ARTICLE IN PRESS

ELSEVIER

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Non-Crystalline Solids xxx (2003) xxx–xxx

JOURNAL OF NON-CRYSTALLINE SOLIDS

www.elsevier.com/locate/jnoncrysol

A machine learning approach to the estimation of the liquidus temperature of glass-forming oxide blends

Catherine Dreyfus [a,*], Gérard Dreyfus [b]

[a] Université Pierre et Marie Curie, Laboratoire PMC, Tour 22, 2ème Etage, C.P. 77 4, place Jussieu, 75005 Paris cedex 05, France
[b] École Supérieure de Physique et de Chimie Industrielles, Laboratoire d'Électronique 10, rue Vauquelin, 75005 Paris, France

Received 2 July 2002; received in revised form 29 July 2002

Abstract

Many properties of glasses and glass-forming liquids of oxide mixtures vary in a relatively simple and regular way with the oxide concentrations. In that respect, the liquidus temperature is an exception, which makes its prediction difficult: the surface to be estimated is fairly complex, so that usual regression methods involve a large number of adjustable parameters. Neural networks, viewed as parameterized non-linear regression functions, were proved to be *parsimonious*: in order to reach the same prediction accuracy, a neural network requires a smaller number of adjustable parameters than conventional regression techniques such as polynomial regression. We demonstrate this very valuable property on some examples of oxide mixtures involving up to five components. In the latter case, we show that neural networks provide a sizeable improvement over polynomial methods.

PACS: 0260; 6470

1. Introduction

Most industrial glasses are blends of oxides, in which the total number of different oxides can be high. The composition of these blends is a key factor for many physical properties of the glasses; the prediction of the latter is a problem of industrial importance. Many oxide glass properties vary smoothly with composition across a wide range of concentration; therefore, data obtained from a large amount of experimental work, extending over years, can be used for inferring many physical properties of glasses from their compositions [1]. Even simple additive equations are found to be quite successful for several important properties, such as thermal expansion, density at room temperature, refractive index [2]. Several approximation procedures have been proposed for the viscosity ([3] and references therein). Among the properties of glasses, the liquidus temperature is expected to be the most difficult to predict [2]: although the liquidus temperature depends only on the composition of the blend, liquidus curves exhibit sharp minima as well as inflexions and maxima across sections of most systems, so that multiple

[*] Corresponding author. Tel.: +33-1 44 27 42 47; fax: +33-1 44 27 44 69.
E-mail address: cad@ccr.jussieu.fr (C. Dreyfus).

0022-3093/02/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0022-3093(02)01859-8

linear regression analysis cannot be expected to be successful. Yet getting a reliable prediction of the liquidus temperature would be a worthwhile result: this temperature is the highest temperature at which crystals are at equilibrium with the liquid, and the value of the viscosity at the liquidus temperature is a key factor in the choice of the glass process to be used. Modeling this industrially important quantity has been the goal of several investigations: in a study of liquidus temperature of several binary glasses [4], thermodynamic modeling was found to be quite successful, but another study concluded that empirical models derived from series of glasses and statistical analysis still remain the most useful approach, because thermodynamic modeling might require the estimation of thermodynamic parameters that are not known accurately [5,6]. Yet, as the liquidus temperature exhibits a very strong non-linear behavior, it is difficult to account for it in a large range of composition, even by high order polynomial approximations. In order to eliminate the problems that arise near the boundaries, it was suggested to divide the data into different regions [7,8]. As long as the composition was kept within the same primary phase field of the phase diagram, polynomial relations between the composition and the liquidus temperature could be derived. Multiple polynomial regression can be found in several papers in order to predict the liquidus temperature in usual industrial glasses with up to 10 different oxides [6] and including some high level waste glasses [9]. The most recently published approximation [6] includes terms up to degree four.

In the present paper, we resort to a recently developed non-linear regression technique, namely, neural networks, in order to model the diagram of liquidus temperature versus composition, for blends with up to five components.

Non-linear regression is nowadays an important part of the engineer's toolbox, because of its numerous applications in data analysis, automatic classification and discrimination, and process modeling and control. One of the most recent developments in the field has been the introduction of networks of formal or artificial (as opposed to natural) neurons, also termed neural networks [10]. A neural network is a non-linear parameterized function, whose parameters (usually called weights) are estimated from examples in a process called *training*. After training, the neural network is expected to generalize, i.e. to give an appropriate response to situations that are not present in the set of examples from which it is trained.

In this paper, we apply neural networks to the prediction of the liquidus temperature of glass-forming liquids of oxide blends. In the first section, we describe the different glass-forming systems examined in this work. The main definitions and properties of neural networks are briefly summarized in Section 2. In Section 3, we describe the results obtained in the prediction of four glass-forming liquids, two ternary blends, one four-oxide blend, and one five-oxide blend, and we consider the problem of the size of the training/test sets. Finally, we compare polynomial models to neural models.

2. Databases

Numerous liquidus temperature measurements have been performed over the years, because of the importance of this parameter in industrial processes. Therefore, useful data are scattered throughout numerous technical reviews, so that collecting numerical data is a lengthy, time-consuming work. In addition, data concerning the most useful systems are often included into proprietary databases of glass manufacturers, the access to which is not granted. Therefore, we resorted to the SciGlass database, available commercially from SciVision Inc. [11], in which a large amount of experimental data points on glassy systems is gathered.

As an example of the non-regular variation of the liquidus temperature with the concentration of its constituents, the liquidus temperatures versus the molar concentration in alkali oxide of the binary blend (Li_2O, SiO_2) is shown in Fig. 1(a). The complexity of the variation of the liquidus temperature obviously increases with the number of constituents, as illustrated in Fig. 1(b). Glass-forming blends investigated in the present work feature up to five constituents; constituents and

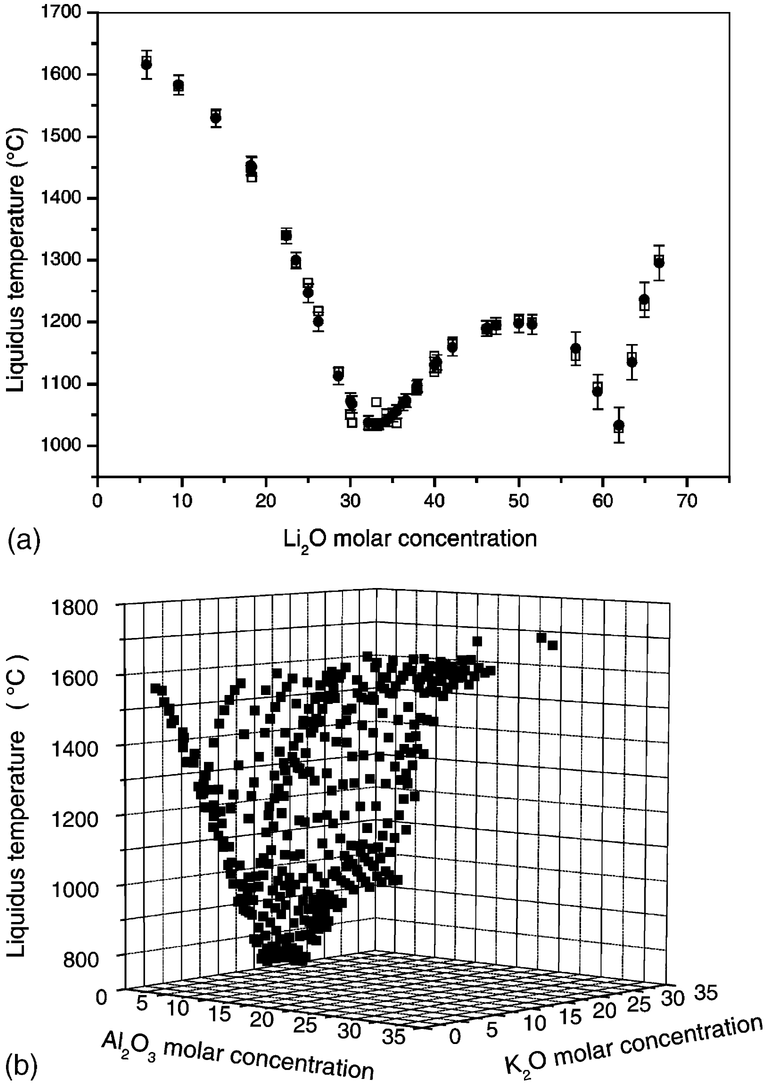

Fig. 1. (a) (Li_2O, SiO_2): variation of the liquidus temperature with the Li_2O molar concentration (□) experimental (●) estimated with a four-neuron network; bars indicate the 95% confidence interval; (b) 3D representation of the liquids temperature versus molar concentrations in (K_2O, Al_2O_3, SiO_2) blend.

composition range are reported in Table 1. Several glass-forming systems were chosen, the main criterion being the number of data points available. We restricted our investigation to systems for which more than 100 data points were available. Nevertheless, they cover a fairly large range of concentrations for each species. Only those reported data points for which the constituent concentrations summed to a number between 99% and 100% were selected; therefore, some experimental temperature points are probably affected by the presence of some minor constituents with unknown concentration. This was regarded as a disturbance, expected not to affect grossly the quality of the prediction. Bibliographical references extracted from SciGlass database are given in Appendix A. In some cases, experimental errors on the value of the liquidus temperature are reported; they vary between ±1 and ±5 K, depending of the reference.

Table 1
Oxide blends, number of data points, composition range and sum of all constituents

	Number of data points	Molar percentage range	Sum of all constituents
Al_2O_3, K_2O, SiO_2	384	$0.6 < n_{Al2O3} < 31$ $0.65 < n_{K2O} < 32$ $46 < n_{SiO2} < 96$	100
CaO, K_2O, SiO_2	155	$0.75 < n_{CaO} < 45$ $2.4 < n_{K2O} < 46$ $33 < n_{SiO2} < 86$	99.99–100
Na_2O, CaO, Al_2O_3, SiO_2	893	$0 < n_{Na2O} < 52.7$ $0 < n_{CaO} < 79.5$ $40 < n_{Al2O3} < 45.4$ $5.6 < n_{SiO2} < 95.74$	99.15–100.1
Na_2O, CaO, Al_2O_3, MgO, SiO_2	309	$0.14 < n_{MgO} < 18.1$ $2.28 < n_{Na2O} < 18.97$ $0.3 < n_{CaO} < 33$ $0.06 < n_{Al2O3} < 11.47$ $46.12 < n_{SiO2} < 79.94$	99.07–100.00001

3. Neural networks

This section is intended to provide the necessary background on neural networks in the context of the present investigation.

3.1. Definitions

3.1.1. Formal neurons

A formal neuron is a function y whose value depends on parameters (or weights). The variables $\{x_n, n = 1 \text{ to } N\}$ of the function are the inputs of the neuron, and the value of the function is called the output of the neuron. Most frequently, the non-linear function of a neuron can be split into two parts:

(i) a weighted sum v of the inputs:

$$v = \sum_{n=1}^{N} w_n x_n + w_0, \tag{1}$$

where x_n is the nth input of the neuron; w_n, the weight (or parameter) related to the nth input of the neuron; and w_0, an additional parameter which can be viewed as the weight related to an additional input variable equal to 1 (this input is called the *bias* of the neuron).

(ii) a non-linear function (termed activation function) of this sum.

Any bounded, differentiable function can be used, but, for reasons that will be developed below, the most frequent activation function is a sigmoid function such as

$$y = \tanh(v), \tag{2}$$

where y is the output of the neuron.

It is useful to note that the output of the neuron is non-linear with respect to the variables x_n and to the parameters w_n. The latter property is of importance, as shown below.

3.1.2. Neural networks

A neural network performs a non-linear function of its inputs, which is a combination of the functions of its neurons. For reasons that are explained below, the general form of a neural network intended to perform non-linear regression is the following: the output of the network is a linear combination of the non-linear functions performed by ‘hidden’ neurons, i.e. neurons whose inputs are the inputs of the network. The output g of the network is given by

$$\begin{aligned} g &= p_0 + \sum_{m=1}^{M} p_m y_m \\ &= p_0 + \sum_{m=1}^{M} p_m \tanh\left(\sum_{n=1}^{N} w_{mn} x_n + w_0\right), \end{aligned} \tag{3}$$

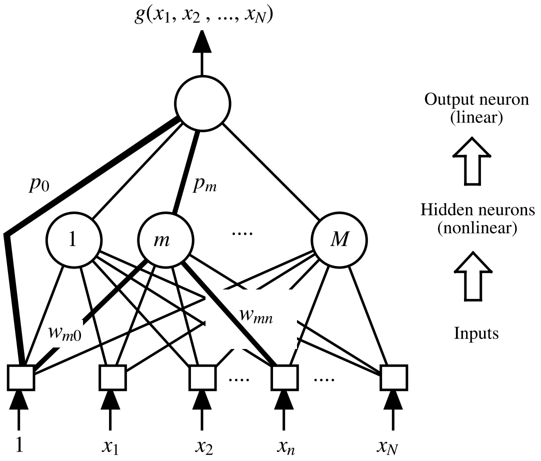

Fig. 2. Schematic representation of a feedforward neural network, or multilayer perceptron, as used in the present paper.

where M is the number of hidden neurons; p_m, the weight of the 'connection' between hidden neuron m and the output; p_0, the weight of the connection between the bias (network input equal to 1) and the output neuron; and w_{mn}, the weight of the connection between input n and neuron m. A graphical representation of the network is shown in Fig. 2. Note that the output neuron is a 'linear neuron', whose output is simply a weighted sum of its inputs, i.e. of the outputs of the hidden neurons; no non-linearity is involved in this neuron. Therefore, a neural network without hidden neurons is simply an affine function of the variables.

Such networks are termed feedforward networks, or multilayer perceptrons.

3.2. Neural networks are parsimonious universal approximators

The universal approximation property can be stated as follows: any sufficiently regular, bounded function can be uniformly approximated, to an arbitrary accuracy, in a given volume of input space, by a neural network of the type shown in Fig. 2, or described by relation (3), having a finite number of hidden neurons.

Other families of functions can also perform uniform approximation: polynomials, Fourier series, spline functions, etc. The specific property of families of approximators that are non-linear with respect to their parameters (such as neural networks, as mentioned in Section 3.1.1) is their parsimony: the number of parameters that are required to obtain a given level of accuracy is linear with respect to the number of inputs of a neural network, whereas it varies exponentially with the number of inputs in the case of approximators that are linear with respect to their parameters, such as polynomials [12]. Therefore, the parsimony of neural networks can essentially be taken advantage of for models that have more than two inputs; the larger the number of inputs, the more valuable this property. Fig. 3 compares the variation of the number of parameters when the number of hidden neurons increases, for a given number of variables, to the variation of the number of parameters involved in a polynomial regression when the degree of the polynomial increases.

The origin of parsimony can be understood as follows: consider an approximator that is linear with respect to its parameters, such as a polynomial. For simplicity, let us consider a third-degree polynomial with a single variable $x : g(x) = w_0 + w_1x + w_2x^2 + w_3x^3 \cdot g(x)$ is a linear combination of four functions (1, x, x^2, x^3) which are determined once and for all; by contrast, the output g of a neural network, as described by relation (3), is a linear combination of the outputs of the hidden

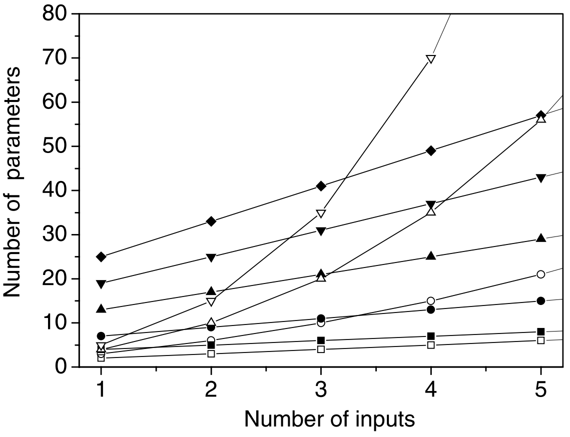

Fig. 3. Comparison of the number of parameters versus the number of inputs for neural and polynomial regression. Hidden neurons in the neural net: 2 (■), 3 (●), 4 (▲), 6 (▼), 8 (◆). Degree of the polynomial regression: 1 (□), 2 (○), 3 (△), 4 (▽).

neurons (the tanh functions), which depend on the weights of the first layer. Thus, instead of combining functions of fixed shapes, one combines functions whose shapes are adjustable. This provides additional degrees of freedom, which, understandably, allow one to build more complex functions with the same number of parameters, or to build functions of comparable complexity with a smaller number of parameters.

Capitalizing on this property, neural networks have been extensively applied as non-linear regression tools in a wide variety of areas of engineering: static modeling [13], dynamic modeling (using recurrent neural networks) [13], automatic discrimination (e.g. for pattern recognition) [10], process control [14], etc. To the best of our knowledge, neural network predictions of liquidus temperatures of oxide mixtures have not been reported in the literature. Investigations of binary and ternary molten salts blends [15] and blends of alkali metal/alkali halide [16] have been published.

3.3. From function approximation to non-linear regression

Actually, neural networks are essentially never used for computing a uniform approximation of a known function. Instead, a finite set (called the *training set*) of N_L measurements y_p^k ($k = 1$ to N_L) of the quantity of interest y_p, and of the corresponding vector of inputs $\boldsymbol{x}^k$, is available. The measured output is assumed to be the sum of an unknown function $r(\boldsymbol{x})$ (called the regression function of the quantity of interest) and of a zero-mean noise. If $r(\boldsymbol{x})$ is a non-linear function, neural networks are excellent candidates for approximating the unknown function $r(\boldsymbol{x})$, given the available data. To this end, the parameters of the network are adjusted through training from the available examples.

3.4. Neural network training

Training is the procedure whereby the parameters of the network are computed so as to minimize a cost function that takes into account the modeling errors of the network. Most frequently, the least squares cost function is used:

$$J(\boldsymbol{w}) = \sum_{k=1}^{N_\mathrm{L}} \left(y_p^k - g^k(\boldsymbol{w}) \right)^2, \tag{4}$$

where $\boldsymbol{w}$ is the vector of the weights of the network; y_p^k, the measured value of the quantity of interest for example k; and $g^k(\boldsymbol{w})$, the output of the model, with weight vector $\boldsymbol{w}$, for example k. If perfect modeling is achieved, all the terms of the sum are equal to zero, so that the cost function $J(\boldsymbol{w})$ is equal to zero, which is the minimum value of the function. Since the measurements are noisy, perfect modeling is definitely not desirable, but still a (non-zero) minimum of the cost function is sought, such that the mean square error $J(\boldsymbol{w})/N_\mathrm{L}$ is of the order of the variance of the noise.

Since this cost function is not linear with respect to the weights, the standard least-squares procedure cannot be used for obtaining the optimal weight vector. Instead, minimization of the cost function with respect to the weights must be performed through iterative updating of the weights, according to one of several algorithms that make use of the gradient of the cost function. An increased complexity of parameter estimation is the price that one has to pay for taking advantage of parsimony.

Therefore, each iteration of the minimization procedure is performed in two steps:

(i) computation of the gradient of the cost function with respect to the weights; this is performed by a computationally economical algorithm called 'backpropagation' [10];
(ii) weight updating by a minimization algorithm such as the Broyden–Fletcher–Goldfarb–Shanno algorithm or the Levenberg–Marquardt algorithm [17].

Training is terminated when the cost function no longer decreases significantly, indicating that a minimum of the cost function $J(\boldsymbol{w}_\mathrm{LS})$, corresponding to a weight vector $\boldsymbol{w}_\mathrm{LS}$, has been found.

3.5. Performance evaluation and model selection

Performance evaluation is a crucial step in the design of neural models, for two reasons:

- the model designer must find a tradeoff between flexibility and generalization ability: if the number of parameters (or, equivalently, of hidden neurons) is too small, the model is not flexible enough to learn the data; conversely, if the number of parameters is too large, the model fits the noise present in the data, hence gives poor predictions on data that is not present in the training set; this phenomenon is known as *overfitting*;
- since the output of the model is non-linear with respect to the weights, the cost function is not quadratic with respect to the weights, hence does not have a unique minimum; therefore, the iterative minimization procedure leads to different minima, hence produces different models, for different initializations of the parameters; therefore, for a given number of hidden units, a choice must be performed between the models obtained after minimization.

In the present work, a recently developed model selection technique, termed local overfitting control via leverages was used [18]. It is based on the computation of the *leverage* h_{kk} of each example k. The leverage of an observation is the proportion of the adjustable parameters of the model that is used to fit that observation, if the latter belongs to the training set. The sum of the leverages of the observations is equal to the number of parameters: the higher the leverage of a given example, the higher the influence of that example on the parameters of the model. As a consequence, if an example that has a large influence on the parameters of the model is withdrawn from the training set, and if a new model is trained with the remaining examples, the resulting model will be substantially different from the previous one, and the prediction error on the example that has been left out will be large. A good approximation of the prediction error on example k when it is left out of the training set is given by

$$R_k^{(-k)} = \frac{R_k}{1 - h_{kk}}, \tag{5}$$

where R_k is the modeling error on example k when the latter belongs to the training set; h_{kk}, the leverage of example k; and $R_k^{(-k)}$, the prediction error on the example k had the latter been withdrawn from the training set, and the model been trained with all other examples. A proof of relation (5), and details of the computation of the leverages, can be found in [18] and references therein.

The quantity

$$E_p = \sqrt{\sum_{k=1}^{N} \left[R_k^{(-k)}\right]^2}, \tag{6}$$

called the *leave-one-out score*, is known to be an unbiased estimate of the generalization error of the model. Hence, E_p is a natural criterion for model selection: the smaller E_p, the better the generalization ability of the model. Furthermore, confidence intervals on the prediction of the model can be computed from the leverages.

In addition, a test set, made of examples that were not used for training, nor for model selection, was used for estimating the performance of the model selected by the procedure described above. The performance was estimated through the standard prediction error on the test set:

$$E_{\mathrm{T}} = \sqrt{\frac{1}{N} \sum_{k=1}^{N_{\mathrm{T}}} \left[y_p^k - g^k(\mathbf{w}_{\mathrm{LS}})\right]^2},$$

where N_{T} is the number of elements in the training set.

It will be demonstrated in Section 4 that, as expected, the standard prediction error on the test set first decreases as the number of neurons increases, and starts increasing when the number of parameters is large enough for overfitting to occur. This is in contrast to the behavior of the standard prediction error on the training set, defined as $E_{\mathrm{tr}} = \sqrt{J(\mathbf{w}_{\mathrm{LS}})/N_{\mathrm{L}}}$, which decreases as the number of hidden neurons increases.

3.6. Numerical procedure

All numerical experiments reported below were performed with the commercial software package NeuroOne [19]. For each experiment, the set of available measurements was divided into two sets, the elements of which were chosen randomly; 80% of the available data were used as the training set, whereas the rest of the data was the test set.

Weights were initialized randomly, with small enough values so that all sigmoids were in their linear region. For a given dataset and a given number of hidden neurons, 100 trainings were performed with different weight initializations. The gradient of the cost function was computed by backpropagation, and the weight updates were performed with the Levenberg–Marquardt algorithm. The model having the smallest leave-one-out score (6) was selected.

4. Results

Neural network regression was performed on two ternary databases: (Al_2O_3, K_2O, SiO_2), (CaO, K_2O, SiO_2), and on the multiple databases (CaO, Na_2O, Al_2O_3, SiO_2) and (Al_2O_3, CaO, Na_2O, MgO, SiO_2). The number of data points chosen respectively for the training set and for the test set is reported in Table 1. The number of hidden neurons was varied between 0 and 8, 0 corresponding to a multilinear regression. Inputs were the molar fractions of each component, except SiO_2: since the molar fractions sum to 1, there is no point in taking all of them into account. Thus, the number of inputs is 2 for ternary blends, and 3 and 4 for the others respectively. The output is the liquidus temperature. We first consider the generalization ability of the neural models thus obtained; the influence of the size of the training set on the accuracy of the regression is subsequently investigated. The last section is devoted to a comparison between the performances of neural and polynomial regressions.

4.1. Generalization ability of the neural models

Fig. 1(a) shows, as an example, the liquidus temperature versus concentration obtained with a model having four hidden neurons in the simple case of a binary mixture, together with the 95% confidence intervals, computed as described in [18]. As expected, the confidence interval increases in the regions of input space where training data is scarce.

For more complex mixtures, other representations must be used: Figs. 4–6 are scatter plots that display the liquidus temperature estimated by the network, together with the confidence intervals on the estimation, versus the measured liquidus temperature for one of the ternary, the four-oxide and the five-oxide glasses. In all cases, the results, both on the training set and on the test set, are less and less scattered as the number of hidden neurons is increased. Figs. 7–10 show the standard prediction error on the test set, and the standard prediction error on the training set, as a function of the number of parameters, for the four-oxide mixtures under investigation. As expected, the prediction error on the training set decreases as the number of neurons increases, whereas the prediction error on the test set starts increasing when the number of parameters becomes too large, indicating the onset of overfitting.

4.2. Size of the training set and generalization ability

One of the difficulties in the liquidus temperature approximation is the number of data points required to obtain a meaningful regression. In order to try to assess the required size of the training dataset in a systematic way, we used the following procedure: selecting the blend (Al_2O_3, K_2O, SiO_2), whose dataset contains 384 data points, Fig. 8 shows that the standard prediction errors on the training set and on the test sets are similar up to 35 parameters. Let us now divide the dataset into three parts: removing the data points belonging to the former test set, the K remaining points are split into a new training set with 0.8 K elements and a new test set with 0.2 K elements, so that one can perform the training process again on the reduced database. The total test set includes both the former test set and the new test set. One can thus compute the standard prediction error of the training set, the same quantity of the total test set, and the leave-one-out score. Iterating the procedure, the (training + test) set is further reduced. Fig. 11 shows the standard prediction errors on the training sets, on the test sets and the leave-one-out scores. The standard prediction errors on the training sets are almost constant, whereas the standard prediction error on the test set and the leave-one-out score increase regularly

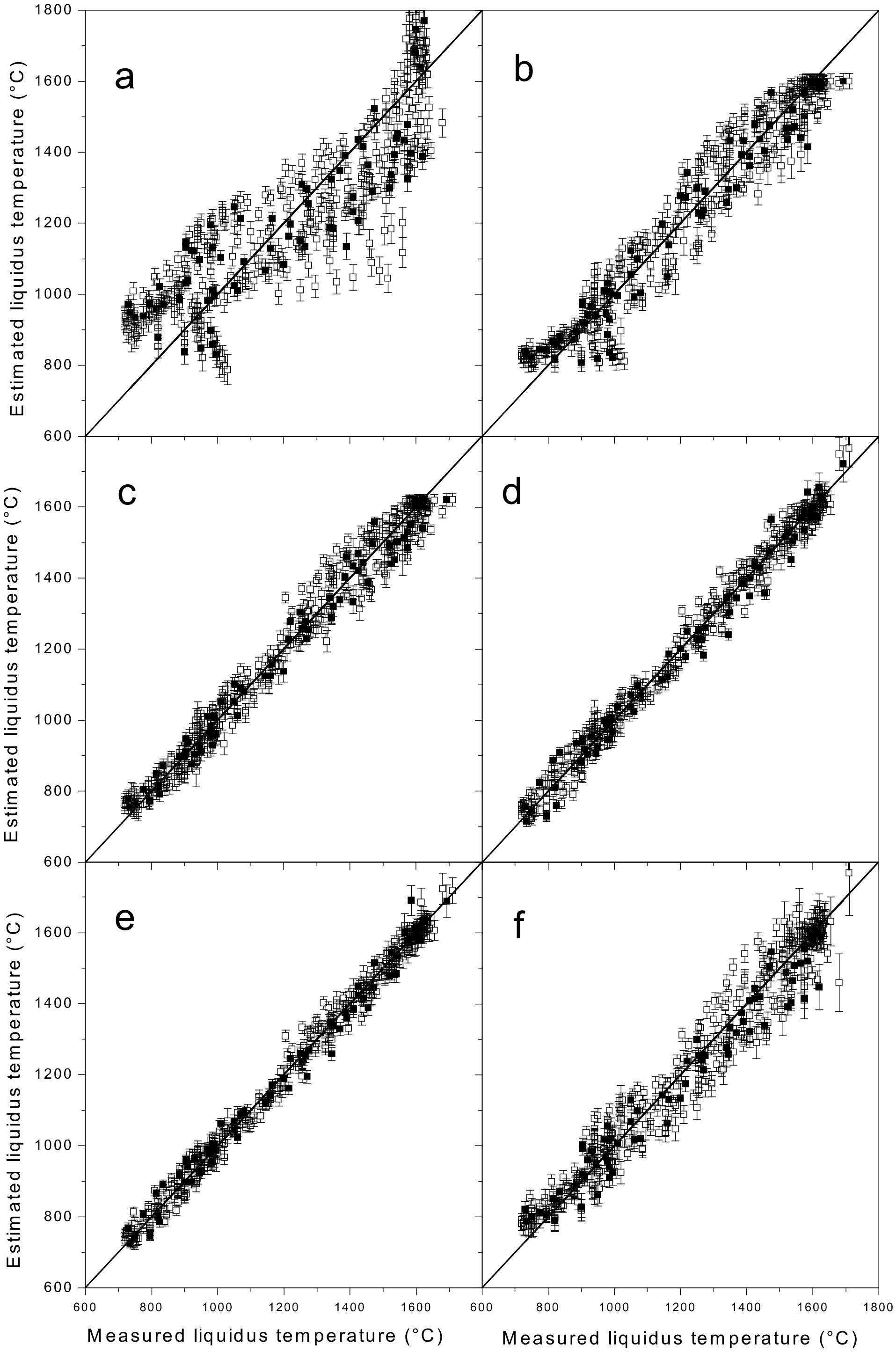

Fig. 4. Estimated versus measured liquidus temperatures (K_2O, Al_2O_3, SiO_2) oxide blend: (□) training set, (■) test set. (a) 0 neuron; (b) 2 neurons; (c) 4 neurons; (d) 6 neurons; (e) 8 neurons; (f) fourth order polynomial regression.

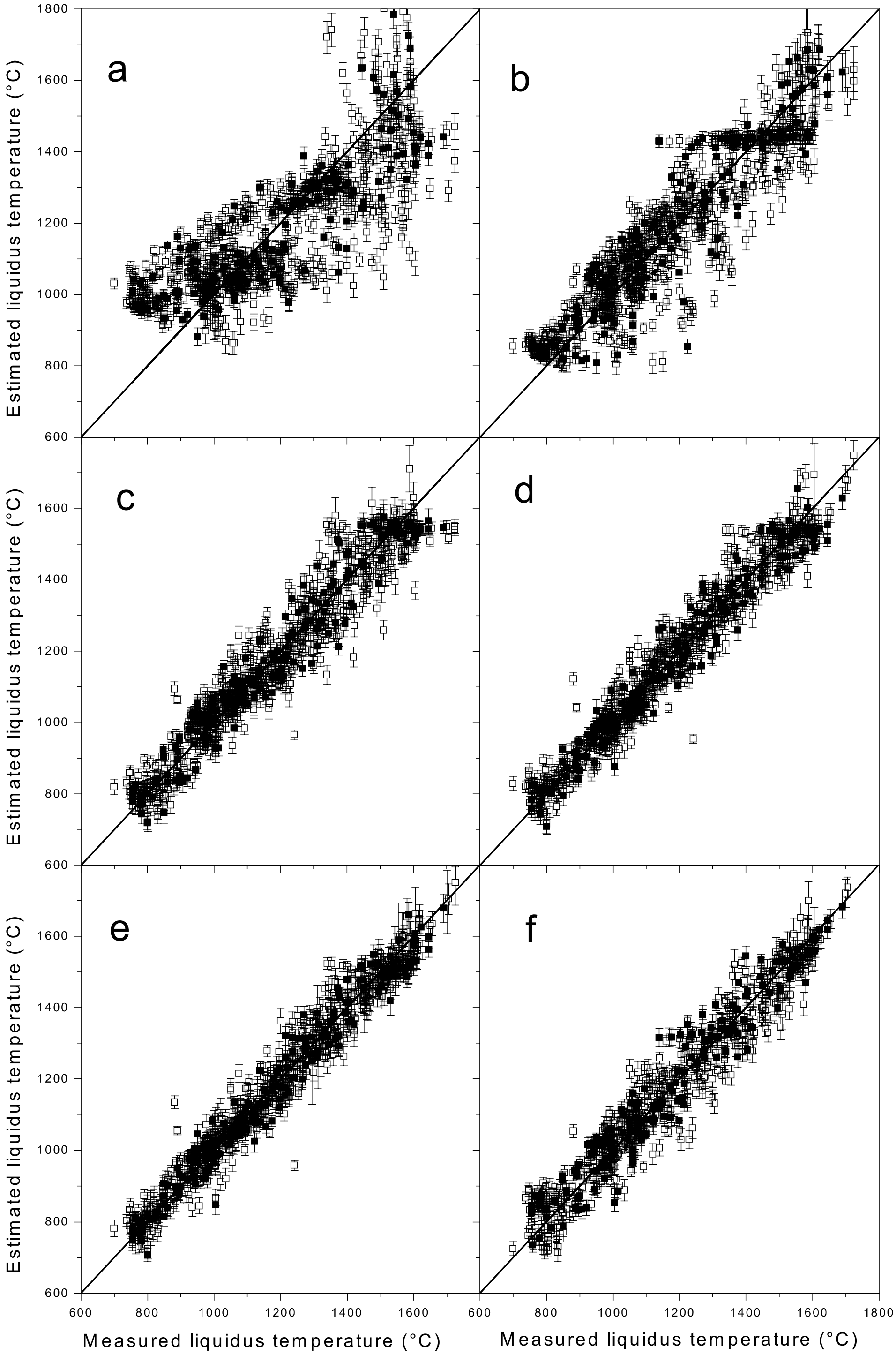

Fig. 5. Computed versus measured liquidus temperatures in (Na_2O, CaO, Al_2O_3, SiO_2) oxide blend: (□) training set, (■) test set. (a) 0 neuron; (b) 2 neurons; (c) 4 neurons; (d) 6 neurons; (e) 8 neurons; (f) fourth order polynomial regression.

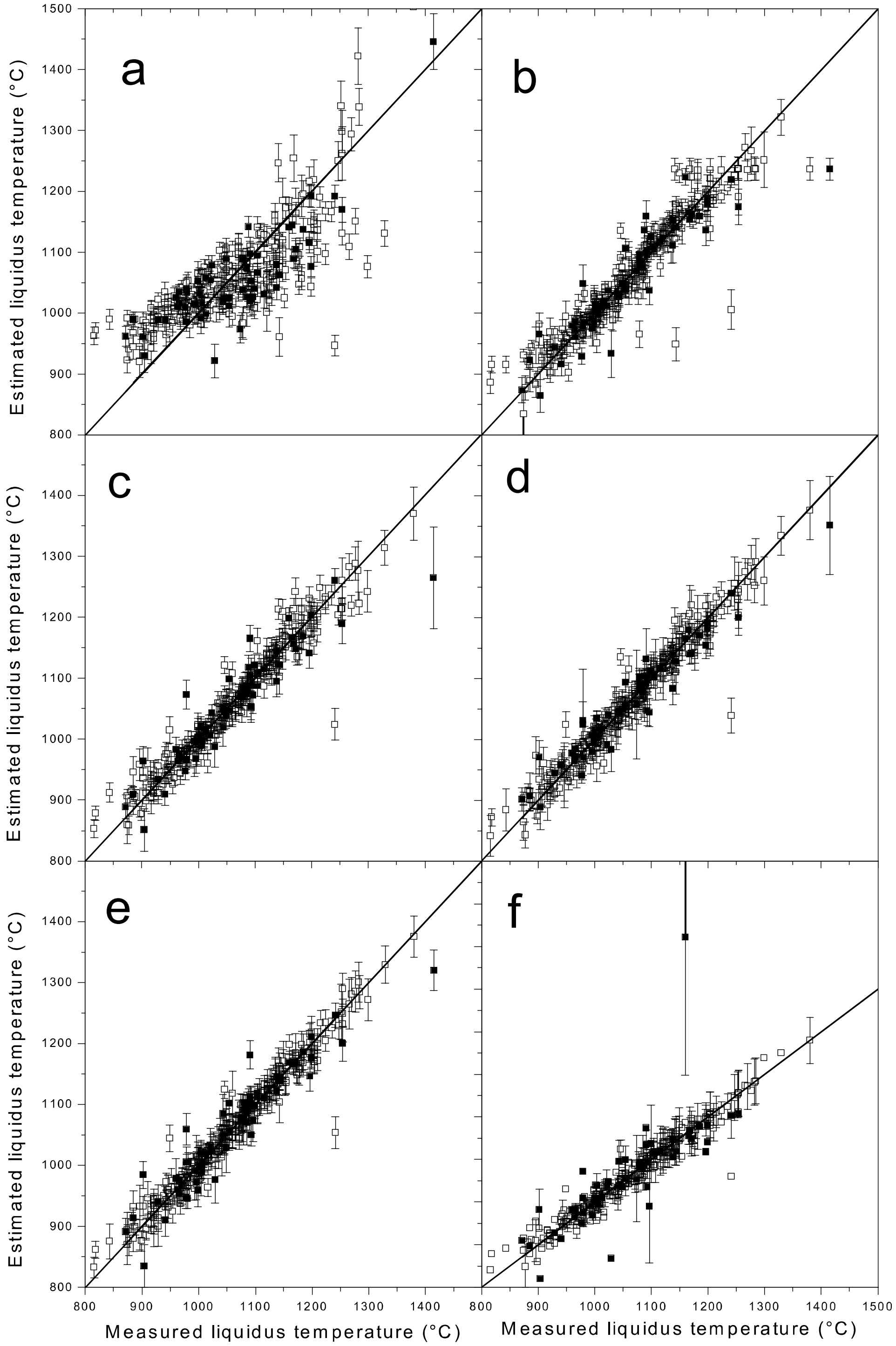

Fig. 6. Computed versus measured liquidus temperatures in (Al_2O_3, Na_2O, CaO, MgO, SiO_2) oxide blend: (□) training set, (■) test set. (a) 0 neuron; (b) 2 neurons; (c) 4 neurons; (d) 6 neurons; (e) 8 neurons; (f) fourth order polynomial regression.

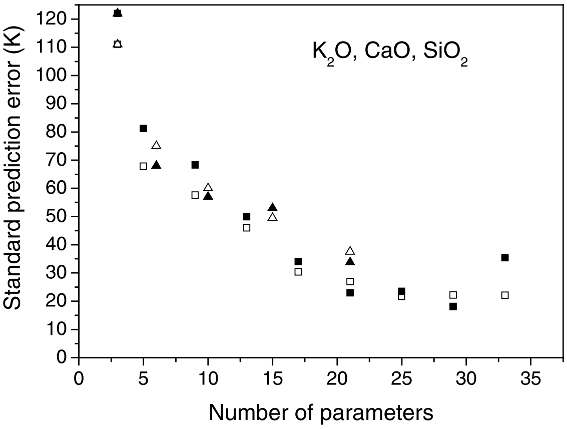

Fig. 7. (K_2O, CaO, SiO_2) blend: standard prediction error versus number of parameters for the two regression methods. Neural regression: (□) training set, (■) test set. Polynomial regression: (△) training set, (▲) test set.

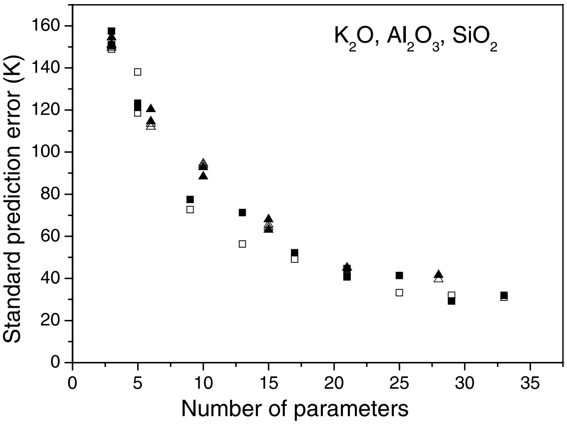

Fig. 8. (K_2O, Al_2O_3, SiO_2) blend: standard prediction error versus number of parameters for the two regression methods. Neural regression: (□) training set, (■) test set. Polynomial regression: (△) training set, (▲) test set.

in a similar way and become substantially larger than the standard prediction error on the training sets when the number of training points decreases. Reducing the number of training points, the excess in standard prediction error of the test set spreads out over the whole temperature range, showing that the training set is becoming too small to allow a reasonable prediction to be performed.

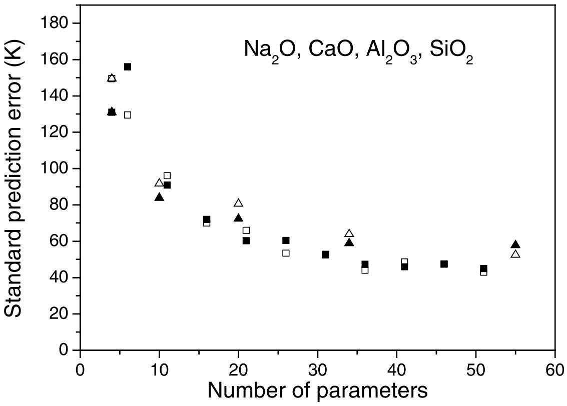

Fig. 9. (Na_2O, CaO, Al_2O_3, SiO_2) blend: standard prediction error versus number of parameters for the two regression methods. Neural regression: (□) training set, (■) test set. Polynomial regression: (△) training set, (▲) test set.

4.3. Polynomial regression versus neural regression

As summarized in the introduction, several polynomial regression models have been proposed to approximate the liquidus temperature. The simplest is a linear approximation scheme, but it became soon apparent that this approximation failed because higher order terms had to be involved. The highest order polynomial to be found in the literature involves expansion of the temperature with respect to the concentrations up to order four. In such a scheme, and if all the terms of the polynomial expansion are kept, the number of parameters increases exponentially with the number of inputs. Thus the number of monomials retained in the model is usually limited, based on 'educated guesses' about monomial relevance. [1] We consider here only the general case: we estimate the prediction accuracy of polynomial regression up to order four when all terms are kept and we compare it to the performance of a neural network involving the same number of parameters. The scatter plots for polynomial regression are shown for three-, four- and five-oxide blends in

[1] A statistical test (Fisher's test) is available for monomial selection; to the best of our knowledge, they have never been used for liquidus temperature prediction.

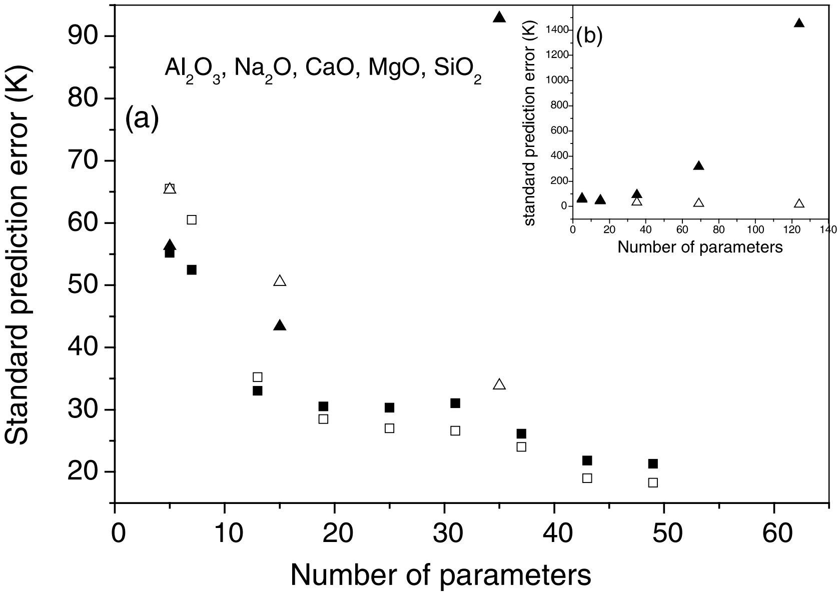

Fig. 10. (Na_2O, CaO, Al_2O_3, MgO, SiO_2) blend: standard prediction error versus number of parameters for the two regression methods. (a) Neural regression: (□) training set, (■) test set. Polynomial regression: (△) training set, (▲) test set. Inset: (b) polynomial regression.

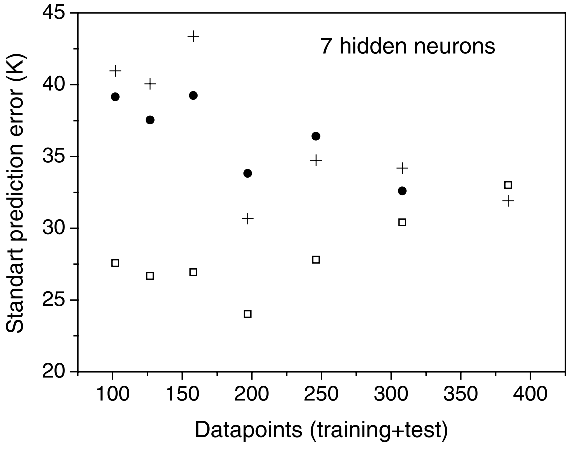

Fig. 11. (K_2O, Al_2O_3, SiO_2) blend: seven neurons regression for a variable size of the training set. Standard prediction error versus the size of the training set : (□) training set, (•) total test set, (+) leave-one-out score.

Figs. 4(f), 5(f) and 6(f), respectively. Because the number of parameters strongly depends on the number of inputs, the efficiencies of the regressions shown in Fig. 4(f), 5(f) and 6(f) must be compared respectively to Figs. 4(c), 5(d) and 6(e). In the first case (ternary mixture), the results are essentially similar (fourth-degree polynomial regression: 15 parameters: $E_{\mathrm{tr}} = 65$, $E_{\mathrm{T}} = 65$; neural regression, four hidden neurons, 17 parameters: $E_{\mathrm{tr}} = 50$, $E_{\mathrm{T}} = 55$). Figs. 7 and 8 also report results of polynomial regression on ternary mixtures: the performances of polynomial regression are similar to those of neural regression for both ternary mixtures. More complex mixtures are shown in Figs. 9 and 10. Comparing the fourth-degree polynomial with the seven-neuron model (36 parameters) for the four-oxide blend, one observes that the E_{tr} and the E_{T} are still very similar, the neural regression results being only marginally better. Fig. 10 shows the results for a five-oxide blend. For six hidden neurons (32 parameters) and third order polynomial, one observes that the neural regression result for E_{tr} is better than the polynomial regression value. The main difference appears in the E_{val}, which is much higher in the polynomial regression. It can be seen in the inset of Fig. 10 that the variance of the performance for higher order polynomials is even larger.

5. Conclusion

We have demonstrated the applicability of neural networks to the prediction of the liquidus temperature of glass-forming oxide blends, due to the ability of neural networks to approximate any non-linear function in a parsimonious fashion. Ternary, 4-oxide and 5-oxide glass databases were used. Neural regression efficiency is expected to increase as compared to other methods when the number of inputs (i.e. the number of components) increases. Since industrial glasses include usually much more than three components, we expect that this application of neural regression methods could be of interest for glass processing problems.

References

[1] A.I. Priven, Glass Phys. Chem. 24 (1998) 67.

[2] M. Cable, in: Materials Science and Technology, in: J. Zarzicki (Ed.), Glasses and Amorphous Materials, vol. 9, VCH, New York, 1991.

[3] A.I. Priven, Glass Phys. Chem. 23 (1997) 33.

[4] S.S. Kim, T.H. Sanders Jr., J. Am. Ceram. Soc. 74 (1991) 1833.

[5] R. Backman, K. Karlsson, M. Cable, N. Pennington, Glastech. Ber. 63K (1990) 460.

[6] R. Backman, K. Karlsson, M. Cable, N. Pennington, Phys. Chem. Glasses 38 (1997) 103.

[7] T. Lakatos, L.G. Johansson, B. Simmingskld, Glastek. Tidsk. 36 (1981) 51;
T. Lakatos, L.G. Johansson, Glastek. Tidsk. 31 (1976) 31.

[8] C.L. Babcock, Silicate Glass Technology Methods, Wiley Interscience, New York, 1977.

[9] Q. Rao, G.F. Piepel, P. Hrma, J.V. Crum;, J. Non-Cryst. Solids 220 (1997) 17.

[10] See for instance: C. Bishop, Neural Networks for Pattern Recognition, Oxford University, 1995.

[11] SciGlass (Glass property information system) version 3.5 Lexington, MA, SciVision, 2000.

[12] K. Hornik, M. Stinchcombe, H. White, P. Auer, Neural Comput. 6 (1994) 1262.

[13] See for instance: S. Haykin, Neural Networks: A Comprehensive Approach, MacMillan, 1994.

[14] R. Zbikowski, K.J. Hunt (Eds.), Neural Adaptive Control Technology, World Scientific, 1995.

[15] Z. Zhang, D. Cui, B. Huang, X. Qin, M. Jiang, J. Mater. Sci. Technol. 16 (2000) 354.

[16] W. Xueye, Q. Guanzhou, W. Dianzuo, Trans. Nonferrous. Met. Soc. China 8 (1998) 505.

[17] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, in: Numerical Recipes C: The Art of Scientific Computing, Cambridge University, 1992.

[18] G. Monari, G. Dreyfus, Neural Comput. 14 (2002) 1481.

[19] NeuroOne software package by Netral S.A.

Further Reading

Appendix A: Liquidus temperatures references (from SciGlass database)

I: Li_2O, SiO_2

O. Asayama, H. Takebe, K. Morinaga, Iron Steel, Inst. J. Int. 33 (1) (1993) 233.

F. Branda, A. Marotta, A. Buri, J. Non-Cryst. Solids 134 (1–2) (1991) 123.

A. Dietzel, H. Wickert, N. Koppen, Glastech. Ber. 27 (5) (1954) 147.

S. Hansel, A. Willgallis, Glastech. Ber. 50 (2) (1977) 45.

P.F. James, Phys. Chem. Glasses 15 (4) (1974) 95; E.G. Rowlands, P.F. James, Phys. Chem. Glasses 20 (1) (1979) 1.

F.C. Kracek, J. Phys. Chem. 34 (12) (1930) 2641.

F.C. Kracek, J. Am. Chem. Soc. 52 (4) (1930) 1436.

J.D. Mackenzie, J.A. Kitchener, Trans. Faraday Soc. 51 (12) (1955) 1734.

M.K. Murthy, F.A. Hummel, J. Am. Ceram. Soc. 38 (2) (1955) 55.

M.K. Murthy, F.A. Hummel, J. Am. Ceram. Soc. 37 (1) (1954) 14.

R. Ota, T. Wakasugi, W. Kawamura, B. Tuchiya, J. Fukunaga, J. Non-Cryst. Solids 188 (1–2) (1995) 136.

R. Ota, J. Fukunaga, in: Advances in Fusion of Glass, American Ceramic Society, Westerville, Ohio, 1988, p. 31.1; R. Ota, J. Fukunaga, Proc. XVth Int. Congr. Glass, Leningrad 1a (1989) 162.

C.S. Ray, H. Wenhai, D.E. Day, J. Am. Ceram. Soc. 70 (8) (1987) 599.

M. Tatsumisago, N. Machida, T. Minami, J. Ceram. Soc. Jpn. 95 (2) (1987) 197; M. Tatsumisago, T. Minami, M. Tanaka, Glastech. Ber. 56K (2) (1983) 945.

II: Al_2O_3, K_2O, SiO_2

J.F. Schairer, N.L. Bowen, Am. J. Sci. 253 (12) (1955) 681.

III: *CaO, K_2O, SiO_2*

G.W. Morey, F.C. Kracek, N.L. Bowen, J. Soc. Glass Technol. 14 (54) (1930) 149.
W.C. Taylor, J. Res. Nat. Bur. Stand. 27 (3) (1941).

IV: *$Al_2O_3, CaO, Na_2O, SiO_2$*

R.J. Araujo, US Patent No. 4439528Cl 3 C 03 C 3/04, 3/10, 3/30, Off. Gazette 1040 (4) (1984).
N.L. Bowen, Am. J. Sci. 35 (210) (1913) 577.
F. Branda, A. Buri, A. Marotta, S. Saiello, Thermochim. Acta 80 (2) (1984) 269; P. Giordano Orsini, A. Buri, A. Marotta, F. Branda, S. Saiello, Riv. Staz. Sper. Vetro 15 (6) (1985) 295.
Corning Glass Works, GB Patent No. 1527785 Cl 2 C 03 C 3/04, Abridg. Specif. 4672 (1978).
P.S. Danielson, US Patent No. 4302250 Cl 3 C 03 C 3/04, 3/10, Off. Gazette 1012 (4) (1981).
R.W. Douglas, Silikattechnik 24 (12) (1973) 405; Z. Strnad, R.W. Douglas, Phys. Chem. Glasses 14 (2) (1973) 33.
W.H. Dumbaugh, R.R. Genisson, M.R. Lestrat, US Patent No. 3978362 Cl 2 H 01 K 1/28, 1/50, 7/00, Off. Gazette, 1976.
W.H. Dumbaugh, US Patent No. 5100452 Cl 5 C 03 C 23/00, Off. Gazette 1136 (5) (1992).
W.H. Dumbaugh, R.R. Genisson, M.R. Lestrat, DE Patent No. 2632690 Cl 2 C 03 C 3/12, 3/30, H 01 K 1/28, Auszuge Offenlegungsschr. Pat. (7) (1977); W.H. Dumbaugh, R.R. Genisson, M.R. Lestrat, FR Patent No. 2320274 Cl 2 C 03 C3/ 04, H 01 K 1/32, Bull. Off. Propr. Industr. (14) (1977).
E.V. Ermolaeva, Ogneupory (4) (1951) 62; E.V. Ermolaeva, Ogneupory (5) (1955) 221.
C.J.R. Gonzalez Oliver, P.F. James, Boll. Soc. Espan. Ceram. Vidrio, Proc. XVI Int. Congr. on Glass, Madrid 31-C (5) (1992) 3.
C.J.R. Gonzalez-Oliver, P.S. Johnson, P.F. James, J. Mater. Sci. 14 (5) (1979) 1159.
O.H. Grauer, E.H. Hamilton, J. Res. Nat. Bur. Stand. 44 (5) (1950) 495.
N. Harry, P.E. Mills, Ceram. Eng. Sci. Proc. 10 (3–4) (1989) 164.
S.P. Jones, Phys. Chem. Glasses 2 (2) (1961) 55.
K.-A. Kumm, H. Scholze, Tonind. Ztg. 93 (9) (1969) 332.
A.J. Milne, J. Soc. Glass Technol. 36 (172) (1952) 275.2.
G.W. Morey, J. Am. Ceram. Soc. 13 (10) (1930) 683.
G.W. Morey, J. Am. Ceram. Soc. 13 (10) (1930) 714.
G.W. Morey, J. Am. Ceram. Soc. 13 (10) (1930) 718.
G.W. Morey, J. Am. Ceram. Soc. 15 (9) (1932) 457.
A.M. Muratov, I.S. Kulikov, Izv. Akad. Nauk SSSR, Metally (4) (1965) 57; A.M. Muratov, I.S. Kulikov, in: Vosstanovlenie i Rafinirovanie Zheleza, Moskva, 1968, p. 60.
O. Yong-Taeg, K. Yoshihara, H. Takebe K. Morinaga, J. Ceram. Soc. Jpn. 105 (12) (1997) 1109.
Y. Oishi, R. Terai, H. Ueda, in: A.R.Cooper, A.H. Heuer (Eds.), Materials Science Research, vol. 9, Mass Transport Phenomena in Ceramics, Plenum, New York, 1975, p. 297.
R. Terai, Rep. Governm. Ind. Res. Inst. Osaka (347) (1975); R. Terai, Y. Oishi, Bull. Gov. Ind. Res. Inst. Osaka 28 (4) (1977) 308; R. Terai, Y. Oishi, Glastech. Ber. 50 (4) (1977) 68.
E.F. Osborn, Am. J. Sci. 240 (11) (1942) 751.
Owens-Illinois Glass Company Gen. Res. Lab., J. Am. Ceram. Soc. 27 (8) (1944) 221.
A.T. Prince, J. Am. Ceram. Soc. 34 (2) (1951) 44.
A. Ram, S.V. Bhatye, K.D. Sharma, Centr. Glass Ceram. Res. Inst. Bull. 2 (4) (1955) 170.
S. Sakka, J.D. Mackenzie, J. Non-Cryst. Solids 6 (2) (1971) 145
C.M. Scarfe, D.J. Cronin, J.T. Wenzel, D.A. Kauffman, Am. Mineral. 68 (11–12) (1983) 1083
J.F. Schairer, N.L. Bowen, Bull. Comm. Geol. Finland 20 (140) (1947) 67.
J.F. Schairer, N.L. Bowen, Am. J. Sci. 254 (3) (1956) 129.
K.A. Shahid, F.P. Glasser, Phys. Chem. Glasses 12 (2) (1971) 50.
W.B. Silverman, J. Am. Ceram. Soc. 22 (11) (1939) 378.
P. Simurka, M. Liska, J. Antalik, Ceram. Silicaty 39 (1) (1995) 9.
L. Spanoudis, US Patent No. 3479217 Cl C 03 C 17/06,3/04, Off. Gazette 868 (3) (1969).
D.A. Weirauch, D.P. Ziegler, J. Am. Ceram. Soc. 79 (4) (1996) 920.

W. Wenjian, J. Zhonghua, D. Zishang, in: Collect. Pap. XIVth Int. Congr. Glass, New Delhi, 1986, vol. 1, p. 379

W.W. Wolf, US Patent No. 3940278 Cl 2 C 03 C 3/04, 13/00, Off. Gazette, 943 (4) (1976)

B. Yale, A.F. Mason, P. Shorrock, N.A. Edwards, GB Patent No. 2256193 Cl 5 C 03 C 3/087, Abstr. Pat. Specif. (49) (1992); B. Yale, A.F. Mason, P. Shorrock, N.A. Edwards, EP Patent No. 0516354 Cl 5 C 03 C 13/00, 3/087, Bull. (49) (1992).

H.S. Yoder, J. Geol. 60 (6) (1952) 586.

V: $Al_2O_3, CaO, Na_2O, MgO, SiO_2$

R.J. Araujo, US Patent No. 4439528 Cl 3 C 03 C 3/04, 3/10, 3/30, Off. Gazette 1040 (4) (1984).

M. Cable, J.W. Smedley, Glass Technol. 28 (2) (1987) 94.

D.R. Cockram, K.M. Fyles, GB Patent No. 2046726 Cl 3 C 03 C 13/00, Abstr. Pat. Specif. (4786) (1980).

P.W.L. Graham, US Patent No. 3473906 Cl C 03 b 25/00, Off. Gazette 867 (3) (1969).

E.F. Grubb, E.C. Hagedorn, J.R. Monks, GB Patent No. 1173758 Cl C 03 C 3/04, Abridg. Specif. (1969).

E.F. Grubb, E.C. Hagedorn, J.R. Monks, US Patent No. 3524738 Cl C 03 b 3/22, 5/16, 29.00, Off. Gazette 877 (3) (1970).

E.F. Grubb, A.W. LaDue, P.W.L. Graham, GB Patent No. 1181581 Cl C 03 C 21/00, Abridg. Specif. (4221) (1970).

F.W. Hammer, US Patent No. 3490885 Cl C 03 C 17/22, Off.Gazette 870 (1970) 3.

F.W. Hammer, J. Jasinski, US Patent No. 3627548 Cl C 03 C 3/34, 5/02, 15/00, Off. Gazette 893 (2) (1971).

N. Harry, P.E. Mills, Ceram. Eng. Sci. Proc. 10 (3–4) (1989) 164.

Owens-Illinois Glass Company Gen. Res. Lab., J. Am. Ceram. Soc. 27 (8) (1944) 221.

C. Ponthieu, D. Petitmaire, D. Jousse, P. Fournier, EP Patent No. 0526272 Cl 5 C 03 C 3/091, G 02 F 1/1333, Bull. (5) (1993).

PPG Industries, Inc., GB Patent No. 1400953 Cl 2 C 03 C 3/04, 21/00, C 03 B 1/00, Abridg. Specif. (4504) (1975).

W.B. Silverman, J. Am. Ceram. Soc. 22 (11) (1939) 378.

W.B. Silverman, J. Am. Ceram. Soc. 23 (9) (1940) 274.

D.R. Stewart, GB Patent No. 1307595 Cl C 03 C 3/04, 21/00, Abridg. Specif. (4378) 1973.

Y. Suzuki, H. Ota, JP Patent No. 61-14096 Cl 4 C 03 C 13/02, C 04 B 14/42, Tokyo, Koho (3-353) (1986).

B. Yale, A.F. Mason, P. Shorrock, N.A. Edwards, GB Patent No. 2256193 Cl 5 C 03 C 3/087, Abstr. Pat. Specif. (49) (1992).

B. Yale, A.F. Mason, P. Shorrock, N.A. Edwards, EP Patent No. 0516354 Cl 5 C 03 C 13/00, 3/087, Bull. (49) (1992).