon. Therefore, scepsis on the

existence of uniform formulae for the hardenability of steels from different manufacturers is justified.

Except for the Database Method all these methods use pre-assumed dependencies for the effect of alloying elements on the hardenability. These dependencies are chosen heuristically; in most cases linear. Ideally for this type of problem a statistical modelling technique must be able to construct a model for any functional dependencies without any a priori assumptions on the nature of these functional dependencies.

3 Statistical modelling

3.1 Introduction

In this chapter the theory of the artificial neural network used is described.
Two general issues for all statistical models, the scaling of data and the validation of models, are described first in section 3.1.1 and 3.1.2. The neural network used for this investigation is described in section 3.2. In this work the results of artificial neural networks are compared with those of classical linear regression models. Some linear regression techniques are described in section 3.3. Non-linear regression techniques have not been used to construct models.

3.1.1 Scaling of data

Most statistical techniques require variables with mean zero and homogeneous variance [19]. When not all parameters have a homogeneous variance, a parameter with a small range is ignored when parameters with large ranges are present. Often data from production processes do not meet these requirements, so the input and output data must be transformed before applying a statistical technique. A mean value of zero is commonly achieved by mean centering: the mean value of a parameter is subtracted from all its data. When all input or output parameters are measured in the same units, this scaling may already be sufficient for statistical modelling. Parameters measured in different units need to be variance scaled: the mean centered data is divided by the standard deviation of that parameter. Instead of dividing by the standard deviation, data can also be scaled using weight factors. This last method is used when certain parameters are known to be of less importance and therefore should not influence the model too much.

It is not necessary that input and output data are scaled in the same way, as differences in scaling are absorbed in the statistical model. These types of scaling assume that the errors in input or output parameters are uncorrelated, a condition that is usually met.

Ideally, the data for a statistical model are spread evenly over the complete domain i.e. the sample density is constant. In that case, a model will have an almost uniform reliability in any part of the domain. However, in practise the sample density in the domain usually shows significant fluctuations. Trying to incorporate a few isolated samples can distort models completely. Modelling techniques less sensitive to this effect will result in models with a high prediction accuracy in regions with high sample density, and poor accuracy in regions with low sample density. In most cases data pre-treatment (e.g. exclusion of

isolated samples thus decreasing the domain, or inclusion of extra samples) is necessary to diminish the effect of sample density fluctuations.

3.1.2 Validation of a statistical model

For the validation of the predictions of a statistical model an estimate of the accuracy of the model is required. Two methods are widely used in statistical validation. The simplest method is to divide the data available into a set of fitting samples and a set of validation samples. Once a model is constructed using the fitting samples, the validation set can be used to evaluate its accuracy. The fitted model is used to predict the response of the validation samples, which is then compared with the measured response of the validation samples. This method is reliable when the validation samples are representative for the model.

When the number of data does not allow the isolation of a large number of data, cross-validation can be used. A model is constructed with all samples but one, and this last sample is used to calculate the error in prediction. Then, a new model is constructed leaving out a different sample, again calculating the error in prediction. This procedure is repeated until every sample is left out once. The errors calculated during this procedure are a measure for the accuracy of the model. This method may seem to generate very high computational cost. In case of artificial neural networks this is true; for multiple linear regression cross-validation can be achieved without repeated regression runs and is therefore relatively simple.

3.2 Artificial neural networks

3.2.1 Introduction

In situations in which the traditional statistical techniques fail to provide an accurate model - such as in the case of unknown functional dependencies or strong non-linearity - a new statistical technique, neural network modelling, may be successful. A wide variety of neural network types exist; for different application areas, different neural networks are used. In this research, a feed-forward hierarchical neural network is used. An extensive, general description of neural networks is given in [20][21][22]. In sections 3.2.2 the artificial neural network used is described; in section 3.2.3 the training procedure is explained. In section 3.2.4-6 requirements for applying neural networks are given.

3.2.2 Theory of the artificial neural network

One type of neural network is commonly used for modelling, the feed-forward, hierarchical neural network. Feed-forward, because the information is processed in one direction, from input to output; hierarchical, because the processing elements are ordered in layers. This network will only be considered here. This feed forward hierarchical neural network specifically is described in [23] (appendix B).

A feed-forward, hierarchical neural network is shown schematically in Figure 3.1. It consists of layers of nodes connected to each other with weight factors. The basic unit in a neural network is its processing element, called a node or a neuron. In hierarchical neural networks these nodes are ordered in layers. The network used throughout this investigation contains three layers: an input layer, a so-called hidden layer, and an output layer. As one hidden layer yielded sufficient accuracy, no networks with a larger number of hidden layers were used.
The number of nodes in the input layer and the output layer are determined by the number of input and output parameters respectively. The input layer contains as many nodes as the number of input parameters considered. The input layer receives a set of values for every process parameter as input, and calculates an output value from its input value. The output value of every node in the input layer is sent to all nodes in the hidden layer. This layer has a number of so-called hidden nodes depending on the complexity of the problem. This number is determined by the designer of the network. Once an output value is calculated for every hidden node, the output of every hidden node is sent to all nodes in the output layer. The number of nodes in the output layer equals the number of output parameters. Every node in the output layer calculates an output value with the output values from the nodes in the hidden layer. The output value of a node in the output layer is the prediction of an output parameter.
The nodes in neural networks perform the elementary processing operations: calculating one output value (a scalar) from the node's input. For any node in the networks considered here, all input values are cast in a so called input vector. One extra element, the offset or bias, with constant value 1 is added to this vector. Then the net input, a scalar, is determined by the inner product of the input vector and the so-called weight vector. The weight vector is a part of the node that determines the effect of each input value on the output of the node. Finally the output value of the node is calculated by passing the net input through a sigmoid transfer function.
The weight factors in all nodes can be considered as the fit parameters of the neural network model. With all weight factors of a neural network model set to the right values, it can calculate corresponding output parameters from a set of input parameters.

To determine the correct weight factors, a so called training procedure is followed. A network is trained using a large number of samples (input data with corresponding output data) and a learning rule which determines how the weights are corrected. The networks used were trained using supervised training with the momentum version of the back-propagation of error training rule. For a detailed description of network training procedures, see [22].

3.2.3 Training of an artificial neural network

The training cycle is outlined in the flowchart of Figure 3.2. To start, random values are attributed to all weight factors. Then, the input data of one sample is processed by the network and its output is compared with a target output. The target output consists of the measured values for every input parameter. The difference between the calculated and measured values - the error in prediction - is a measure for the adaptation of the weights. This adaptation takes place in the reverse direction - back propagation of error - so first, the weights of the output nodes are adapted and then the weights of the nodes in the hidden layer. This process is repeated for all samples in the database. The samples are presented to the neural network in random order. Once the weights have been adapted for all samples, one training cycle is finished. This training cycle or iteration is repeated until the differences between calculated and target output values are minimised sufficiently. Once the network is trained, the weight factors are fixed and the neural network may be used to calculate the output for any arbitrary set of input data

In the following section the data processing for the nodes in a network is described in more detail in mathematical terms. The processing of data in all layers is shown schematically in Figure 3.3.
The net input, a scalar, is determined by the inner product of the input vector and the weight vector of a node:

$$\mathbf{net}_j = \sum_{i=1}^{i_{max}} \mathbf{w}_{ji}\mathbf{o}_i \qquad (7)$$

Where: $\mathbf{i}_{max}$ = the number of nodes in the preceding layer

$\mathbf{o}_i$ = the i^{th} input value of a layer, consisting of the output elements $\mathbf{o}_{i=1} \ldots \mathbf{o}_{i_{max}}$ of the preceding layer

$\mathbf{w}_{j\,i}$ = the i^{th} value of weight vector w belonging to the j^{th} node in the considered layer

Then, the transfer function $\mathbf{f_j(net_j)}$ transforms the net input value into the output value $\mathbf{o_j}$. This value is broadcast to all nodes in the succeeding layer, except for the output nodes, where the output values of the nodes serve as output of the neural network. (Figure 3.3) However, to lead to accurate results, an neural network needs to be trained first. Many training rules exist; the one considered in this report is the delta learning rule.

In the delta learning rule the change in the weight of a node is proportional to the error signal δ_{pj} during a training step p. In mathematical form:

$$\Delta_p \mathbf{w_{ji}} = \eta\, \delta_{pj}\, \mathbf{o_{pi}} \qquad (8)$$

Where: η = the learning rate (a constant)

$\mathbf{o_{pi}}$ = the i^{th} input value of the node

$\Delta_p \mathbf{w_{ji}}$ = the change of the i^{th} component of the weight vector in the j^{th} node during training step p.

When the target vector $\mathbf{t_{pj}}$ of an output unit during training is known, the error signal for an output unit is calculated using:

$$\delta_{pj} = (\mathbf{t_{pj}} - \mathbf{o_{pj}})\, \mathbf{f_j'(net_{pj})} \qquad (9)$$

Where: $\mathbf{f'}$ = the derivative of the transfer function.

For a hidden or input unit the error signal must be determined from the error signal in the succeeding layer k, by:

$$\delta_{pj} = \mathbf{f_j'(net_{pj})} \sum_{k=1}^{k} \delta_{pk} \mathbf{w_{kj}} \qquad (10)$$

In this way weights in all layers can be adjusted recursively starting from the output units working back to the input units. This procedure is denoted as backward propagating or backpropagation. As derived in [20] this method constructs a model by a least squares approximation. This procedure can also be interpreted geometrically. For a given problem each combination of weights of an neural network for a given network architecture has a certain error. If a multidimensional weight space is constructed with one axis for each weight, and one extra axis for the degree of error, then a combination of weights can be represented by its corresponding coordinates. The error in this network of this combination of weights can be represented by the 'height' on the error axis. In this way an error surface

is created. The global minimum of this error surface corresponds with the best possible combination of weights for this problem. The aim of training is to find this optimum combination of weights.

The learning rule determines in which way the error surface is explored.
The delta learning rule minimises the error in the training set by a gradient descent procedure in the error space as long as the learning rate is small [4]. To increase the convergence speed several modifications of the delta learning rule have been proposed. A modification of the delta rule for incremental learning almost used universally is the momentum term μ ($0<\mu<1$):

$$\Delta_p w_{ji} = \eta\, \delta_{pj}\, o_{pi} + \mu\, \Delta_{p-1}\, w_{ji} \qquad (11)$$

A fraction of the previous weight change, the momentum, is added to the current weight change. In this way opposing weight changes are canceled out partially, and parallel changes are reinforced. An acceleration of convergence in the error space across long regions of constant and low gradient is achieved, and escape from certain local minima is made possible.

During training an neural network constructs an internal representation using the training data. If the right training data and network parameters are chosen, this internal representation yields a useful model. However, the choice of network parameters and training data is not straightforward. This choice is elucidated in the next sections.

3.2.4 Input parameters

One very important aspect of implementing neural networks is the selection of input parameters. All relevant input parameters must be represented in the input data of the neural network. Usually this choice is based on the physical background of a process. As the number of training samples needed and the training time increase at least exponentially with increasing number of input parameters, the number of actual input parameters should be restricted. Therefore, preferably a small subset of input parameters must be chosen. To diminish the number of input parameters, linearly dependent input parameters can be ignored, as they carry no extra information.

Alternatively, all input parameters can be transformed into a small number of new parameters using Principal Component Analysis (PCA) with minimum loss of variance.

3.2.5 Training data

As stated earlier, a large number of training data is needed to train a neural network. Up to now it is not possible to estimate the number of training data needed for a given problem in a statistically sound quantitative way. This number depends upon the number of input parameters and the complexity of the problem. When the number of orthogonally independent input parameters is denoted n, a rough indication is: $2^n <$ number of training data needed $< 3^n$.

3.2.6 Overtraining

As the complexity of a problem and therefore the number of hidden nodes needed is not known in advance, usually more hidden nodes than strictly needed to fit the underlying model are used in an neural network. This network then has more degrees of freedom than necessary for a model.
At large numbers of training cycles the network uses these degrees of freedom to model not only the functional dependencies between input and output parameters but also the noise in the data set. This is called overtraining. To prevent the network from overtraining a validation set is used. The weight factors in the network are corrected using the data in the training set only. The validation set evaluates the performance of the network after every training cycle. In case of overtraining the error for the training set decreases while that for the test set increases with further training (figure 3.4). At this point training is stopped.

3.3 Multiple linear regression analysis

In this paragraph multiple linear regression methods are explained. The fundamentals of multiple linear regression are given in section 3.3.1. In section 3.3.2 the Principal Component Analysis (PCA) method is described which can be used to determine the most relevant parameters (or sets of parameters) in a data set. The Principal Component Regression (PCR) and the Partial Least Squares (PLS) method are described in section 3.3.3 and 3.3.4. Both methods yield models linking input data to output data assuming linear dependencies.

3.3.1 Multiple linear regression

Multiple linear regression methods are statistical techniques based on the first order relation:

$$y = b_1x_1 + b_2x_2 + \ldots + b_mx_m + e = \sum_{j=1}^{m} x_j b_j + e \qquad (12a)$$

where y = the dependent (output) variable
x_j = the independent (input) variables (j = 1..m)
b_j = the sensitivities
e = the error or residual

This relation can alternatively be written as:

$$y = \underline{\mathbf{x}}^T \underline{\mathbf{b}} + e \qquad (12b)$$

where $\underline{\mathbf{x}}^T$ = a row vector consisting of the input variables (^{T} = transpose)
$\underline{\mathbf{b}}$ = a column vector consisting of the sensitivities

This relation applies for situations with one dependent variable and one sample. For n samples (12^b) becomes:

$$\underline{\mathbf{y}} = \mathbf{X}\underline{\mathbf{b}} + \underline{\mathbf{e}} \qquad (13a)$$

where $\underline{\mathbf{y}}$ = the vector containing n output values
$\mathbf{X}$ = (n×m) matrix containing independent variables (j = 1..m)
$\underline{\mathbf{e}}$ = vector of residuals

With the number of samples n and the number of independent variables m, three cases can be distinguished. When the number of samples equals the number of variables (m=n), one unique solution can always be found resulting in a perfect match of all samples: $\underline{\mathbf{e}} = \underline{\mathbf{o}}$. With less samples than parameters (m>n) an infinite number of solutions with $\underline{\mathbf{e}} = \underline{\mathbf{o}}$ can be found. With more samples than parameters, normally no exact solution is found. In this case an optimum solution is found by minimising the length of the residuals vector $\underline{\mathbf{e}}$.
By rewriting (13) to:

$$\underline{\mathbf{e}} = \underline{\mathbf{y}} - \mathbf{X}\underline{\mathbf{b}} \qquad (13b)$$

the vector of residuals can be minimised using the least squares solution:

$$\mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\bar{\mathbf{y}} \qquad (14)$$

These equations are all valid for one dependent variable only. Every dependent variable generates one set of equations. In case of p dependent variables these equations can also be arranged in one equation:

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E} \qquad (15)$$

where $\mathbf{Y}$ = (n×p) matrix with dependent variables as columns
$\mathbf{B}$ = (p×m) matrix with sensitivities as columns
$\mathbf{E}$ = (n×p) matrix with residuals as columns

The general case of (15) is the starting point for the next sections.

3.3.2 Principal Component Analysis

Any matrix can be decomposed into a sum of r matrices of rank 1. These matrices in turn can be written as the outer product of two vectors: a score vector and a loading vector:

$$\mathbf{X} = \mathbf{M}_1 + \mathbf{M}_2 + \ldots + \mathbf{M}_r = \bar{\mathbf{t}}_1 \bar{\mathbf{p}}_1^T + \bar{\mathbf{t}}_2 \bar{\mathbf{p}}_2^T + \ldots + \bar{\mathbf{t}}_r \bar{\mathbf{p}}_r^T \qquad (16a)$$

where $\mathbf{M}_h$ = a (nxm) rank 1 matrix

$\bar{\mathbf{t}}_h$ = a score vector

$\bar{\mathbf{p}}_h^T$ = the transpose of a loading vector

In matrix notation (16a) becomes:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T \qquad (16b)$$

where $\mathbf{T}$ = a (nxr) matrix with $\mathbf{t}_1 \ldots \mathbf{t}_r$ as columns

$\mathbf{P}^T$ = a (rxm) matrix with $\bar{\mathbf{p}}_1^T \ldots \bar{\mathbf{p}}_m^T$ as rows

The $\mathbf{M}_h$ matrices are arranged in order of decreasing eigenvalues. A $\mathbf{M}_h$ matrix with a large eigenvalue contains a large fraction of the variance of the data: so the first few matrices in the decomposition contain most of the variance in the data. Usually the matrices with small eigenvalues are ignored.

Many ways exist to determine this decomposition. One method, the Non-lInear PArtial Least Squares (NIPALS) [24] algorithm is discussed in more detail here, as it is necessary for understanding multiple regression techniques.

The NIPALS algorithm calculates the first principal component of a data matrix. First, as an initial guess for $\bar{t}_1$ some data vector $\bar{x}_j$ is chosen.
Then, $\bar{t}_1$ is adjusted by an iterative procedure:

(1) calculate $$\bar{p}_1^T = \frac{\bar{t}_1^T \mathbf{X}}{\bar{t}_1^T \cdot \bar{t}_1} \qquad (17)$$

(2) normalise $\bar{p}_1$ to length 1:

$$\bar{p}_{new} = \frac{\bar{p}_{old}}{\|\bar{p}_{old}\|} \qquad (18)$$

(3) calculate $$\bar{t}_1 = \frac{\mathbf{X}\bar{p}_1}{\bar{p}_1^T \cdot \bar{p}_1} \qquad (19)$$

This procedure is repeated until $\bar{t}_1$ is constant with further iteration, i.e. $\bar{t}_1$ in steps (1) and (3) are sufficiently identical. With the $\bar{t}_1$ and $\bar{p}_1$ calculated in the last iteration $\mathbf{M}_1$ is determined. Next $\bar{t}_2$ and $\bar{p}_2$ can be calculated with this procedure by replacing $\mathbf{X}$ with $(\mathbf{X}-\mathbf{M}_1)$.
Once the decomposition of the data matrix completed, the data can be represented by the scores and loadings vectors, where the first few components contain most of the variance in the data.

3.3.3 Principal Component Regression (PCR)

The data matrix $\mathbf{X}$ can be decomposed using (16b). First, the number of principal components (a) is chosen, where matrices with small eigenvalues are ignored. Then the data matrix is represented by the matrix $\mathbf{T}$ using:

$$\mathbf{T} = \mathbf{XP} \qquad (16c)$$

where $\mathbf{T}$ = a (nxa) matrix with $\mathbf{t}_1 \ldots \mathbf{t}_a$ as columns
$\mathbf{P}$ = a (mxa) matrix with $\mathbf{p}_1 \ldots \mathbf{p}_a$ as columns

resulting in:

$$\mathbf{Y} = \mathbf{TB+E} \qquad (20)$$

This equation is used in Principal Component Regression (PCR). By using $\mathbf{T}$ instead of $\mathbf{X}$ the data is represented by a small number of orthogonally independent variables.

As the score vectors in **T** are orthogonal, **B** can easily be solved. Noise usually only affects the components with a small eigenvalue. As these are excluded, noise in the data is reduced.

3.3.4 Partial Least Squares (PLS)

Partial least squares [24, 25] is a multivariate linear regression technique which uses the principal components of both the independent variables (15) and the dependent variables:

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^T + \mathbf{F} \qquad (21)$$

Where: $\mathbf{U}$ = (nxa) score vector matrix
$\mathbf{Q}^T$= (axp) loading vector matrix
$\mathbf{F}$ = (nxp) matrix with residuals

Equations 15 and 19 are called the outer relations. With the so-called Partial Least Squares algorithm a relationship between **U** and **T** , the inner relation, is calculated. This inner relation for component h is:

$$\bar{\mathbf{u}}_h^* = b_h \mathbf{t}_h \qquad (22)$$

Where: b_h = the regression coefficient.
For PLS these regression coefficients are calculated by a procedure similar to NIPALS. The **X** and **Y** matrices are decomposed simultaneously by the NIPALS algorithm: The scores vectors of both decompositions are exchanged during the decomposition resulting in a better inner relation. Furthermore weight vectors are used to keep the scores vectors of the **X** matrix orthogonal. This procedure is given in appendix C. Similar to PCA, usually not all components are used for regression, as the smaller components merely contain noise and can cause collinearity problems. To determine the number of relevant components a criterion is adopted. Using only the training data some low threshold value for $\|\mathbf{F}\|$ can be chosen. When a separate set of data is used for validation, the optimum number of components can be determined by minimising the sum of squares of the prediction of this validation set.

While the PLS method is certainly a very powerful technique to analyse large data sets, its value is reduced by the necessity of a priori specifying functional dependencies.

4 Experimental

4.1 Introduction

As stated in the introduction, the main objective of this research lies in the prediction of the hardenability from the chemical composition using artificial neural networks. Neural network models, as any model based on statistical techniques, are very sensitive to the data upon which they are trained. Therefore extra attention must be paid to the handling of the data. The data handling from the original Nedstaal data to the dataset used to train the neural networks is given in section 4.2.
Due to computational restrictions it is not possible to train a network with more than 11 input parameters. So, some parameters in the original database were excluded from the model. The selection of input parameters for the different networks is described in section 4.3. The procedure of data scaling and the choice of network parameters is described in section 4.4. In this section the evaluation of the network's prediction during training (needed to prevent overtraining) is also described. The method of evaluating the network models is given in section 4.5.
In section 4.6 the prediction of the model versus those of analytical models (Just and SteCal) and those of PLS models using the same data set is presented.

4.2 Data handling

The original Nedstaal database consists of some 4000 data sets, each consisting of the measured austenitising temperature and chemical composition as input data, and the Jominy hardness profile of a steel as output data. An excerpt of the original database is given in Table 4.1. The database consisted of low alloy steels: semi mild steels (case hardening steels), carbon steels (heat treatment steels), tool steels (spring steels) and specialty steels (ball bearing steels). The domain and range of the original database is given in Table 4.2; the distribution of the original data in Figure 4.1a.
The hardness values were determined using the standard Rockwell C hardness test. The hardness at every position is determined at both sides of the test specimen. The two hardness values are averaged and rounded to a whole number. The expected error in hardness measurement is 1 HRc.

4.2.1 Input data

The chemical composition of a steel is characterised by the weight percentages of 17 elements: C, Mn, P, S, Si, Cu, Cr, Ni, Sn, Mo, Ti, As, B, V, Al, and N. The

concentrations of these elements were determined spectrometrically. The austenitising temperature was set on basis of the steel composition.
The input data of the original database, the austenitising temperature and chemical composition of the steels, are by no means evenly spread over the domain, as shown in Figure 4.1 which gives the number density of samples per domain (the full size of each input parameter divided into 10 equal domains). As the size of the domain is determined by the highest and the lowest value of one parameter, just one sample with an anomalously high value already causes an enormous distortion in the distribution. By reducing the size of the domain the distribution can be improved. A clear example for this effect is the distribution of Cr (Figure 4.2). Just a few samples cause all other samples to fall in the categories 'low'. After excluding these samples from the database, the samples are distributed much more evenly.
Excluding some 75 samples from the database reduces the domain significantly, and the data is spread more evenly as can be seen in Figure 4.1. The database obtained after this operation was used to train the first neural network.

4.2.2 Training using individual hardness values

The Rockwell C hardness values are given for positions J_1 to J_{19}. These positions refer to the distance from the quenched end as shown in Figure 2.1. The distances at which the hardness is measured on the test specimen differs slightly from those in the standardised Jominy End Quench test [16].
As can be seen in Table 4.1 for some samples the hardness value at every specified distance was not measured. This is because for some steels the core hardenability response is not of interest; these hardness values had not been measured. For some steels the hardness at some Jominy position only needs to be below a certain value; beyond this position the hardness values were also not measured. The hardness values at these positions are denoted as '.' in the original data. For this investigation the values '.' in a profile were replaced by the known measured hardness value of the highest J-value. This crude estimate is expected to introduce a significant error for the modified hardness profiles.
Also some hardness values lower than 15 HRc were measured. Measurements of these low hardness values are unreliable. Data sets containing hardness values lower than 15 HRc are therefore excluded from the data.
The data resulting from this operation were used to train networks predicting the hardness at every Jominy position, i.e. with 19 output parameters.

4.2.3 Training using hardness profiles

Networks trained using the data as described in the section 4.2.2 model the hardness values for every position independently. By fitting a predefined function to the hardness profile coherence between hardness values at adjacent positions can be attained. The predefined function chosen for this purpose is:

$$J_x = a_1\left(\frac{1}{2} + \frac{1}{\pi} \cdot \arctan(a_3(a_2 - x))\right) + a_4 \qquad (4.1)$$

where: $a_{1..4}$ = fit parameters
J_x = hardness value at position x
x = position on the test specimen.

Parameters a_1 to a_4 determine the exact shape of this sigmoidal function as shown schematically figure 4.3.

After determining the fit parameters $a_1..a_4$ for all samples in the database, a network was trained using these four parameters as output data.

4.2.4 Training and validation set size

As a rule of thumb 75% of the data available was used for the training set, and the remaining 25% was used as a validation set. Samples were distributed randomly over the training and validation set.
To determine the minimum number of training data needed to acquire an accurate model, the number of training samples was varied. When networks were trained with a small training set, the rule of thumb was not followed, as it would result in a very small validation set. In these cases the validation may be unreliable, and therefore the size of the validation set was chosen much larger.

4.3 Choice of input parameters

The original database contained 17 input parameters. To train a network with all 17 input parameters would be very time consuming, because the training time increases more than exponential with the number of input parameters[26]. Obviously, parameters without influence on the hardenability can be ignored as input to the neural network. Correlated parameters carry no extra information, so they can be ignored as well. For a first network the austenitising temperature and the ten elements expected to be most relevant were

selected. Based on the results of this network, superfluous and irrelevant input parameters could be identified and excluded from the database. In this way other elements expected to be of influence were included in databases used to train new neural networks.

One input parameter known to have a significant effect on the hardenability, the austenite grain size, is not present in the database. However, according to Nedstaal the austenite grain size was always small (ASTM 9) [29], and correlated to the composition of the steel.

4.4 Network parameters

The austenitising temperature and alloying element concentrations all have different ranges. As described in section 3.2.1 the data needs to be scaled. Therefore the input data were scaled between -1 and 1 using:

$$x_{scaled} = 2 \cdot \left(\frac{x_{meas} - x_{min}}{x_{max} - x_{min}} \right) - 1 \qquad (4.2)$$

where: x_{scaled} = the scaled value of a certain parameter for an input sample
x_{meas} = the measured value for this parameter for this sample
x_{min} = the lowest value in the database for this input parameter
x_{max} = the highest value in the database for this input parameter

The output data were scaled between 0.3 and 0.7 using:

$$y_{scaled} = \left(\frac{y_{meas} - y_{min}}{y_{max} - y_{min}} \right) \cdot (0.7 - 0.3) + 0.3 \qquad (4.3)$$

where: y_{scaled} = the scaled value of a certain parameter for an output sample
y_{meas} = the measured value for this parameter for this sample
y_{min} = the lowest value in the database for this output parameter
y_{max} = the highest value in the database for this output parameter

By scaling the output between 0.3 and 0.7 the range of the output data falls within the part of the sigmoid transfer function of the output node that approximates a straight line.

During training the overall error in the training set and the validation set were evaluated separately after every iteration. These overall errors were calculated with:

$$\text{overal error} = \sqrt{\frac{\sum_{1}^{\text{nsamples}} \left\{ \sum_{1}^{\text{noutputs}} (y_{target} - y_{pred})^2 \right\}}{\text{nsamples} - 1}} \qquad (4.4)$$

where: y_{target} = the given output value for one parameter of a sample

y_{pred} = the calculated output value for one parameter of a sample

nsamples= the number of samples in the training set or validation set

noutputs = the number of output parameters

When the overall error in the validation set showed a minimum, the current weights of the network were saved. When the number of iterations is sufficiently large, the network with the lowest overall error for validation is found. In this way overtraining is not avoided, but it does not have influence on the final network's performance, since the weights were saved just before the onset of overtraining.

For every training the number of hidden nodes in the network must be chosen. A general rule for the optimum number of hidden nodes for a given problem does not exist. For the prediction of Jominy curves initially 15 hidden nodes were chosen. The effect of the number of hidden nodes has been investigated by training several networks on the same dataset with different numbers of hidden nodes and comparing the errors in prediction of the different networks. In this way the minimum number of hidden nodes for this problem was determined.

4.5 Error analysis

The error in training set and validation set is determined per Jominy position. For one position the error of prediction is calculated via:

$$RSD = \sqrt{\frac{\sum_{1}^{\text{nsamples}} (y_{target} - y_{pred})^2}{\text{nsamples} - 1}} \cdot \frac{1}{0.7 - 0.3} \qquad (4.5)$$

where: RSD = residual standard deviation.

The second factor of the equation equals one over the scaling range of the output data. This RSD is calculated for both training set and validation set separately for all Jominy positions. For networks predicting the hardness values at every position separately, this formula calculates the RSD for hardness values; for networks predicting hardness profiles it

calculates the RSD in fit parameters. To ensure compatibility between the RSD calculated for both types of networks, the fit parameters predicted were used to calculate hardness values at every position, after which the RSD for the latter networks was calculated via (4.5).

4.6 Other models

Two literature models, the SteCal [27] model and models by Just [17], were used to predict Jominy curves. It should be noted that both models are valid for a much smaller range of steels than the current data set (Table 4.3). For the Just models all steels with compositions within the domain of this model were selected. The Jominy hardness values calculated with the Just models were compared to the measured ones. As the distance scale of the Nedstaal data and the Just model were different, the hardness values of the Just model were modified using linear regression. The error for Justs models were calculated using (4.5) with an adapted scaling range.
The SteCal model was evaluated similarly, using 45 steel samples. This model is based on both the Grossmann and the Field method. The factors used are taken from the work of Crafts and Lamont [28]. The domain for this model is given in Table 4.3. Boron is specified to be either present or not present in a steel. For the evaluation of the SteCal model no boron steels were chosen.
For this model the austenite grain size needs to be specified. For Al-killed steels (97% of the data) this grain size is estimated to be 9 on the ASTM standard scale[29]. Only these steels were used to evaluate the SteCal model. The extra error introduced by this assumption is evaluated by calculating the influence of a deviation in austenite grain size on the prediction of the SteCal model. Both of these models are based on data from many steel mills. For this reason these models are unlikely to predict Jominy curves for the Nedstaal data more accurately than the neural network, which is based entirely on this data.

Therefore the neural network model, essentially the result of a non-linear regression technique, is compared with other statistical models based on the same Nedstaal data using a classical multilinear regression technique. These models were all constructed using Partial Least Squares (PLS), a robust general purpose multilinear regression technique. First a model was constructed with the concentration of alloying elements as input variables. Then some models were generated where functional dependencies reported in literature were added to the data as extra input parameters.
For PLS models the number of components taken into account needs to be specified. The optimum number of components was determined by minimising the error in the validation set.

5 Summary of networks trained

The present work resulted in basically four different neural network models:

1) A neural network model for a wide range of steel grades: Network 1.
2) A network for a wide range of steel grades without high carbon steels and different input parameters: Network 2.
3) A neural network for a specific steel grade of boron containing steels: the Boron Network.
4) A neural network predicting hardness profiles: the Hardness Profiles Network.

In Table 5.1 the architecture and training procedure of these networks are described. The choice of these networks is elucidated in this chapter.

5.1 Network 1

The choice of parameters of Network 1 has been made in a straightforward way. Some 17 candidate input parameters were present in the original database: 16 alloying elements: C, Mn, P, S, Si, Cu, Cr, Ni, Mo, Al, Sn, Ti, B, V, N, As, and the austenitising temperature. For all steel samples the content of As, Ti, Sn, and V was negligible; therefore these were not used as input parameters. In order not to exceed computational limits, two extra input parameters B and N were excluded. These two alloying elements were chosen arbitrarily. After the data handling as described in section 4.2.1, 3923 samples remained. These were used to train Network 1.
To determine the optimum architecture, three networks were trained with 6, 10 and 15 nodes in the hidden layer. Up to 15 hidden nodes the accuracy of the network increased. For all steel grades the effect of every input parameter on the hardenability was determined. Four parameters Cu, P, S, and the austenitising temperature appeared to have no significant effect on the hardenability. Therefore these parameters were not used for Network 2, leaving room for B and N.
For high carbon steels ($C > 0.65\%$) the prediction of this network was accurate for one steel class only. The 50 steel samples with high carbon content were excluded from networks following.

5.2 Network 2

Network 2 was trained with nine alloying elements as input parameters: C, Mn, Si, Cr, Ni, Mo, Al, N, and B. As for Network 1 the network with 15 nodes in the hidden layer turned out to result in the network with highest accuracy, this number was chosen for Network 2. Furthermore an accurate model of the hardenability was constructed using some 2900 data sets for training. With this network the minimum number of training data sets needed for an accurate model was determined by training identical networks with varying numbers of training data.

5.3 Boron network

For one specific steel class a separate network was trained. For this network boron steels were chosen, as boron has a special place among the steel alloying elements. Even at low concentrations (10 ppm) it has a extremely high effect on the hardenability. As was already recognised by Grossmann [2], the effectiveness of boron depends on the carbon content. To study these effects in detail, the boron network was constructed. All 1100 steels containing boron were selected for training from the database.
The number of hidden nodes needed for this network was determined in a similar way as for Network 2.

5.4 Hardness profile network

For the steel grades in the database the Jominy curve is known to be descending continuously. For some steel grades Network 2 showed a small increase in hardness at the Jominy positions before and after the steep part of the Jominy curve. As no secondary hardening is expected to occur in these steel grades, this effect is undesirable, although it is also found in the data (Table 4.1 row 39). This effect considered to be due to measurement errors.
By modelling the Jominy curve with (4.1), a continuous descent can be imposed on the Jominy curve, and the measurement error can be averaged. Therefore a network was trained using the parameters a_1 .. a_4 as output parameters.

6 Results

6.1 introduction

The results obtained during this research are pesented in this chapter. First those of the neural network are given in the same order as outlined in chapter 5. Then the results of the Just model, the SteCal model and the PLS models are described.

6.2 Network 1

The distribution of the training data for this neural network is given in Figure 6.1. The range of alloying elements was already given in Table 4.2. The error in prediction for both training and validation set for a network with 15 hidden nodes is given in Figure 6.2. The influence of the number of hidden nodes was investigated and the error in prediction for three neural networks with 6, 10 and 15 hidden nodes respectively is given graphically in Figure 6.3 and numerically in Table 6.1. This error is determined with the validation set. For the neural network with 15 hidden nodes the measured hardness value vs. the calculated hardness value for all steels in the validation set for the Jominy positions with the lowest and highest RSD, J_1 and J_5, are given in Figure 6.4. The composition of these steels is given in Table 6.2.

Four input parameters, P, S, Cu, and the austenitising temperature showed no effect on the hardenability, as is visualised for Cu in an arbitrary steel grade in Figure 6.6, showing nine Jominy curves with varying Cu concentration. For the austenitising temperature this effect is due to its dependence on the carbon concentation, as already stated in section 2.2. The correlation between the austenitising temperature and the carbon concentration is shown in Figure 6.7.
Finally some examples of the prediction of three high carbon steel grades is given in Figure 6.8.

6.3 Network 2

The distribution of the training data is given in Figure 6.9; the domain is given in Table 6.3. The error in prediction of both training and validation set for this new network with 15 hidden nodes is given in Figure 6.10. The predicted Jominy hardness profiles are compared to the measured ones in Figure 6.11 for identical steel compositions as the steel samples selected for Network 1.

The influence of the number of training data on the accuracy of the network model is given in Figure 6.12; The RSD at every Jominy position of a validation set with 667 samples is given for networks with 200, 400, 800, 1200 and 2000 training data. Also the error of randomly selected validation sets of 25 % of the training data is given for the networks trained with 200 and 400 data. In Table 6.2 of some outliers in the training data of Network 2 are given.

6.4 Boron network

The range of the boron steels in the original database is given in Table 6.3; the distribution of the data is given in Figure 6.13. Again, the number of hidden nodes is investigated for the Boron network. For networks with 1, 3, 5, 7, 13 and 15 hidden nodes the RSD at every Jominy position is given in Figure 6.14.
For two steel grades the prediction of the Jominy curve by Network 2 and the Boron Network is given in Figure 6.15. The effect of boron on the hardenability of these steels is shown in Figure 6.16, where the hardness at two Jominy positions is given as a function of the boron concentration.

6.5 Hardness profile network

In order to train the Hardness profile network the parameters a1 .. a4 need to be determined first for all Jominy curves. These parameters were determined using a Gauss-Newton technique [19]. For many Jominy hardness profiles this method did not converge to the optimum parameters or did not converge at all. The calculated curves were compared to the measured ones. Steel samples for which the curve as calculated with (4.1) matching the measured one (RSD < 1 HRc) were selected for this network. Some 1298 data remained. However, the network trained with these data resulted in an unacceptably high RSD (19% of the range) and showed overtraining effects after 1500 iterations.

6.6 Just and SteCal model

The error in prediction for the SteCal model, Just's comprehensive model and Just's model for hardenable steels and case hardening steels are given in Table 6.5 and Figure 6.17.

6.7 PLS

The data used to develop three PLS models was identical to the data used for Network 2. In Table 6.6 the input and output parameters and the number of components taken into account are given. The error in prediction for the three PLS models and of Network 2 are given in Figure 6.18 and table 6.7.

7 Discussion

7.1 Introduction

In this chapter the results of the neural networks presented in chapter 6 are discussed in the order outlined in chapter 5. Then the resultst of the the literature models and the PLS models are compared with those of the neural network models.

7.2 Neural network models

7.2.1 Network 1

The overall accuracy of this network with 15 hidden nodes was high (figure 6.2), with an average error in prediction in the same order as the experimental error ($\pm 1\,1/2$ HRc). The prediction of the hardness at the first two Jominy positions and at positions far away from the quenched end are more accurate than the predictions for positions 3 to 8. This can be explained as follows. At the surface of the quenched end the hardness is almost entirely determined by the carbon content. Far from the quenched end the cooling rate shows little variation. In these two regions the hardness varies little with position. However, in the intermediate zone the hardness measurement is very sensitive to the exact location at which the hardness is measured as the hardness depends strongly on the position. At these positions a small error in the distance at which the hardness is measured results in a measured hardness value that strongly deviates from the hardness value at the point one thinks one is measuring. The intermediate zone gradually blends into the zone far from the quenched end. This is due to the different positions at which the intermediate zone is located for different steel grades. This effect affects all hardenability models.
As a representative validation set is used and the error in the training set matches the error in the validation set, overtraining of the network has been avoided. However the error in the validation set is slightly lower than the error in the training set at Jominy distances further than 13 mm from the quenched end. This is possibly due to a relatively large number of samples with large errors at these Jominy distances.

After training Network 1 the influence of the alloying elements and the austenitising temperature on the hardenability was evaluated. Four input parameters turned out to be superfluous for modelling the hardenability: P, S, Cu and the austenitising temperature. For all steel grades these four parameters were varied without showing significant changes in the Jominy hardness profile, as is illustrated for one (arbitrary) steel grade for Cu in Figure 6.6. For the austenitising temperature this was to be expected as it is directly correlated to the carbon content of the steel and offers therefore no extra information (Figure 6.7). P and

S are only present as contamination for the steel grades in the data and are therefore always kept to a sufficiently low level. For the amounts present Cu appeared of no influence on the hardenability.

For steels with high carbon content (C > 0.65 %) however, the prediction of the hardenability seems inaccurate for many steel grades. This behaviour is probably due to the low number of steels with this high carbon content: only some 50 steels in a data set of 3800 (1.25 %). These steels consisted mainly (90%) of one steel grade: 102Cr6. Obviously there is a large gap in the domain of the carbon content. As can be seen in Figure 6.8a, Network 1 predicts the hardenability of 102Cr6 accurately. For two other steel grades, C80 and 70SiMoV5, the prediction shows large errors. This is due to the small number of samples of these steels in the database used for training (2 and 0 respectively). Therefore it is likely that all high carbon steels but 102Cr6 were considered as noise by the network.
As the carbon content can be chosen independently from other alloying elements, Network 1 suggests to be able to make a prediction for any high carbon steel. When a steel other than 102Cr6 is chosen, an extrapolation is made causing an unreliable prediction. Therefore high carbon steels were excluded from subsequent networks.

7.2.2 Influence of the number of hidden nodes of Network 1

The number of hidden nodes needed to develop an accurate model depends on the complexity of the problem. The accuracy of a model is expected to improve when the number of hidden nodes is increased, until an optimum number of hidden nodes is reached. Using more nodes than the optimum number does not improve the accuracy in prediction of the network. More hidden nodes increase the training time and the risk of overtraining. For Network 1 this number of nodes in the hidden layer needed to model the hardness at 19 Jominy positions was determined. As can be seen in Figure 6.3, the accuracy of the network improves with increasing number of nodes. A network trained with 20 nodes in the hidden layer showed an error in prediction closely matching a network with 15 hidden nodes. For this problem 15 nodes in the hidden layer is sufficient and chosen as the optimum number.

Network 1 with 15 nodes in the hidden layer predicts the hardenability of steels with an accuracy close to the experimental error. It shows two shortcomings: the hardenability of an arbitrary high carbon steel cannot be predicted accurately, and two important input parameters, boron and nitrogen, are not incorporated in the model.

7.2.3 Network 2

For Network 2 the alloying elements B and N were added as input parameters as these are expected to influence the hardenability significantly. This network performed even slightly better than Network 1 (Figure 6.10). The network's performance has improved for two reasons:

1) by excluding the high carbon steels a source of noise is removed
2) two relevant input parameters were added in Network 2

Moreover, by removing superfluous input parameters the network has become more compact, thus avoiding superfluous weight factors.

A number of outliers for Network 2 are collected in Table 6.4. Some outliers (sample 6) are clearly due to measurement errors. This can be seen by comparing these Jominy profiles with those of steels with closely matching compositions.
Outlier samples 4, 5, 7 and 8 are caused by the effect that a small error in position results in a high deviation of the hardness value in the intermediate zone as described in section 7.2.1.
One category of outliers (samples 1 and 3) is related to sets of input data for which the hardness values were not measured at all Jominy positions. These outliers are the result of our assumption in data handling that the hardness at these positions were equal to the hardness value measured at the highest Jominy position.
Sample 9 is labelled a 38Mo3 steel, but the Cr content is far too high (1.26) for this steel class. This error may be due to a sample mix-up. Sample 2 is labelled a 36CrMo4 steel, but its Mo content is 0, whereas it is usually some 0.25 % for this steel class. Besides, the Sn content on this steel in the adjacent data cell is 0.27, where it is expected to be 0.01, indicating an administrative error.

7.2.4 Influence of the number of data

The number of data necessary depends on complexity of the problem: the number of input parameters and the nature of the non-linearities and interactions present. The number of training data needed is usually estimated to be between 2^n and 3^n, where n is the number of input parameters. The number of training data needed was estimated by training networks with different numbers of training data. As expected, the error in prediction of these networks in the training set decreased with decreasing number of training data. This was due to overtraining: a network trained with a small number of training data can fit noise in the data more easily.

For a reliable validation of a model using a validation set, this validation set needs to be representative for the model in every part of the domain. For a sufficiently large validation set this condition is usually met. Shown in Figure 6.12a is the error in a fixed validation set of 969 data for these networks. From this figure it is clear that at least some 800 data are needed for modelling the hardenability for these steels with sufficient accuracy.
When the size of the validation set was determined by the rule of thumb mentioned in section 4.2.4, this condition was violated, as is shown in Figure 6.12b. As for the error in the training set the error in the validation set decreased with decreasing number of training data. The error in the validation set was unrealistically low for small numbers of samples. Comparison of identical models with validation sets of different sizes showed that even a validation set of 400 samples may result in some (small) deviation in the estimation of the accuracy of a model. Therefore for these models the validation set must at least contain 400 samples. Smaller randomly chosen validation sets do not cover the input space in a representative manner.

7.2.5 Boron network

The accuracy of the Boron network is high (Figure 6.14); the error in prediction is of the same order as the experimental error. The network with 5 hidden nodes will be considered here, for reasons that will be explained in 7.2.6. The shape of the error vs. Jominy distance plot resembles the one of Network 2 (Figure 6.10). These plots are different in two ways:

1) For the Boron network the RSD far from the quenched end is lower.
2) For the Boron network the region with a high RSD (the intermediate zone) is more distinct.

The first difference can be explained by the range of both networks. Network 2 has a lager range at high Jominy positions than the Boron network (45 HRc and 12 HRc respectively at a Jominy distance of 70 mm).
The second difference is caused by the shape of the Jominy curve of boron steels specifically. As can be seen in Figure 6.15 the hardness of boron steels in the intermediate zone drops steeply at one Jominy position, whereas for Network 2 both steepness at and exact position of the intermediate zone vary greatly. Therefore for boron steels large errors in the hardness measurement occur at a few Jominy positions resulting in a distinct intermediate zone.

For two boron steels Jominy curves are given as predicted by Network 2 and the Boron network. All curves match the measured curves well; the Boron network is more accurate

than Network 2. This is to be expected because the hardness profiles of the boron steels in Network 2 must align with steel grades without boron. The difference between measured and predicted curves for steel (2) at positions 20 mm and higher are due to the modification to correct for hardness values that were not measured.
To study the effect of boron on the hardenability, the predicted hardness values for both steels previously mentioned at 5 mm and 7 mm from the quenched end as a function of the boron concentration are given in Figure 6.16. For the overlapping concentration range the predictions of the Boron network and Network 2 agree. Considering that the hardness at these positions for steels without boron is much lower, the predictions of the Boron network indicate that an optimum boron concentration exists. The predictions of the Boron network do not disagree with the optimum boron concentration (15-20 ppm) reported in literature [30]. However, the optimum boron concentration is not well defined for the Boron network.

7.2.6 Influence of the number of hidden nodes of the Boron network

For modelling the hardenability of the selected boron steels, a network with just one hidden node already gives surprisingly good results. Some five hidden nodes seems to be the optimum. Employing more hidden nodes does not improve the neural network model, but increases the risk of overtraining. As the number of boron steel samples does not allow the subtraction of a sufficiently large validation set, overtraining can not be circumvented directly. Therefore the number of superfluous weight factors is kept low by using a network with only 5 hidden nodes. In this way overtraining is avoided more easily. Comparing the architecture of network 1, where 15 nodes were needed, and the boron network clearly shows that modelling the hardenability for a wide compositional range is far more complex than for a small range. This is because the boron network models steels with similar characteristics, and the model need not align with other steel grades.

7.2.7 Hardness profile network

A neural network with 15 nodes in the hidden layer trained with some 800 data could predict the hardness value at every Jominy position accurately, a neural network with 15 nodes in the hidden layer trained with some 1000 data could not predict the four parameters characterising the complete hardness profile (4.1). The problem has become more complex due to the transformation of 19 hardness measurements into these four parameters. This was to be expected as the parameters of (4.1) interact. As the network showed overtraining

effects, adding more hidden nodes to a network is not likely to solve this problem. The hardness profile network is likely to have become stuck in a local minimum during training. To overcome this problem a larger number of training data and possibly a different learning rule is needed.

7.3 Literature models

7.3.1 Just's model

The equation proposed by Just for the hardness value at the first Jominy position has almost the same accuracy of the prediction as the neural network model (Table 6.5 and Figure 6.17). The hardness at the surface is known to depend mainly on the carbon content of the steel and is hardly influenced by other alloying elements or the steel mill it originates from [7]. Thanks to these two effects the Just model performs well for the first Jominy position. For the comprehensive model Just reports a variance of 4% not accounted for. However, for all Jominy positions but the first one both the comprehensive model and the models for case hardenable and hardenable steels were not applicable to Nedstaal data, when a variance of 4% is required, as this variance is estimated to correspond with a RSD of some 2 HRc. When a model is constructed with identical parameters in a similar way using PLS as is shown in Figure 6.18, a much higher accuracy is attained. This indicates that a statistical model based on chemical composition only is limited to the steel mill for which it was developed.

7.3.2 SteCal model

The SteCal model is reasonably accurate. The error at the quenched end is not as accurate as the Just model, but at the intermediate part of the Jominy curve its prediction is surprisingly accurate. This is probably thanks to the effort that is spent for this model to predict the critical diameter accurately. Far from the quenched end the model has a lower accuracy due to the fixed shape of the curve used.

7.4 Comparison with Partial Least Squares models

7.4.1 Additive PLS model with linear parameters

This model was the result of the PLS technique assuming only linear dependencies. This way the PLS technique is used with the same a priori knowledge as Network 2. As was suspected the error in prediction (Figure 6.18, Table 6.7) is substantially lower than the error in the empirical models; a fairly accurate model is obtained. Compared to Network 2

only the hardness near the surface is predicted with the same accuracy. In all other parts of the Jominy curve the RSD of the PLS 1 model is some 1 to 1.5 HRc higher than the error in Network 2. This is due to the absence of non-linear dependencies and interaction terms.

7.4.2 Additive PLS models with other functional dependencies

Some functional dependencies considered relevant in the literature [17] were added to the PLS model. In this way the prediction of the PLS model at the first 5 Jominy positions improved slightly. Adding these functional dependencies hardly improves the model, while the computational cost and the risk of overfitting increase. The functional dependencies were mainly chosen on statistical basis, i.e. they evolved from trial and error processes. Therefore relevant functional dependencies are easily overlooked. Also interaction terms lack in this model. To incorporate these terms estimates on the nature of the interactions taking place are needed.
These limitations cause the additive PLS models to be inferior to the neural network model.

7.4.3 Multiplicative PLS model

Multiplicative models can handle some non-linearities and are therefore expected to be better for modelling the hardenability. Near the quenched end and far from it, the multiplicative model has a smaller error than the additive models. In the intermediate zone however, this error is larger. This can be explained as follows.
In contrast to the additive models the multiplicative model has a non zero average error in prediction Table 6.7: the mean predicted value is lower than the mean measured value. This is caused by the way the multiplicative model is implemented in the linear regression model. To linearise the model (2.6) is used. The parameters of this equation are determined using linear regression. For this equation the average error in prediction is zero. The error structure in this model is additive. By re-transforming (2.6) into (2.5) an additive error in (2.6) becomes a multiplicative error in (2.5), i.e. an error that was added or subtracted in (2.6) becomes a factor in (2.5). As this transformation is exponential, a positive error in a transformed output variable results in a higher error on the actual variable than a negative deviation of equal size. In this way an extra error is introduced in the multiplicative model. The size of this extra error depends strongly on the size of the error in the transformed model. This extra error amplifies the (higher) error in the intermediate zone more than the error in other parts of the Jominy curve. The only accurate way to construct multiplicative models is by using non-linear regression.

Using multilinear regression techniques accurate statistical models can be constructed for the prediction of the Jominy curve. However, all these models are outperformed by the

neural network models. The RSD of the neural network is still 1 HRc lower than the most accurate multilinear regression model.

7 Conclusions

Hardenability is a complex multivariant non-linear problem which has been solved successfully using artificial neural networks. The neural network clearly outperforms literature models and linear regression models over a wide range of steel compositions.

The modelling error equals the experimental error in the data. No further modelling advances are possible with these data.

Further improvements of a statistical hardenability model require more accurate input data (hardness values and chemical composition).

A statistical model predicting the hardenability from chemical composition is only accurate for the steel production process the model is based upon.

Acknowledgements

I express my thanks to:

Hans Janssen and Marc van Woensel of Nedstaal B.V. for supplying the necessary data and knowledge of the production process of Nedstaal B.V,

Ton de Weijer for his neural network programmes and experience,

Willem Vermeulen for his assistance and supervision of this research,

Sybrand van der Zwaag for his incentive, supervision and patience.

References

[1] A. Freiburg, Stahl u. Eisen, 1986, 106, 63-68.
[2] D. Frodl, K. Krieger et al., Stahl u. Eisen, 1986, 106, 69-75.
[3] Anon. British standard for Determining hardenability of steel by end quenching BS 4437, British Standards Institution,London, 1987.
[4] M.A. Grossmann, M. Asimow et al., Hardenability in alloy steels, American Society for Metals, Cleveland, 1939.
[5] J.C. Boyd, J. Field, Calculation of the Standard End-Quench Hardenability curve from Chemical Composition and Grain Size, New York, 1946.
[6] I.R. Kramer, S. Siegel et al., Trans. Amer. Inst. min. metall. Eng., 1946, 167, 670-697.
[7] D.V Doane, J.S. Kirkaldy (Ed.), Hardenability concepts with application to steel, The Metallurgucal of Society AIME, Warrendale (PA), 1978, 421-447.
[8] C.F. Jatcak, Metal Progress september 1979, 60-69.
[9] A. Moser, A. Legat, Härterei-Tech. Mitt., 1969, 24, 100-104.
[10] J. Field, Met. Progr.,1943, 43, 402-405.
[11] R. Kern, M.E. Suess, Steel Selection, John Wiley & sons, New York.
[12] metal progress,1975,108, 86-87
[13] C.F. Jatcak, Metal Progress september 1979, 66.
[14] E.A. Geary, W.T. Cook et al., Materials Science Forum, 1994, 163-165, 689-696.
[15] K. Thomas, E.A. Geary et al., Materials & Design, 1992, 13,17-22.
[16] Anon., ASTM Designation A 255-89, Standard Method for End Quench Test for Hardenability of Steel, in The 49th Annual Book of ASTM Standards, American Society for Testing and Materials, Philadelphia (PA), 1989, 27-44.
[17] E. Just, metal progress, 1969,96, 87-88
[18] E. Just, Z. wirtsch. Fertigung, 1981, 76, 424-428
[19] R.H. Myers, Classical and modern regression with applications, Duxbury Press, Boston (MA), (1986).
[20] D.E. Rumelhart, J.L. McCleland et al., Parallel distributed processing, 8th Ed., Vol 1, The MIT Press, Cambridge (MA), 1986.
[21] J. Zupan, J Gasteiger, Neural networks for chemists, VCH Verlaggeselschaft mbH, Weinheim 1993.
[22] R. Hecht-Nielsen, Neurocomputing, Addison Wesley Publishing company, Reading (MA), 1990.
[23] P.J. van der Wolk, Modelling materials production processes using artificial neural networks, Delft, 1994.
[24] P. Geladi, B.R. Kowalski, Anal. Chim. Acta., 185, 1986, 1-18
[25] P. Geladi, B.R. Kowalski, Anal. Chim. Acta., 185, 1986, 19-32
[26] M.H. El-Sharkawi, R.J. Marks (Ed.), Applications of neural networks to power systems, Seattle (WA), 1991, 231-136.
[27] P. Tarin-Remohi, SteCal, ASM international, Materials Park (OH), 1989.
[28] W. Crafts, J.L. Lamont, Trans. Amer. Inst. min. metall. Eng.,1946, 167, 698-718.
[29] anon., Annnual book of ASTM standards,1971, 31, 424-431, Philadelphia.
[30] D.V Doane, J.S. Kirkaldy (Ed.), Hardenability concepts with application to steel, The Metallurgical of Society AIME, Warrendale (PA), 1978, 421-447.