

1N-24
169424
P20

Radial Basis Function Network Learns Ceramic Processing and Predicts Related Strength and Density

Krzysztof J. Cios
University of Toledo
Toledo, Ohio

George Y. Baaklini and Alex Vary
Lewis Research Center
Cleveland, Ohio

and

Robert E. Tjia
University of Toledo
Toledo, Ohio

May 1993

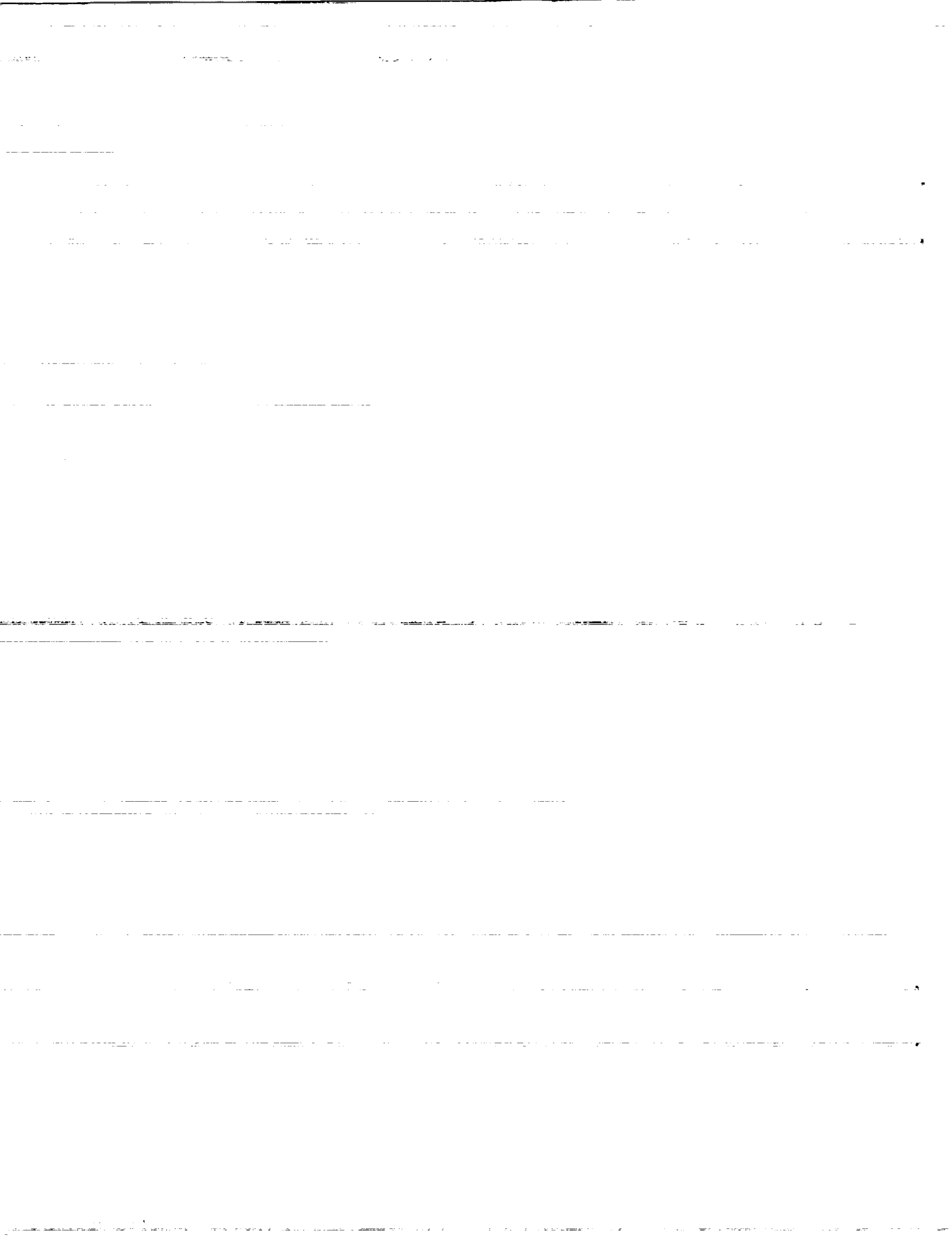
(NASA-TM-106048) RADIAL BASIS
FUNCTION NETWORK LEARNS CERAMIC
PROCESSING AND PREDICTS RELATED
STRENGTH AND DENSITY (NASA) 20 p

N93-27129

Unclass

NASA

G3/24 0169424



RADIAL BASIS FUNCTION NETWORK LEARNS CERAMIC PROCESSING AND PREDICTS RELATED STRENGTH AND DENSITY

Krzysztof J. Cios*
University of Toledo
Toledo, Ohio 43606

George Y. Baaklini and Alex Vary
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

and

Robert E. Tjia
University of Toledo
Toledo, Ohio 43606

ABSTRACT

Radial basis function (RBF) neural networks were trained using the data from 273 Si_3N_4 modulus of rupture (MOR) bars which were tested at room temperature and 135 MOR bars which were tested at 1370 °C. Milling time, sintering time, and sintering gas pressure were the processing parameters used as the input features. Flexural strength and density were the outputs by which the RBF networks were assessed. The "nodes-at-data-points" method was used to set the hidden layer centers and output layer training used the gradient descent method. The RBF network predicted strength with an average error of less than 12% and density with an average error of less than 2%. Further, the RBF network demonstrated a potential for optimizing and accelerating the development and processing of emerging ceramic materials.

INTRODUCTION

Ceramics such as silicon nitride (Si_3N_4) are under investigation as a candidate material for heat engine applications due to their high operating temperatures, reduced weight, resistance to oxidization, and thermal shock resistance [1]. The major drawback currently encountered with this type of ceramic is its widely varying strength and low fracture toughness, which occur due to discrete defects introduced into the material during processing [1, 2, 3]. In their work, Sanders and Baaklini [3], were concerned with the

* On sabbatical leave from the University of Toledo and NASA Resident Research Associate at Lewis Research Center.

problem of designing a silicon nitride ceramic with the goal of achieving fully dense material that possesses high strength with the lowest amount of scatter. In the process of manufacturing they tried to optimize several variables such as milling time, sintering temperature, sintering time, nitrogen pressure and setter contact. In addition, they investigated the effect of sintering and temperature variations and whether wet powder sieving was superior to dry sieving. Hence, in their work, they were trying to optimize the manufacturing process by using sound engineering judgement coupled with trial and error methodology.

In our work we are interested in finding whether it is possible to utilize neural networks to help in the process design of ceramics. Neural networks excel in function approximation making it easy to identify variables that contribute most toward a desired output parameter, say strength, from a few trials. This should help in speeding up process modelling for new materials. Designers can usually comprehend the combined effects of a few variables but it becomes very difficult to do so for a large number of variables. From the data collected by Sanders and Baaklini [3], we selected three input variables, namely, the milling time of the $\text{Si}_3\text{N}_4\text{-SiO}_2\text{-Y}_2\text{O}_3$ powder, the sintering time, and the nitrogen pressure employed during sintering of the modulus of rupture (MOR) test bars. From the output variables we selected flexural strength and density. The rationale for using the above mentioned variables is that there were not enough training pairs (outputs associated with inputs) for processing variables such as temperature and sieving. It should be noted that the available data was not originally obtained nor intended for neural network analysis. However, we expected that an RBF network would give reasonably accurate predictions despite the fact that the data points are unevenly distributed in the input space.

In this paper we attempt to find the effects of milling time, sintering time and nitrogen pressure on resultant strength and density with the aid of a neural network. We make use of the data obtained from the previous study [3] for training and testing the neural network. The original data had exhibited MOR test bar strength and density variations for different combinations of milling times, sintering times, nitrogen pressure, powder wet sieving, etc. Thus, the data set used in this study is based on 273 MOR bars tested at room temperature and 135 MOR bars tested at 1370 °C. Therefore, the purpose of this study is to determine how effectively a neural network can be trained to predict the resultant strength and density of a batch of MOR bars.

RADIAL BASIS FUNCTION NETWORKS

One of the common uses of feedforward neural networks is the approximation of complex, non-linear functions. Theoretically, a neural network can be made to approximate any given function provided that the network has a sufficient number of processing elements (nodes). The traditional backpropagation network has been shown to be successful in this area. However, its major disadvantage is the fact that the iterative gradient descent method it employs to optimize the weights is computationally demanding and slow and results in long training times.

A three layer network with "locally-tuned" processing units in the hidden layer has been proposed as an alternative to backpropagation [4]. Also known as the radial basis function (RBF) network, this type of network requires less training time because the approach uses a combination of self-organization and supervised learning. The network is considered as self-organized because the hidden layer nodes are RBF nodes centered at the training data points (or some subset of it) and each node only responds to an input which is close to its center. The output layer nodes are usually linear or sigmoidal functions and their weights may be obtained using some form of supervised learning method, such as an iterative error reduction scheme, similar to that used in backpropagation. In the case of linear outputs, direct approaches involving matrix inversion can be used in place of the slower iterative methods.

Description of the RBF Network

Figure 1 shows a general RBF network with n inputs and one linear output. This network performs a mapping $f: \mathbf{R}^n \rightarrow \mathbf{R}$ given by the following equation [5] :

$$f(\mathbf{x}) = \lambda_0 + \sum_{i=1}^{n_r} \lambda_i \varphi(\|\mathbf{x} - \mathbf{c}_i\|) \quad (1)$$

where $\mathbf{x} \in \mathbf{R}^n$ is the input vector, $\varphi(\cdot)$ is a function from $\mathbf{R}^n \rightarrow \mathbf{R}$, $\|\cdot\|$ denotes the Euclidean norm, λ_i ($0 \leq i \leq n_r$) are the weights of the output node, \mathbf{c}_i ($0 \leq i \leq n_r$) are the RBF centers, and n_r is the number of centers. As a variation of the linear output,

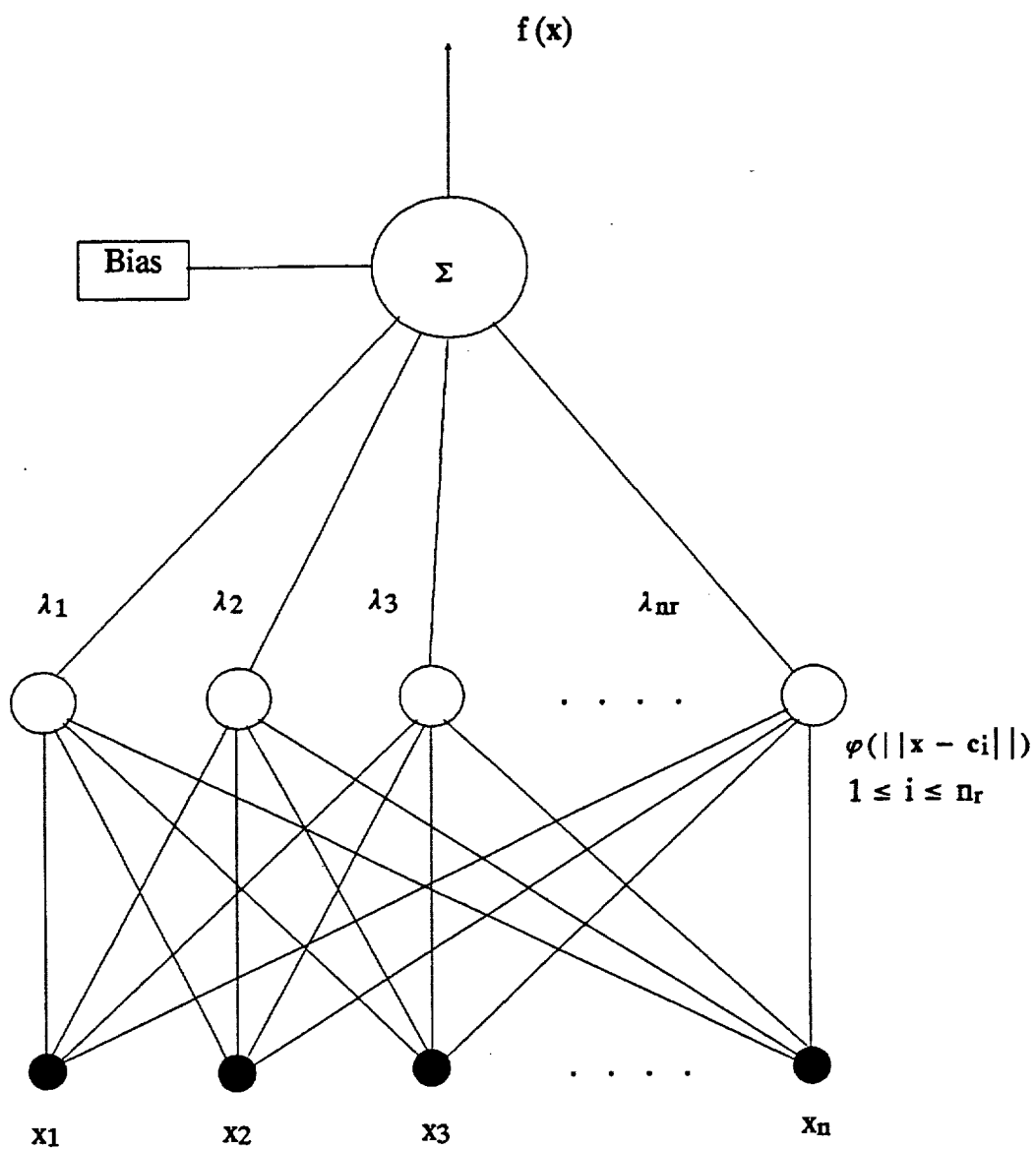


Figure 1: Single linear output radial basis function network

the output node may be given the sigmoidal function, if required. In this case, the mapping function would be:

$$f(x) = \frac{1.0}{1.0 + e^{-[\lambda_0 + \sum_{i=1}^{n_r} \lambda_i \varphi(||x-c_i||)]}} \quad (2)$$

Studies have shown [5] that the choice of the nonlinear function $\varphi(.)$ is not crucial to the overall performance of the network. One of the more common functions used for $\varphi(.)$ is the Gaussian function:

$$\varphi(||x-c_i||) = \exp\left(-\frac{||x-c_i||^2}{\sigma_i^2}\right) \quad (3)$$

where σ_i is a constant which determines the width of the input space of the i -th node. A heuristic method to determine the value of σ_i will be described later. It is obvious that this function has a maximum value of 1 when $||x-c_i||$ is 0, and this value drops off to 0 as $||x-c_i||$ approaches infinity. Other functions may be used in place of the Gaussian function, such as the thin-plate spline function [2].

Setting the Hidden Layer Parameters

The centers of the RBF functions c_i are usually chosen from the training data points x_t ($1 \leq t \leq N$). This method, known as the "nodes at data points" method [6], is suitable for small to medium sized training data sets. For larger data sets, it is not practical to have an RBF center at each data point as the network would become too big. Some of the methods used for reducing of the number of RBF centers are: the random selection of centers, clustering of data points, and orthogonal least squares reduction [5].

The random selection method simply uses a random selection of n_r centers from N data points, where $n_r < N$. While this method is simple, it has its drawbacks. First of all, the data points in the training set might not be evenly distributed over the input space. The concentration of data points in some regions may be sparse to begin with, and random selection might end up with some regions having too few data points or even none at all. Second, the desired output of the mapping function might change drastically over certain regions of the input space, while at the same time it may remain relatively constant over other regions. We intuitively know that a larger concentration of RBF centers would be

required in regions where the function changes rapidly, and that in regions where the function changes little a lesser number of centers would be required. Unfortunately, the random selection method ignores this fact.

If the desired outputs are discrete and represent, say, C different classes, then clustering methods [3] may be used to cluster the data points within each class. Algorithms such as k-means clustering may be used on each class of data, and the resulting cluster centers (prototype vectors) are used as the RBF centers. If k_i ($1 \leq i \leq C$) are the number of clusters in each class, then the number of RBF centers, n_r , would be:

$$n_r = \sum_{i=1}^C k_i \quad (4)$$

Although clustering methods result in a choice of centers which cover the input space evenly, it still does not take into account the portions of the input space where the function changes rapidly and thus require a higher concentration of centers. This is true, for example, in regions at or near class boundaries.

For the case of linear output nodes, one very effective method of choosing a set of RBF centers from the training data set is the orthogonal least squares (OLS) reduction method. The OLS reduction method, which is described in [5], enables the selection of the most significant RBF centers from a given training data set. The OLS reduction algorithm is an extension of the OLS learning method of which a very general description is given next.

Output layer training - Orthogonal Least Squares Learning

The OLS learning algorithm is best explained by viewing the RBF network as a linear regression model [5]. For the case of a single node output, this will be:

$$d(t) = \sum_{i=1}^M p_i(t)\theta_i + \varepsilon(t) \quad (5)$$

where $d(t)$ is the desired output, θ_i are the parameters corresponding to the weights λ_i in Figure 1, $\varepsilon(t)$ is the error signal, and $p_i(t)$ are the regressors given by:

$$p_i(t) = p_i(\mathbf{x}(t)) = \varphi(\|\mathbf{x}(t) - \mathbf{c}_i\|) \quad (6)$$

Each nonlinearity $\varphi(\cdot)$ with its center c_i corresponds to a $p_i(t)$. The mapping of the regressors to the output space can then be represented in the matrix form:

$$\mathbf{d} = \mathbf{P}\boldsymbol{\Theta} + \mathbf{E} \quad (7)$$

where

$$\mathbf{d} = [d(1) \dots d(N)]^T$$

$$\mathbf{P} = [\mathbf{p}_1 \dots \mathbf{p}_M], \quad \mathbf{p}_i = [p_i(1) \dots p_i(N)]^T, \quad 1 \leq i \leq M$$

$$\boldsymbol{\Theta} = [\theta_1 \dots \theta_M]^T$$

$$\mathbf{E} = [\varepsilon(1) \dots \varepsilon(N)]^T$$

The matrix \mathbf{P} can then be decomposed into:

$$\mathbf{P} = \mathbf{W}\mathbf{A} \quad (8)$$

where \mathbf{W} is a set of orthogonal vectors spanning the identical space spanned by \mathbf{P} , and \mathbf{A} is an upper triangular matrix. \mathbf{W} and \mathbf{A} may be obtained in several ways, such as by using the Gram-Schmidt method [7]. Equation (7) can thus be rewritten as:

$$\mathbf{d} = \mathbf{W}\mathbf{g} + \mathbf{E} \quad (9)$$

The orthogonal least squares solution $\hat{\mathbf{g}}$ which minimizes \mathbf{E} is given by:

$$\hat{\mathbf{g}} = \mathbf{H}^{-1}\mathbf{W}^T\mathbf{d} \quad (10)$$

where \mathbf{H} is a diagonal matrix with elements $h_{ii} = \mathbf{w}_i^T \mathbf{w}_i$. Then the solution for the parameters $\hat{\boldsymbol{\Theta}}$ can be found from the following relation:

$$\hat{\boldsymbol{\Theta}} = \mathbf{A}^{-1}\hat{\mathbf{g}} \quad (11)$$

We then set each λ_i to θ_i and the network is complete.

Although the method just described is for an RBF network with one output node, solving for the weights with several output nodes simply requires that the process be repeated several times. The weights for each output node may be solved separately, one node at a time, without regard for the other output nodes. This is so because unlike backpropagation, the previous layer parameters are already determined and, hence, are unaffected by changes in output layer weights.

Output layer training - Gradient Descent (Delta Rule)

When the output node has a non-linear function, it usually is not possible to use direct approaches, like the one described above, to obtain the values for the weights λ_i which would give the least error over the entire training data set. Hence, for output nodes with transfer functions such as the sigmoidal or hyperbolic-tangent functions, a gradient descent method has to be used instead. Given an output node with a transfer function $g()$, the mapping function can be expressed as:

$$f(\mathbf{x}) = g(\varphi_1(\mathbf{x}), \dots, \varphi_M(\mathbf{x}), \lambda_1, \dots, \lambda_M) \quad (12)$$

For a given input \mathbf{x} , the raw error is merely the difference between the desired output, d , and the network output, $f(\mathbf{x})$, which is simply $(d - f(\mathbf{x}))$. To ensure that learning is biased towards those nodes that can make more significant contributions towards reducing the current error, gradient descent algorithms make use of the scaled error rather than the raw error, which is given by:

$$\varepsilon = (d - f(\mathbf{x})) * g'(\varphi_1(\mathbf{x}), \dots, \varphi_M(\mathbf{x}), \lambda_1, \dots, \lambda_M) \quad (13)$$

Having obtained the scaled error, each weight may be incrementally updated by a small amount $\Delta \lambda_i$ (hence the name "delta rule") in an attempt to reduce the error:

$$\Delta \lambda_i = \text{Learning_Coefficient} * \varepsilon * \lambda_i \quad (14)$$

This process is repeated iteratively for all values of \mathbf{x} in the training set until the value of some global error function has been reduced to an acceptable level. One function that may be used is the mean squared error (MSE) function:

$$\text{Mean Squared Error} = \frac{1}{K} \sum_{k=1}^K (d_k - f(\mathbf{x}_k))^2 \quad (15)$$

or the RMS error, which is the square root of the MSE.

As with the OLS learning method, this method can be easily extended for networks with more than one output node. As with backpropagation techniques, several improvements to the method may be added. Using a momentum term will speed up the learning process and incorporating simulated annealing [8] will reduce the likelihood of the network becoming trapped in a local minima. Although the gradient descent method is slower in training the output layer of RBF networks than the OLS method, it is still much quicker than backpropagation since the weights that are being updated are only the output layer weights λ_i .

APPROACH AND DATA FORMATTING

In order to validate the results in terms of confidence which can be associated with predictions of new untried combinations of input parameters, we calculate errors for the following steps.

Firstly, the maximum-strength value batch is removed from the data and used as the test data. Next, 70%, and later 60%, of the data is reserved for training, and the remaining data for testing. The 60% size of training data gives indication as to how much processing information is required to make accurate predictions.

Secondly, several combinations of the three input parameters are used to determine whether a material having equal or higher values of flexural strength and density can be obtained. Although the first experiment validates to some degree the results obtained from the second, it needs to be confirmed by real experiments to find whether the new combinations of variables agree with actual material strength and density. Comments on the validity of the resultant predictions are given in the Discussion section of the paper.

For the room temperature, 18 different combinations of milling time, sintering time, and nitrogen pressure yield the composition strengths and densities listed in Table I. Also listed in Table I are the strengths and densities for 9 combinations at 1370 °C.

Table I: Strength/Density at Room Temperature and 1370 °C for different Processing/Sintering conditions

Room Temperature Batch no.	No. of Specimens	Milling Time, hr	Sintering Time, hr	Nitrogen Pressure, MPa	Actual Strength, MPa	Actual Density, g/cm ³
6Y1B	30	24	1	2.5	556	3.12
6Y2B	30	24	1	2.5	532	3.18
6Y11	15	100	1	2.5	490	3.23
6Y12	15	300	1	2.5	579	3.25
6Y13	15	100	1	2.5	684	3.24
6Y14	14	300	1	2.5	746	3.24
6Y15, 6Y16	19	24	2	5	664	3.22
6Y17	10	100	2	5	646	3.23
6Y18	10	100	1.5	5	608	3.21
6Y19	10	100	1.5	5	570	3.22
6Y20	10	100	2	5	650	3.22
6Y23	15	100	1.25	5	631	3.24
6Y24A	15	100	1.25	3.5	586	3.26
6Y24B	15	100	2	3.5	619	3.26
6Y25	10	300	2	5	714	3.28
6Y26A	15	100	1	3.5	479	3.20
6Y26B	15	100	1	5	503	3.18
6Y28	10	100	2	5	671	3.21
1370 °C Batch no.						
6Y9B	29	24	1	2.5	382	3.12
6Y11	13	100	1	2.5	445	3.23
6Y12	14	300	1	2.5	417	3.25
6Y13	15	100	1	2.5	405	3.24
6Y14	14	300	1	2.5	424	3.24
6Y15, 6Y16	20	24	2	5	402	3.22
6Y17	10	100	2	5	441	3.23
6Y18	10	100	1.5	5	460	3.21
6Y25	10	300	2	5	467	3.28

In order to determine the validity of the network predictions, it is necessary to test the network using known test vectors and then calculate the error of the predictions. Of particular interest is the ability of the network to predict the output values for batch number 6Y25, as this batch number represents the optimum combination for the processing variables from the available data set.

Batch number 6Y25 was first removed from the data sets. The data sets were then pseudo-randomly divided into a ratio of approximately 70 % training to 30 % testing. Batch

number 6Y25 was then inserted into the test data set. This was repeated for 5 times in order to have 5 different pairs of training and test data sets which were labeled as combinations A through E (Table II). This entire process was then repeated using a ratio of approximately 60 % training to 40 % testing.

Table II: Selected batch numbers for 70% and 60% training sets A through E

Room Temp. Batch no.	Training Sets 70%					Training Sets 60%				
	A	B	C	D	E	A	B	C	D	E
6Y1B	*		*	*	*			*	*	*
6Y2B		*		*	*				*	*
6Y11	*	*	*		*	*	*			*
6Y12		*	*	*	*		*	*	*	*
6Y13	*	*		*		*				
6Y14	*	*	*	*		*	*	*		
6Y15	*		*		*					*
6Y16										
6Y17		*	*	*	*		*	*	*	*
6Y18		*		*			*		*	
6Y19	*		*	*		*		*	*	
6Y20	*	*	*		*	*	*	*		*
6Y23	*	*	*	*	*	*	*	*	*	*
6Y24A			*	*	*			*	*	*
6Y24B	*		*		*	*		*		
6Y25										
6Y26A	*	*	*	*		*	*	*	*	
6Y26B	*	*			*	*	*			*
6Y28	*	*		*	*	*	*		*	*
1370°C Batch no.										
6Y9B	*	*		*	*	*			*	*
6Y11		*	*	*	*		*	*	*	
6Y12	*	*	*		*	*	*			*
6Y13	*		*	*	*		*	*	*	*
6Y14	*	*			*	*				*
6Y15		*	*	*			*	*		
6Y16										
6Y17	*	*	*	*		*	*	*	*	
6Y18	*		*	*	*	*		*	*	*
6Y25										

Next, a training data set consisting of all the batch numbers (100 %) except 6Y25 was created. Batch number 6Y25 was placed in the test data set as the sole vector. Finally, all the batch numbers were placed in a training data set and the test data set was constructed using vectors for which we do not know the outputs in order to demonstrate the capability of the RBF network in material process optimization. This gives us a total of 12 pairs of training and test data sets for room temperature tested materials, and another 12 for materials tested at 1370 °C.

RESULTS

The RBF networks were trained using different training sets described above. The "nodes at data points" method were used to set up the hidden layer. The gradient descent (delta rule) method was used to train the output layer nodes, which used the sigmoidal function. The RBF networks used consisted of three input nodes and two output nodes. The number of nodes in the hidden layer ranged from 5 to 18, depending upon the number of training vectors in the data set. Tables III and VIII show the detailed results for the 70% training and 30% test data set, for combination (A). The overall results for combinations A through E are shown in Table IV for 70% training, and in Table V for 60% training. Table VI shows the results obtained to predict 6Y25 strength and density using 100% of the data. Table VII shows predictions made for selected sets of processing and sintering variables that resulted in strengths and densities similar to that of the optimum batch 6Y25.

Table III: Predicted room temperature strength with 70% training, Combination A

Batch Number	Actual Strength, MPa	Predicted Strength, MPa	% Error	Actual Density, g/cm ³	Predicted Density, g/cm ³	% Error
6Y2B	556	544	2.26	3.18	3.17	0.46
6Y12	579	752	29.84	3.25	3.24	0.27
6Y17	646	660	2.13	3.23	3.21	0.49
6Y18	608	616	1.37	3.21	3.24	0.91
6Y24A	586	507	13.51	3.26	3.23	0.88
6Y25	714	681	4.85	3.28	3.21	2.28
Average Error			8.95			0.88

Table IV: Overall results for room temperature, 70% training

Combination	Strength - average % error for all test vectors	Strength - % error for 6Y25	Density - average % error for all test vectors	Density - % error for 6Y25
A	8.95	4.85	0.88	2.28
B	7.84	12.95	1.41	2.86
C	10.78	3.67	0.87	2.30
D	10.21	11.90	0.73	1.89
E	15.63	17.74	1.07	3.04
Combined Average % error	10.54	10.17	0.98	2.50

Table V: Overall results for room temperature, 60% training

Combination	Strength - average % error for all test vectors	Strength - % error for 6Y25	Density - average % error for all test vectors	Density - % error for 6Y25
A	9.50	0.04	1.09	1.73
B	12.24	8.32	1.02	1.49
C	8.72	5.26	1.03	1.61
D	11.31	6.88	0.85	1.14
E	15.41	9.30	1.16	2.71
Combined Average % error	11.34	5.96	1.03	1.74

Table VI: Prediction for 6Y25 density and strength at room temperature, 100% training

Batch Number	Actual Strength, MPa	Predicted Strength, MPa	% Error	Actual Density, g/cm ³	Predicted Density, g/cm ³	% Error
6Y25	714	614	13.99	3.28	3.19	2.67

Table VII: Prediction of selected processing and sintering variables for optimum room temperature strength and density, 100 % plus 6Y25 training

Milling Time, hr	Sintering Time, hr	Nitrogen Pressure, MPa	Predicted Strength, MPa	Predicted Density, g/cm ³
150	1.5	3	692	3.28
175	1.5	3	700	3.28
200	1.5	3	706	3.28
200	1.75	4	689	3.27
250	1.5	3	709	3.28
250	1.5	4	705	3.28
250	1.75	4	705	3.28
300	1.5	4	711	3.28
300	1.75	4	713	3.28
300	2	5	712	3.28

Tables VIII-XII show the results obtained for 1370 °C.

Table VIII: Predicted strength at 1370 °C with 70% training, Combination A

Batch Number	Actual Strength, MPa	Predicted Strength, MPa	% Error	Actual Density, g/cm ³	Predicted Density, g/cm ³	% Error
6Y11	445	399	10.32	3.23	3.21	0.56
6Y15 6Y16	402	442	10.16	3.22	3.20	0.64
6Y25	467	440	5.8	3.28	3.24	1.28
Average Error			8.77			0.83

Table IX: Overall results for 1370 °C, 70% training

Combination	Strength - average % error for all test vectors	Strength - % error for 6Y25	Density - average % error for all test vectors	Density - % error for 6Y25
A	8.77	5.80	0.83	1.28
B	7.61	11.88	1.50	2.71
C	7.22	11.17	1.69	1.14
D	10.36	16.45	1.62	2.34
E	6.69	3.80	1.52	2.82
Combined Average % error	8.21	9.82	1.43	2.06

Table X: Overall results for 1370 °C, 60% training

Combination	Strength - average % error for all test vectors	Strength - % error for 6Y25	Density - average % error for all test vectors	Density - % error for 6Y25
A	7.19	3.40	1.56	0.96
B	10.78	12.13	1.21	0.60
C	7.53	14.70	1.30	1.23
D	8.96	17.82	2.83	3.64
E	8.07	1.50	1.71	3.03
Combined Average % error	8.52	9.91	1.72	1.89

Table XI: Prediction for 6Y25 density and strength at 1370 °C with 100% training

Batch Number	Actual Strength, MPa	Predicted Strength, MPa	% Error	Actual Density, g/cm ³	Predicted Density, g/cm ³	% Error
6Y25	467	402	13.83	3.28	3.20	2.46

Table XII: Prediction of selected processing and sintering variables for optimum density and strength at 1370 °C with 100 % plus 6Y25 training

Milling Time	Sintering Time	Nitrogen Pressure	Predicted Strength	Predicted Density
150	1.5	4	466	3.24
175	1.5	4	469	3.25
200	1.5	4	470	3.26
200	1.5	5	471	3.25
200	1.75	5	471	3.27
250	2	5	467	3.27
300	1.5	4	468	3.27
300	1.5	5	470	3.26
300	1.75	5	471	3.27
300	2	5	467	3.27

Using 60% of the room temperature data for training, the strength and density values were predicted with an average percentage error of less than 11.4% and 1.1%, respectively. When the slightly larger training set of 70% was used, the average percentage errors for strength and density either remained the same or dropped slightly to less than 10.6% and 1.0%, respectively. Similar results were obtained for the 1370 °C data. With 60% training the average percentage errors for strength and density were less than 8.6% and 1.8%, respectively. With 70% training these values were 8.3% and 1.5%, respectively.

DISCUSSION

Relatively large errors occurred in several cases. In Table III, the error of 29.84% on the predicted strength can be explained by the fact that the training vector from batch 6Y14 biased the results of 6Y12 and this was totally due to a sintering variable that was not included as an input feature. In Table IV, the error of 17.74% on the predicted strength was due to the bias in the training set which incorporated a majority of training vectors with 24 hours grinding time. In Tables VI and XI, the 13.99% and 13.83% errors with 100% training can be attributed to biased regions and sharp gradients in the data set; many of the training vectors are concentrated within regions of the input hyperspace which correspond to shorter grinding times. In Table IX, the 16.45% error for combination "D" can be attributed to the absence of training vectors with 300 hours grinding time. Similarly, in Table X, the errors of 14.7% and 17.82% in combinations "C" and "D", respectively, can be attributed to the absence of training vectors with 300 hours grinding time, whereas the other cases performed well because they had at least one such vector.

Bias in the training sets may also result in a very good prediction. In Table V, for instance, the result of an error of only 0.04% for 6Y25 was obtained because the training data set did not have any vectors having grinding times of 24 hours, meaning that most of the training vectors are relatively close to 6Y25 in the input hyperspace.

The information in Tables VII and XII suggest that there may be other combinations of sintering and processing variables that will produce results almost as good as that obtained for 6Y25 but more efficiently. For example, in Table VII, using a milling time of 250 hours, a sintering time of 1.5 hours, and a nitrogen pressure of 3 MPa, the network predicts that a strength of 709 MPa can be obtained. This is only slightly less than the 6Y25 value of 712 MPa, but with a reduction in milling time of 50 hours.

Similarly, Table XII indicates that a slightly higher than optimal for 6Y25 value of 471 MPa can be achieved with milling time of 200 hours, sintering time of 1.5 hours, and nitrogen pressure of 5 MPa, which is a 100 hours saving in milling time over 6Y25. A word of caution here. Although the confidence in prediction results for strength and density lies within 12% and 2%, respectively, these predictions need to be confirmed by manufacturing of ceramics using the same input parameters. From the theoretical point of view, if there is a steady trend in data, namely, if increase in the value of one input variable leads to an increase (or decrease) in value of the output parameter, than RBF predictions will be very

accurate and valid. In other words, if the function between the input and output variables is smooth and either increasing or decreasing the RBF network, and other neural networks as well, will make valid predictions from a sufficiently large training data.

Using even the smaller training data set of 60% did not increase the prediction errors in a significant way. This suggests a potential for speeding up the optimization of processing by using neural networks. Apparently, correlations between the input and desired output variables can be established by diminished training sets when using significant input variables.

In this study we have used only a small subset of input and output variables, and still the results achieved were quite reasonable. If larger number of input and output variables could be used that would certainly improve the predictions and their reliability.

CONCLUSIONS

The radial basis function (RBF) network was found to be applicable for learning silicon nitride processing and consequently predicting strength and density using three processing variables as input features. Predicting strength and density values for the 30% or 40% of the modulus of rupture batches subsets which were not used for training was successful with an average error of less than 12% for strength and 2% for density for both room and high temperatures. Predicting strength for the optimum batch was only successful (less than 12% error) where the training set reflected a reduced gradient and less biased regions. Predicting bulk density was more successful than predicting strength. This was due to the fact that bulk density is directly related to milling time, sintering time and pressure, whereas the flexural strength is additionally dependent on pore morphology, microstructure, and the presence of failure causing defects. This work shows that RBF neural networks have a potential for accelerating improvements in ceramic materials processing.

REFERENCES

- [1] Klima, S., Baaklini, G.: Nondestructive Characterization of Structural Ceramics. SAMPE Quarterly, Vol. 17, No. 3, 1986, pp. 13-19.
- [2] Baaklini, G.: NDE Reliability and Process Control for Structural Ceramics. Journal of Engineering for Gas Turbines and Power, Vol. 109, 1987, pp. 263-266.
- [3] Sanders, W.A., Baaklini, G.Y.: Correlation of Processing and Sintering Variables with the Strength and Radiography of Silicon Nitride. Ceramic Eng. and Sci. Procs., Vol. 7, No. 7-8, July-August, 1986, pp. 839-859.
- [4] Moody, J., Darken, C.: Fast-learning in Networks of Locally-tuned Processing Units. Neural Computing, Vol. 1, No. 2, 1989, pp. 281-294.
- [5] Chen, S., Cowan, C.F.N., Grant, P.M.: Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. IEE Trans. on Neural Nets., Vol. 2, No. 2, 1991, pp 302-309.
- [6] Zahirhiak, D.R., Chapman, R., Rogers, S.K., Suter, B.W., Kabrisky, M., Pyati, V.: Pattern Recognition using Radial Basis Function Networks. Conf. Aerospace Appl. of AI, Dayton, 1990, in press.
- [7] Strang, G.: Linear Algebra and its Applications. Harcourt Brace Jovanovich Publ., 1988, pp. 172-174.
- [8] Szu, H., Hartley, R.: Fast Simulated Annealing. Phys. Lett. A, Vol. 122. No. 8, 1987, pp. 157-162.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE May 1993		3. REPORT TYPE AND DATES COVERED Technical Memorandum
4. TITLE AND SUBTITLE Radial Basis Function Network Learns Ceramic Processing and Predicts Related Strength and Density			5. FUNDING NUMBERS WU-505-63-1M	
6. AUTHOR(S) Krzysztof J. Cios, George Y. Baaklini, Alex Vary, and Robert E. Tjia				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191			8. PERFORMING ORGANIZATION REPORT NUMBER E-7795	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, D.C. 20546-0001			10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA TM-106048	
11. SUPPLEMENTARY NOTES Krzysztof J. Cios, University of Toledo, Toledo, Ohio 43606 (on sabbatical leave from the University of Toledo and NASA Resident Research Associate at Lewis Research Center). George Y. Baaklini and Alex Vary, NASA Lewis Research Center; and Robert E. Tjia, University of Toledo, Toledo, Ohio 43606. Responsible person, George Y. Baaklini, (216) 433-6016.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Categories 24 and 38			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Radial basis function (RBF) neural networks were trained using the data from 273 Si ₃ N ₄ modulus of rupture (MOR) bars which were tested at room temperature and 135 MOR bars which were tested at 1370 °C. Milling time, sintering time, and sintering gas pressure were the processing parameters used as the input features. Flexural strength and density were the outputs by which the RBF networks were assessed. The "nodes-at-data-points" method was used to set the hidden layer centers and output layer training used the gradient descent method. The RBF network predicted strength with an average error of less than 12% and density with an average error of less than 2%. Further, the RBF network demonstrated a potential for optimizing and accelerating the development and processing of ceramic materials.				
14. SUBJECT TERMS Ceramics; Processing; High temperature materials; Neural networks; RBF network; Silicon nitride; Monolithic ceramics			15. NUMBER OF PAGES 20	
			16. PRICE CODE A03	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	

