Available online at www.sciencedirect.com
SCIENCE DIRECT®
Comput. Methods Appl. Mech. Engrg. 192 (2003) 2249–2267
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Truncated-Newton training algorithm for neurocomputational viscoplastic model

M.S. Al-Haik [a], H. Garmestani [b], I.M. Navon [c,*]

[a] Department of Mechanical Engineering, FAMU-FSU College of Engineering, Florida State University, Tallahassee, FL 32310, USA
[b] Materials Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
[c] Department of Mathematics and School of Computational Science and Information Technology, Florida State University, Tallahassee, FL 32306, USA

Received 16 August 2002; received in revised form 24 February 2003; accepted 19 March 2003

Abstract

We present an estimate approach to compute the viscoplastic behavior of a polymer matrix composite under different thermomechanical environments. This investigation incorporates computational neural network as the tool for determining the creep behavior of the composite. We propose a new second-order learning algorithm for training the multilayer networks. Training in the neural network is generally specified as the minimization of an appropriate error function with respect to parameters of the network (weights and learning rates) corresponding to excitory and inhibitory connections. We propose here a technique for error minimization based on the use of the truncated Newton (TN) large-scale unconstrained minimization technique with quadratic convergence rate. This technique offers a more sophisticated use of the gradient information compared to simple steepest descent or conjugate gradient methods. In this work we briefly specify the necessary details for implementing the TN method for training the neural networks that predicts the viscoplastic behavior of the polymeric composite. We provide comparative experimental results and explicit model results to verify the effectiveness of the neural networks-based model. These results verify the superiority of the present approach compared to the explicit modeling scheme. Moreover, the present study demonstrates for the first time the feasibility of introducing the TN method, with quadratic convergence rate, to the field of neural networks.

Keywords: Neural networks; Viscoplasticity; Optimization creep; Truncated Newton method

1. Introduction

Structural polymer matrix composites (PMCs) possessing superior strength-to-weight and modulus-to-weight ratios are being considered as alternative materials for structural applications [10]. As PMCs continue

* Corresponding author. Address: Department of Mathematics, Florida State University, Supercomputer Computations Research Institute, Dirac Science Library Building #415, Tallahassee, FL 32306-4052, USA. Tel.: +1-850-644-6560; fax: +1-850-644-0098.
E-mail addresses: navon@csit.fsu.edu, navon@scri.fsu.edu (I.M. Navon).

0045-7825/03/$ - see front matter © 2003 Published by Elsevier Science B.V.
doi:10.1016/S0045-7825(03)00261-5

to find applications in critical structural components, accurate constitutive modeling becomes increasingly important. Constitutive modeling of polymeric composite materials presents a difficult and distinct challenge.

While significant progress has been made in constructing models applicable for small strain and limited strain-rate and temperature regimes, much less progress has been made for more general conditions. Polymers behave very differently depending on the loading conditions to which they are subjected, from brittle to viscoelastic, to viscous (fluid-like). Another impediment to modeling polymeric behavior is that the mechanisms of deformation in polymers are distinct from those in metals, for which extensive modeling work has been done. Consequently, well-established constitutive models and concepts for metals are not directly transferable to modeling polymeric composites.

Polymeric materials behavior has been repeatedly demonstrated to be very temperature and rate dependent [46]. The creep response of carbon fiber-PMC, compounds the inherent time-dependent behavior of the polymeric phase and the temporal behavior of carbon fibers due to fiber-matrix debonding, interlayer delamination and microcracking. Since all the later damage mechanisms are also time-dependent and occur co jointly with polymeric creep, it is extremely difficult to separate out the individual contributions of the composite constituents.

The creep response of polymers within the linear range of behavior was modeled with a high degree of success by linear viscoelasticity theory [5,43]. In spite of steady but slow progress over the last 30 years, the understanding and modeling of nonlinear behavior of polymers is still the subject of ongoing research [39,44]. Beyond the linear range of behavior, progress in the understanding and modeling of creep in polymeric composites seems to be hindered as much by the absence of a comprehensive database as by inadequacies and controversies in the mathematical and basic mechanics formulation.

While there is a vast literature related to the modeling of the creep behavior of polymers, there are few reviews on the modeling of creep in polymeric composites.

Among the earlier investigations are the articles by Haplin [19], Schapery [45], and Dillard [7]. Most of this literature utilizes linear viscoelastic models to describe the creep behavior of the composite.

Another approach to the modeling of creep derives form plasticity theory, which was applied initially to metal matrix composites [20,27]. For uniaxially reinforced polymeric composites, Sun [49,50] developed a simplified single parameter plasticity model for creep in unidirectionally rein-forced composites. Another unidirectional model was proposed by Robertson [41]. These models were subsequently extended to laminated plates and thermal effects by Gates [13,14]. Rate dependence was included by means of viscoplastic model by Gates and Sun [15,16].

Viscoplastic constitutive equations presented earlier by the authors [1,3], were written explicitly and they involve many parameters, which significantly influence the behavior of the constitutive equations. Appropriate parameters must be determined accordingly, such that the accurate behaviors of the material can be expressed.

Some of the problems involved with the explicit constitutive models, from our earlier investigation [3], include:

(1) The models are simply based on the phenomenological investigation of material properties while a real behavior of material is very complex. Therefore, inevitably the model contains errors.
(2) All the models are limited in their mathematical capabilities when they are tackled as parameter identification problem, since they are written explicitly.
(3) Compared to metals, composite materials encounter extra parameters like fiber orientation, volume fraction, fiber-matrix interface... etc. These parameters will raise the degree of complexity of the earlier constitutive models.

Bearing in mind the shortcoming of the phenomenological model, an alternative modeling technique is to use a computation and knowledge representation paradigm i.e.; *neural networks.* Recently, computa-

tional mechanics, which is quantitatively reliable by itself, has widened its feasibility to practical engineering problems by merging the *artificial neural network's* (*ANN*) flexibility.

Neural networks architecture is a promising implicit modeling scheme aiming to replace the traditional explicit constitutive equations used to describe the material behavior. The rest of this investigation is dedicated to the design of implicit creep model by means of ANN to predict the creep behavior of the PMCs under different thermomechanical environments.

At this point we emphasize that the approach we followed, does not produce an "explicit formula" supplying the creep properties for each viscoplastic—temperature—stress–strain law, but to the construction of an appropriate neural network, which, at a given time period "produces" a set of data corresponding to different viscoplastic states. This neural network "learns", and if applied to another set of experimental data, may fulfill its task more accurately in shorter time. The present method may replace to some extent the experiments after a period of "learning". Moreover, the method developed here may replace the classical explicit constitutive models for viscoplastic behavior, since it takes better into account experimental data and automatically improves itself through learning.

The proposed neural-creep model strongly contradicts the reference to neural network as a "black box model", as referred to in the majority of the neural networks-based materials models. The concept of black box arises in several investigations that use a "crude" training algorithm such as steepest descent where the learning rate is determined by a random walk approach. The current study eliminates this randomness by implementing recent but well-established numerical optimization techniques such as *the conjugate gradient* (*CG*) *and the truncated Newton* (*TN*) methods.

While these methods have been employed through several applications in numerical optimization (system theory, control systems, economy, meteorology, and many more) surprisingly, not many applications in material science have adopted these powerful techniques. One explanation of the lack of interest is that these techniques CG and TN, are basically unconstrained optimization techniques that are usually used to find an optimal solution for large-scale multidimensional problems, while, most of the neural networks-based materials models fall within the small-scale one-dimensional spectrum [11,18,26,28,55,56].

Therefore, for the first time in the neural networks-based material constitutive models, a powerful unconstrained multidimensional optimization algorithm i.e., TN method was introduced as training algorithm for neural networks. This novel application of the TN was not cited before in the literature to the best of our knowledge.

2. Neural networks

Neural networks consist of layered-processing elements (PEs) and weighted connections. Each layer in a neural network consists of a collection of PEs; neurons, each PE in a neural network collects the values from all of its input connections, performs a predefined mathematical operation, and produces a single output value. The main advantage of neural networks is the fact they are able to use some 'a priori' unknown information hidden in data (but they are not able to extract it). The process of *capturing* the unknown information is called *learning* or *training* of neural networks. A training cycle consists of the following steps: An input vector is presented at the inputs together with a set of desired responses, one for each node, at the output layer. A forward pass is carried out and the errors between the desired and actual response for each node in the output layer, are found. These are then used to determine weight changes in the net according to the prevailing learning rule. The best-known learning algorithm is the *backpropagation algorithm* [38,53,54].

A popular measure of the error E for a single pattern, is the sum of the square differences

$$E = \frac{1}{2}\sum_i (t_i - y_i)^2, \tag{1}$$

where t_i is the desired or target response on the ith unit and y_i is that actually produced on the same unit.

3. Neurocomputing and optimization

It is not our goal to discuss the formulation of different optimization techniques to be used in the training of neural networks. In the following, we just introduce briefly the outlines of three optimization techniques that will be used for building the neural networks-based viscoplastic model. Detailed discussion about these methods can be found in [9,17,21,37,40,42,48].

3.1. The steepest descent with momentum algorithm

The standard backpropagation implements the *steepest descent method* (also called the *gradient descent* method). At each step of the steepest descent method the weights are adjusted in the direction in which the error function decrease most rapidly. This direction is determined by the gradient of the error surface at the current point in the weight space.

Having the gradient of the error (also referred to as *error sensitivity*), then the weight adjustments for the connections are updated in a negative direction to the gradient with a certain rate, as given by

$$w_{ij}^{\text{new}} = w_{ij}^{\text{old}} - \alpha \frac{\partial E}{w_{ij}^{\text{old}}}, \tag{2}$$

where α is a positive-valued constants that regulate the amount of adjustments made with each gradient move, they are called *learning rates*. The learning rates determine what amount of the calculated error sensitivity to weight change will be used for the weight correction. The *momentum* was introduced to prompt learning while minimizing unstable behavior. Here, the error function is modified so that a portion of the previous weight is fed through to the current weight. Hence, the weight is updated according to

$$w_{ij}^{\text{new}} = w_{ij}^{\text{old}} - \alpha \frac{\partial E}{\partial w_{ij}^{\text{old}}} + \lambda \Delta w_{ij}^{\text{old}}, \tag{3}$$

where λ, is the momentum factor ($\lambda \in \langle 0, 1 \rangle$).

3.2. The conjugate gradient algorithm

Since learning in realistic neural networks application often involves adjustment of several hundred weights, only optimization methods applicable to large-scale problems are relevant as alternative learning algorithms. The general opinion in the numerical analysis community is that *CG methods* are well suited to handle large-scale problems in an effective way [8,9]. Several CG algorithms have recently been introduced as learning tools in neural networks. Some earlier applications of CG as training algorithm were carried out by Battiti [4] and Johansson [24]. The CG algorithm combines the advantages of simplicity of the steepest descent method and better convergence without the evaluation, inversion, and storage of the Hessian matrix $\mathbf{H}$. Among unconstrained optimization methods, it is widely acknowledged that the CG method is perhaps the easiest method applicable to large-scale problems [8]. The algorithm provides a tool to update the learning rate to minimize the mean square error; hence, the learning rate is updated at each epoch. The Polak-Ribiere version of the CG algorithm will be used in the current investigation. This algorithm is explained meticulously in [37].

3.3. The truncated Newton algorithm

In this method, a search direction is computed by finding an approximate solution to the Newton equations,

$$\mathbf{H}_k\mathbf{p}_k = -\mathbf{g}_k, \tag{4}$$

where $\mathbf{p}_k$ is a descent direction and $\mathbf{H}_k$ is the Hessian of the cost function $E(\mathbf{w}_k)$.

The use of an approximate search direction is justified by the fact that an exact solution of the Newton equation at a point far from the minimum is computationally wasteful in the framework of a basic descent method. Thus, for each outer iteration (Eq. (4)), there is an inner iteration loop applying the CG method that computes this approximate direction, $\mathbf{p}_k$, and attempts to satisfy a termination test of the form:

$$\|\mathbf{r}_k\| \leqslant \eta_k\|\mathbf{g}_k\|, \tag{5}$$

where residual

$$\mathbf{r}_k = \mathbf{H}_k\mathbf{p}_k + \mathbf{g}_k \tag{6}$$

and the sequence $\{\eta_k\}$ satisfies $0 < \eta_k < 1$ for all k.

The CG inner algorithm is pre-conditioned by a scaled two-step limited memory BFGS method in Nash TN method with Powell's restarting strategy used to reset the pre-conditioner periodically. A detailed description of the pre-conditioner may be found in Nash's work [30].

In TN method the Hessian vector product $\mathbf{H}_k\mathbf{v}$ for a given $\mathbf{v}$ required by the inner CG algorithm is obtained by a finite difference approximation,

$$\mathbf{H}_k\mathbf{v} \approx [\mathbf{g}(\mathbf{w}_k + h\mathbf{v}) - \mathbf{g}(\mathbf{w}_k)]/h. \tag{7}$$

A major issue is how to adequately choose h; in this work we use

$$h = \sqrt{\varepsilon(1 + \|\mathbf{w}\|)}, \tag{8}$$

where ε is the machine precision. The inner algorithm is terminated using a quadratic truncation test, which monitors a sufficient decrease of the quadratic model

$$\begin{aligned} &\mathbf{q}_k = \mathbf{p}_k^{\mathrm{T}}\mathbf{H}_k\mathbf{q}_k/2 + \mathbf{p}_k^{\mathrm{T}}\mathbf{q}_k, \\ &(1 - \mathbf{q}_k^{i-1})/\mathbf{q}_k^{i-1} \leqslant c_q/i, \end{aligned} \tag{9}$$

where i is the counter for the inner iteration and c_q is a tolerance, satisfying $0 < c_q < 1.0$.

TN methods can be extended to more general problems, which are not convex in much the same way as Newton's method [35,36,51,52,56].

If

$$c_q \leqslant \|\nabla f(\mathbf{w}_k)\| \tag{10}$$

then the TN method will converge quadratically, see [6,33,(34, p. 393)]. This fact explains the faster rate of convergence and the better quality of results obtained with the TN method.

Similar conclusions and more detailed discussion of the TN method can be found at [25,29–32,47].

4. Phenomenological (explicit) viscoplastic creep model

4.1. Analytical model description

For the purpose of comparison, the outline of an explicit viscoplastic model is presented here. A detailed investigation about this model can be found in earlier investigation by Gates and Sun [15,16] and recently in the work of Al-Haik et al. [1–3,12]. The model describes the rate-dependant stress/strain behavior in graphite reinforced polymeric composite.

The model treats the composite as a homogeneous orthotropic material, and it was constructed to approximately predict the observable phenomena (creep, relaxation, and quasi-static tensile) over a range of pertinent operating conditions (temperature, load level, and loading case, i.e., tension/compression). The viscoplastic creep strain is divided into two components according to

$$\dot{\varepsilon}^{\mathrm{vp}} = \dot{\varepsilon}^{(\mathrm{i})\mathrm{vp}} + \dot{\varepsilon}^{(\mathrm{ii})\mathrm{vp}}, \tag{11}$$

the first component of the viscoplastic term is given as

$$\dot{\varepsilon}^{(\mathrm{i})\mathrm{vp}} = \begin{Bmatrix} An(\sigma)^{n-1}\dot{\sigma} & \text{for } \dot{\sigma} > 0 \\ 0 & \text{for } \dot{\sigma} \leqslant 0 \end{Bmatrix}, \tag{12}$$

where A and n are materials parameters. The second part of the viscoplastic term is described as

$$\dot{\varepsilon}^{(\mathrm{ii})\mathrm{vp}} = \left[\frac{\langle H \rangle}{K}\right]^{1/m}, \tag{13}$$

where H is the overstress, $\langle\ \rangle$ are McCauley brackets, and K and m are material constants found from experimental data.

The overstress, $H = \sigma - \sigma^*$ is considered as a scalar quantity that relates the quasi-static stress σ^* to the dynamic or instantaneous stress σ at the same strain level.

Therefore,

$$\dot{\varepsilon}^{(\mathrm{ii})\mathrm{vp}} = \begin{Bmatrix} \left[\frac{(\sigma-\sigma^*)}{K}\right]^{1/m} & \text{if } \sigma > \sigma^* \\ 0 & \text{if } \sigma \leqslant \sigma^* \end{Bmatrix}. \tag{14}$$

The quasi-static stress, σ^* is found by solving the quasi-static stress–strain relationship,

$$\varepsilon = \frac{\sigma^*}{E} + A(\sigma^*)^n \tag{15}$$

while the dynamic stress is the stress resulting from the time-dependent material behavior.

The material parameters K, m, A, and n are temperature dependent and are found from experimental data (tensile and load relaxation tests at different temperatures). All these parameters were obtained from load relaxation and tensile data at in previous published investigations [1–3,12] and they are presented in Table 1.

During creep, the stress is constant and the stress rate is zero; therefore the total strain rate may be written in the form

$$\dot{\varepsilon}^{\mathrm{vp}} = \dot{\varepsilon}^{(\mathrm{i})\mathrm{vp}} = \left[\frac{(\sigma - \sigma^*)}{K}\right]^{1/m}. \tag{16}$$

This relationship is a first-order nonlinear differential equation and is coupled to a nonlinear expression of quasi-static stress through Eq. (15). Combined methods of numerical analysis for solving nonlinear equations (Newton method) and differential equations (Runge–Kutta) were implemented to solve this

Table 1
Polymeric composite properties and parameters for viscoplastic model

T (°C)	m	K (MPa)	n	A (MPa)
25	0.73	4.10×10^6	3.619	0.0088
35	0.69	1.04×10^5	1.166	0.0749
45	0.70	9.41×10^3	0.630	0.0749
50	0.62	1.16×10^3	1.372	0.0563
55	0.67	2.10×10^3	2.705	0.0247
60	0.69	1.57×10^5	1.518	0.0616
65	0.70	5.80×10^3	1.266	0.0637
75	0.66	1.39×10^6	1.793	0.0975

differential equation. The simulation of the viscoplasitc model was carried out for two representative temperatures: 35 and 65 °C, under different stress levels. The results will be compared to those obtained via the new proposed TN neural network model.

5. Neurocomputational creep model

Loading a specimen using a Universal Testing Machine to a specific load usually performs the creep test and the load is then maintained at a constant value and the strain is recorded. During creep experiment the load P remains constant

$$\dot{P} = 0 \tag{17}$$

since the total strain can be decomposed into elastic and inelastic components;

$$\varepsilon = \varepsilon_{\mathrm{e}} + \varepsilon_{\mathrm{i}} \tag{18}$$

the strain rate $\dot{\varepsilon}$ is found by differentiating Eq. (18) and noting that ε_{e} is a constant

$$\dot{\varepsilon} = \dot{\varepsilon}_{\mathrm{i}}. \tag{19}$$

Fig. 1 depicts the behavior of the creep strain at different stress levels, where σ_0, σ_1 and σ_2 are normalized stress levels at which the creep testes were performed. The normalization of the stress level was done with respect to the strength at the working temperature T^0. Hence, the input to the neural network was an array of 3×1 cells where the elements of each cell represents the [temperature; stress; time] respectively. The targets were chosen to be the corresponding values of the creep strain for each input cell.

The data sets were produced from a combination of eight temperatures (25, 35, 45, 50, 55, 60, 65, and 75 °C that were selected according to the measured glass transition temperature of the polymer ($T_{\mathrm{g}} = 86$ °C), six normalized stress levels (30%, 40%, 50%, 60%, 70%, and 80%), and 36 time steps with 100 s increment, i.e. $100, 200, \ldots, 3600$ s. The total number of data points produced was calculated as: $8(T^0) \times 6(\sigma) \times 36(t) = 1728[T^0, \sigma, t]$ input-target cells.

The neural networks computations do not favor the ''raw'' data to be used for training; hence, the input data $[T^0, \sigma, t]$ and the targets $[\varepsilon]$ were normalized to values ranging between -1 and 1 using the following formula

$$x_n = 2\frac{x - x_{\min}}{x_{\max} - x_{\min}} - 1, \tag{20}$$

where x_n is the normalized value of the vector $x = [T^0, \sigma, t, \varepsilon]$, $x_{\min}$ and $x_{\max}$ are the minimum and maximum values in the database for the vector x. After scaling the 1728 cells of inputs-targets, the scaled results will be

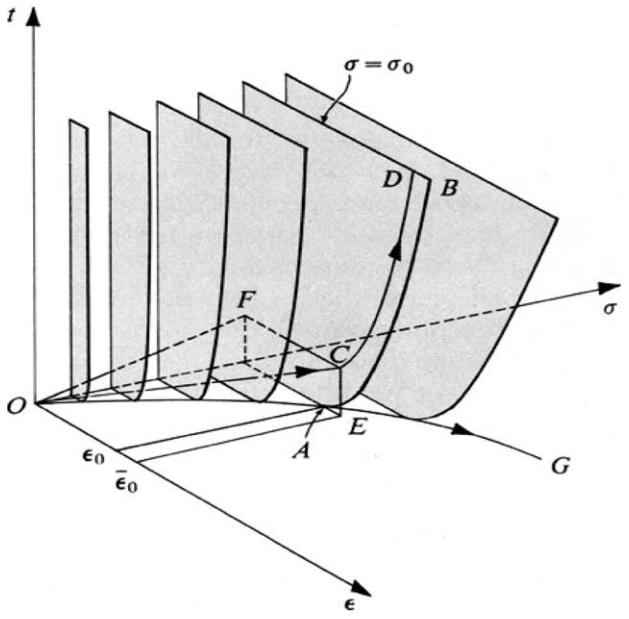

Fig. 1. Stress–strain–time space with schematic creep curves.

split into three subsets; one set will be used for training, another set for validation, and the last set for testing the network performance. The training and validation sets consist of 1200 (roughly $2=3$ of the entire data sets) pairs of scaled data that covers the temperature ranges of 25, 45, 55, 60, and 75 °C: These 1200 data points will be split into 800 pairs for training and 400 pairs for validation. The remaining 528 cells left that cover the creep tests at temperatures: 35, 50, and 65 °C will be kept aside to test the neural network creep model after the network has been trained and validated. The scaled data sets should be randomized so that the training process of the network does not consist of a table look up problem, and to eliminate any bias that might exist in the training data set.

Based on the Universal Approximation Theorem, Hornik [23] proposed that: ''A two hidden layer network is capable of approximating any useful function''. Also, Hornik stated that the mapping power of feed forward neural network (FF NN), is not inherent in the choice of a specific activation function, rather ''it is the multilayer feed forward structure that leads to the general function approximation capability''.

Following the universal approximation theorem and the reasoning provided by Hornik [22], the current investigation adopts a two-hidden layer neural network structure. The number of neurons at each hidden layer was obtained through training the network using a standard backpropagation algorithm with two design parameters: learning rate and momentum. The performance of several structures of neural networks is shown in Fig. 2. From this simulation it is obvious that, the [6-20-1] structure (six neurons at the first hidden layer, twenty neurons at the second hidden layer and a single neuron at the output layer) achieved an optimal number of neurons. This optimal structure produced MSE = 0.12105, which still higher than the pre-assigned error goal i.e. $\mathrm{MSE}_{\mathrm{goal}} = 1 \times 10^{-5}$. The results of this crude network were used to monitor the performance index; i.e. the mean square error. The structure of this optimal-size network is shown in Fig. 3.

Both hidden layers have tansigmoidal activation functions φ_1, and φ_2 respectively, while the activation function of the output layer is a linear function φ_3. Thus, the objective function built through the proposed neural networks structure can be written in a compact notation as:

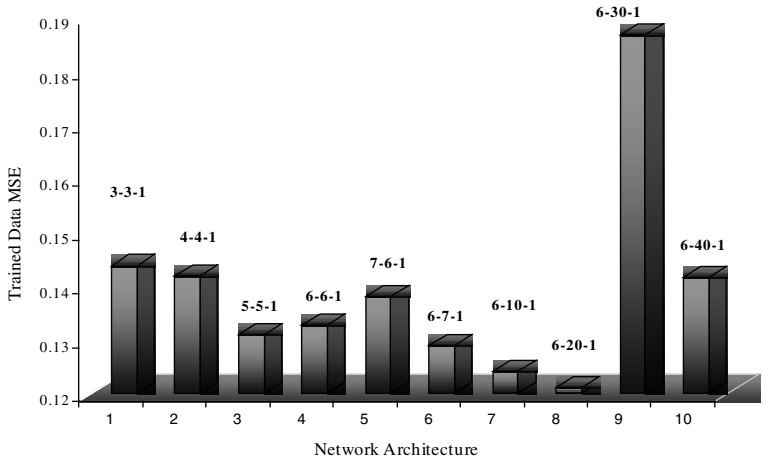

Fig. 2. Structure analysis for the two-hidden layers neural network based on the mean squared error for the 798 pairs of training data.

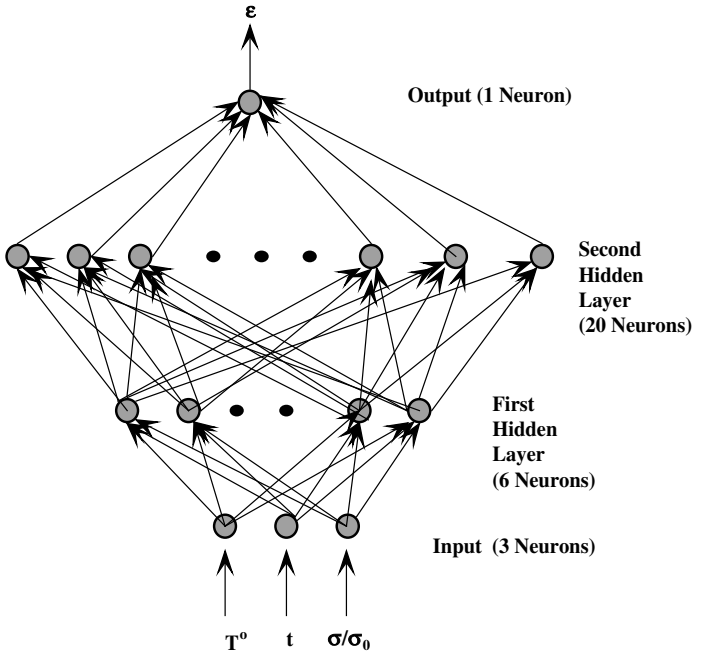

Fig. 3. The structure of the proposed neural model to predict the creep behavior of PMC.

$$f_k(x,w) = \sum_{j=1}^{6} w_{jk}\varphi_3\left(\varphi_2\left(\sum_{m=1}^{20} w_{lm}\varphi_1(\mathbf{X})\right)\right),$$
$$\text{where } \mathbf{X} = \sum_{i=1}^{6}(w_{pi}x_i),\ \ p = 1,\ldots,3,\ \ m = 1,\ldots,6,\ \ k = 1 \tag{21}$$

and the corresponding objective function to be minimized is then given as the mean square error between the approximation function f and the actual target t for each training pair $k = 1; 2, \ldots, 798$

$$E(\mathbf{w}) = \frac{1}{2}\sum_{k=1}^{798}(t_k - f_k)^{\mathrm{T}}(t_k - f_k). \tag{22}$$

This function should be minimized with respect to the weight values w_{jk}, w_{lm}, and w_{pi} that can be stacked into one vector $\mathbf{w}$ consisting of: 3×6 values, at first hidden layer, 6×20 values at second hidden layer, and finally 20×1 values at the output layer, hence $\mathbf{w}$ is a vector of 158 variables. Training the neural network consists of finding the optimal values of w that minimize the error function $E(\mathbf{w})$ using the optimization techniques described earlier.

6. Results and discussion

Using the steepest descent algorithm with momentum, the network performance can be improved by finding optimal values for learning rate (α) and the momentum coefficient (λ). First the number of iterations (epochs) was extended to 1000 and the learning rates were set to different values (0.05, 0.1, 0.3, 0.5, and 0.8), similarly the momentum values varied from 0.05 to 0.8. The corresponding MSE surface can be visualized as shown in Fig. 4. The visualized MSE surface is highly nonsmooth. However, the values of the learning rate of $\alpha = 0{:}10$ and momentum $\lambda = 0{:}3$ generated the least MSE $= 0{:}119678$, that is much higher than the pre-assigned MSE goal of 1×10^{-5}.

Bearing in mind the problem associated with choosing an optimal combination of the training parameters, the implementation of the standard backpropagation algorithm to minimize the MSE is not the perfect learning algorithm for the implicit creep model. Validating a model in ANN requires using some data points which were not utilized in the training phase, while these data points still fall in the range of the training set. The validation phase makes use of the 400 data points that were left out of the 1200 data points initially generated. These validation data have to be preprocessed in the same scaling and randomizing fashion of the training data.

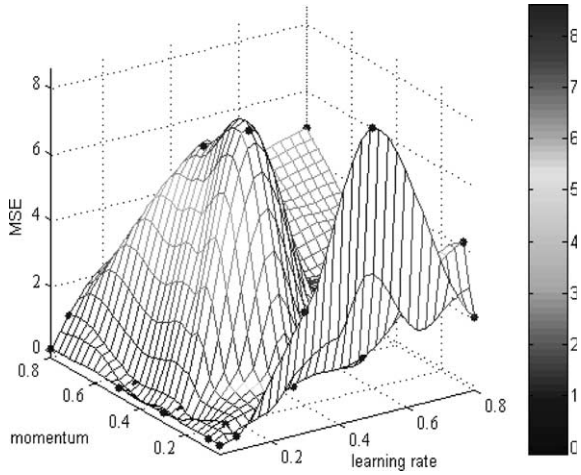

Fig. 4. The behavior of the MSE error as calculated using different values of learning rate and momentum for the steepest descent method.

For testing the network we introduce a completely new data set that does not belong to the training and validation data. For example, one can test the network at $T = 35$ °C with stress levels 30%, 50%, and 80% respectively, over one hour simulation time span. The performances for training, validation, and test sets are simulated as shown in Fig. 5. The network was trained for 1000 epoch to check if the performance (MSE) for either validating or testing sets might diverge, which did not occur as shown in Fig. 5.

The results of MSE appear reasonable in terms of generalizing the ANN for new test sets, but in order to confirm these results we need to compare the actual values for creep strain with those produced by the ANN. After scaling back the ANN results for creep strain, they were plotted together with the experimental values as shown in Fig. 6.

Fig. 6 shows clearly that the standard backpropagation algorithm failed to capture the creep behavior at 35 °C. Similar unsuccessful results were reported for the ANN creep model at $T = 50$ °C and $T = 65$ °C, see [2].

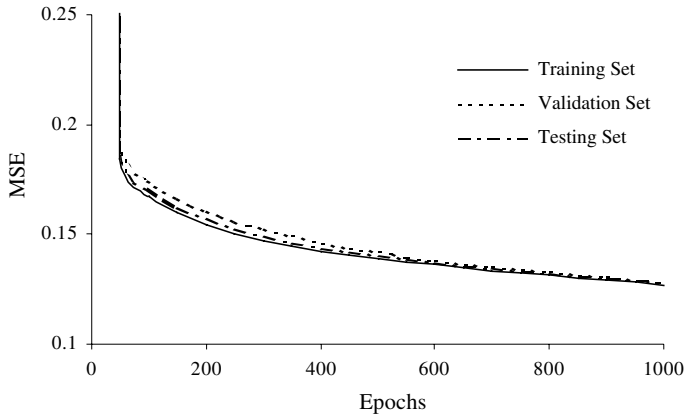

Fig. 5. MSE error for training, validation, and testing sets, for the [6-20-1] ANN based on the steepest descent with momentum backpropagation training algorithm.

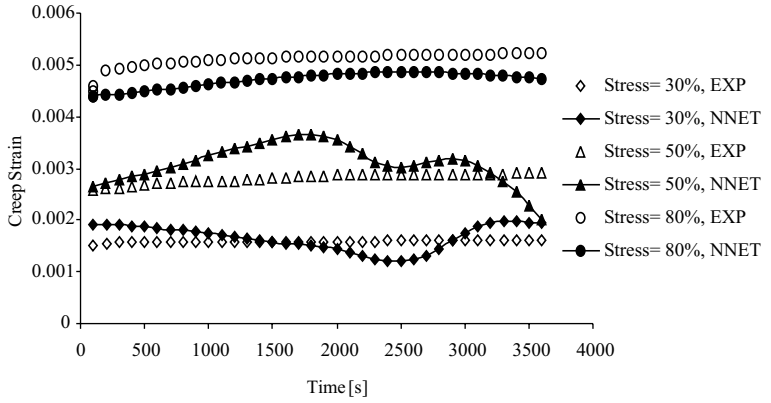

Fig. 6. Testing the [6-20-1] ANN based on the steepest descent with momentum backpropagation, for $T = 35$ °C, at 30%, 50%, and 80% stress levels respectively. The experimental creep strain is plotted together with the simulated results of the ANN.

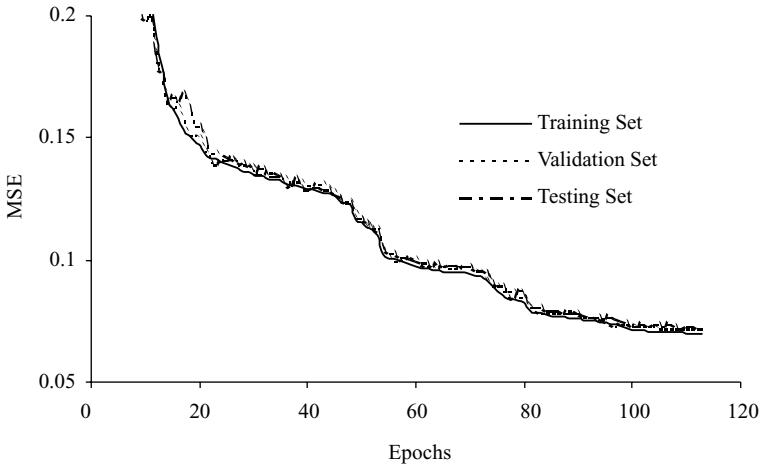

Fig. 7. MSE error for training, validation, and testing sets for the [6-20-1] ANN with backpropagation training algorithm that utilizes the CG (Polak-Ribiere) algorithm.

By applying it to the implicit creep model, the CG algorithm will be used to determine the weights updates. The [6-20-1] structure with tansigmoidal activation functions in a batch mode will be used; the gradient of the error function is computed after the entire training has been presented to the entire network.

Similarly, the training with the CG algorithm, validation, and testing sets as in the standard backpropagation were used. Compared to the steepest descent backpropagation, the CG minimization algorithm produced a smaller MSE for all the three phases of training, validation, and testing. Fig. 7 shows that, the resulting MSE error for the CG training algorithm is 50% less than that for the standard backpropagation. Another, important conclusion that can be drawn from Fig. 7 is that, the CG with line search backpropagation is an order of magnitude faster than the steepest descent backpropagation. The CG with line search required exactly 113 epochs for the MSE (for the training set) to drop to a value of 0.06997, compared to 1000 epochs required to reach a value of 0.12707 MSE for the standard backpropagation method.

We tested the backpropagation with CG training algorithm for the implicit creep model at $T = 35$ °C for levels of stress of: 30%, 50%, and 80% respectively. The CG-based simulation produced far more acceptable results than those obtained via standard method, as shown in Fig. 8. While the simulations for the cases of 30% and 50% stress levels look satisfactory, the ANN model results have a considerable error for the case of 80% stress level.

We conclude that the standard CG algorithm was by far more superior to the standard backpropagation in both reducing the mean squared error in less number of epochs, and in better generalization for the network for the test data set. However, these attractive results were achieved at the expense of additional computational effort, namely the line search technique in order to achieve the optimal learning rate that will be modified at each successive step to reach the goal of minimizing the mean square error function.

While also converging linearly, the CG method has better convergence rate in particular when the condition of $\mathbf{H}$ is large (see [34, pp. 344, 388]).

Implementing the same architecture of the implicit creep model network; [6-20-1], and using now the TN minimization as the training algorithm, the error function as a stopping criteria, the algorithm converged to a satisfactory error of 0.01323 after 160 epochs as shown in Fig. 9.

The results of the simulation of the TN minimization algorithm were very close to the actual experimental results as shown in Fig. 10.

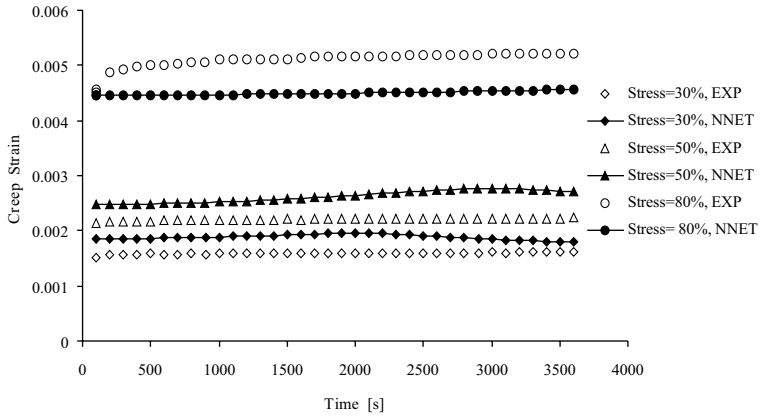

Fig. 8. Testing the [6-20-1] ANN, with nonlinear CG (Polak-Ribere) backpropagation, for $T = 35$ °C, at 30%, 50%, and 80% stress levels respectively. The experimental creep strain is plotted together with the simulated results of the ANN.

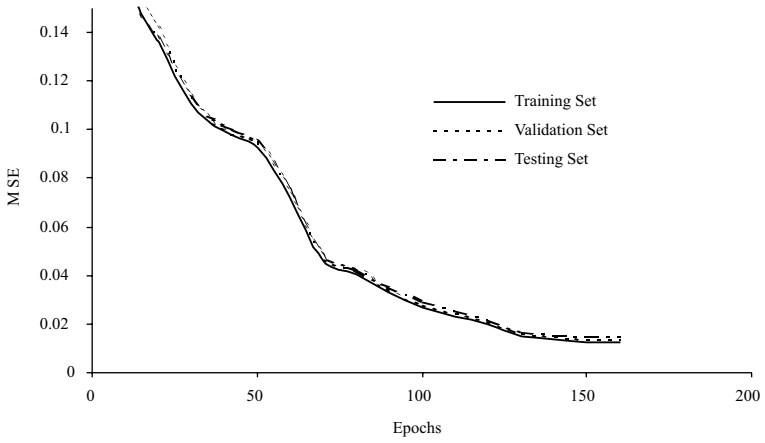

Fig. 9. MSE error for training, validation, and testing sets, for [6-20-1] ANN with backpropagation training algorithm that utilizes the TN method.

The performance of the TN algorithm was compared to that of both the steepest descent and the nonlinear CG as shown in Figs. 11 and 12. The TN algorithm attained the lowest MSE requiring only a relatively moderate number of epochs. This is to be expected in view of the quadratic convergence rate of the TN.

Considering the mean square error function described in Eq. (22), we compared the performance of the steepest descent, TN, and the nonlinear CG minimization method based on Polak-Ribiere formula (CG-PR), all methods being allowed up to 350 iterations (the outer loop for the case of TN). The results show that TN algorithm converges by one order of magnitude faster than both steepest descent and CG-PR, and it attained a lower gradient norm as shown in Fig. 11. Also, the TN algorithm significantly outperforms both the steepest descent and the CG-PR methods on two counts; the final accuracy achieved and the total number of function and gradient evaluations required to achieve a prescribed accuracy. The comparison is shown in Figs. 11 and 12. The performance of the three algorithms carried out in the current investigation is summarized in Table 2.

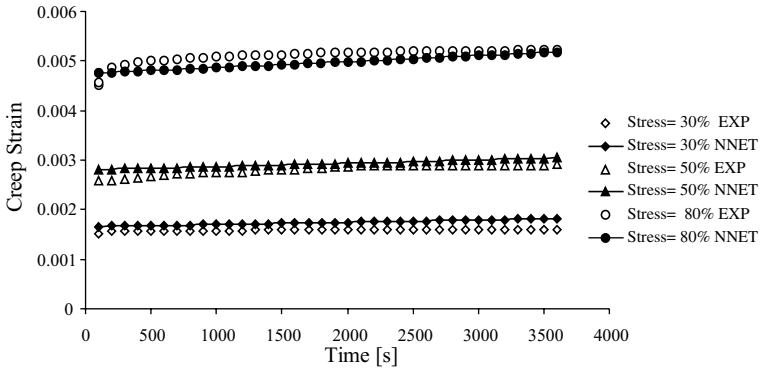

Fig. 10. Testing the [6-20-1] ANN, with TN-based backpropagation, for $T = 35$ °C, vat 30%, 50%, and 80% stress levels. The experimental creep strain is plotted together with the simulated results of the ANN.

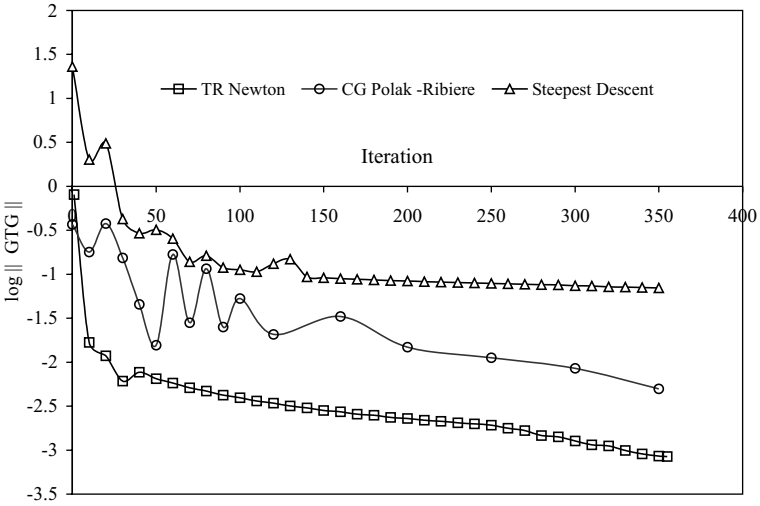

Fig. 11. Comparison of the gradient norm for the error function constructed by neural networks. Gradient was generated by steepest descent, TN and Polak-Rebiere CG methods respectively.

The numerical results reported here highlight the fact that the TN algorithm using, in its inner-loop iteration, a CG-based pre-conditioner significantly outperforms the standard nonlinear CG in solving large-scale unconstrained minimization problems associated with neural networks models.

The neural networks-based creep model is more cost efficient compared to the explicit viscoplastic model [3]; only one type of data is required, i.e., creep data at different thermomechanical histories, while the explicit viscoplastic model required both tensile tests data along with load relaxation data, and of course creep data is still required to verify the performance of the model. While the duration of tensile and load relaxation tests are short compared to the creep tests, the outcome of using tensile load relaxation-based viscoplastic model produced considerable errors when predicting the creep strain [2,3]. On the other hand, the results of the neural model, justifies the cost of the longer duration creep tests.

The experimental creep behavior of the composite together with viscoplastic model simulation and the neural network simulation are presented in Figs. 13 and 14. Unlike the explicit viscoplastic model, neural

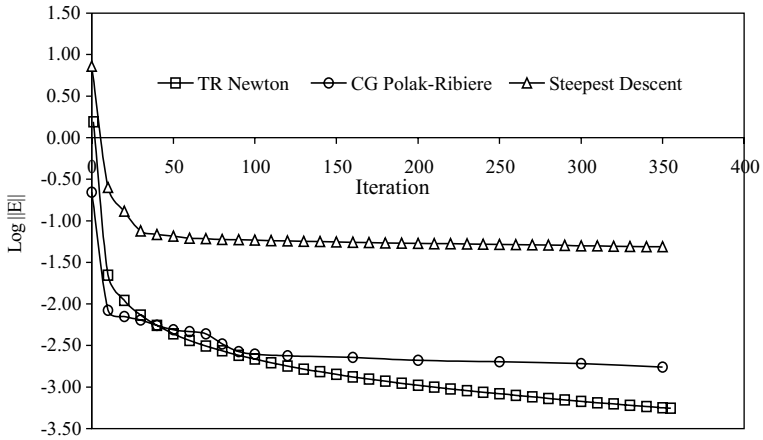

Fig. 12. The decrease of the neural networks mean squared error function by. Function was evaluated by steepest descent, TN method and Polak-Rebiere CG methods respectively.

Table 2
Performance of three different algorithms for the implicit creep model using the [6-20-1] topology with tansigmoidal activation functions in the hidden layers

Training algorithm	MSE	Epochs
Steepest descent	0.11967	1000
CG (Polak-Ribiere)	0.06997	113
TN with CG-pre-conditioner	0.01323	160

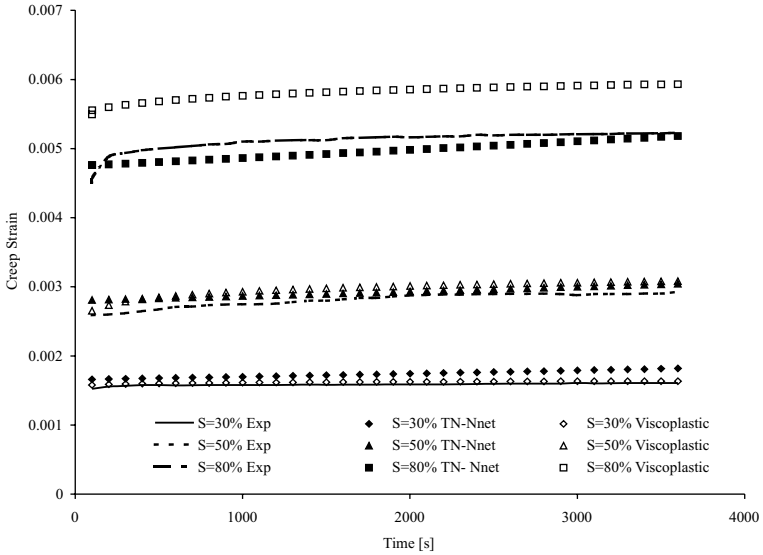

Fig. 13. Validation of explicit viscoplastic and neural networks model (TN) for the creep evaluation at $T = 35$ °C and stress levels 30%, 50%, and 80% of the composite strength at this temperature.

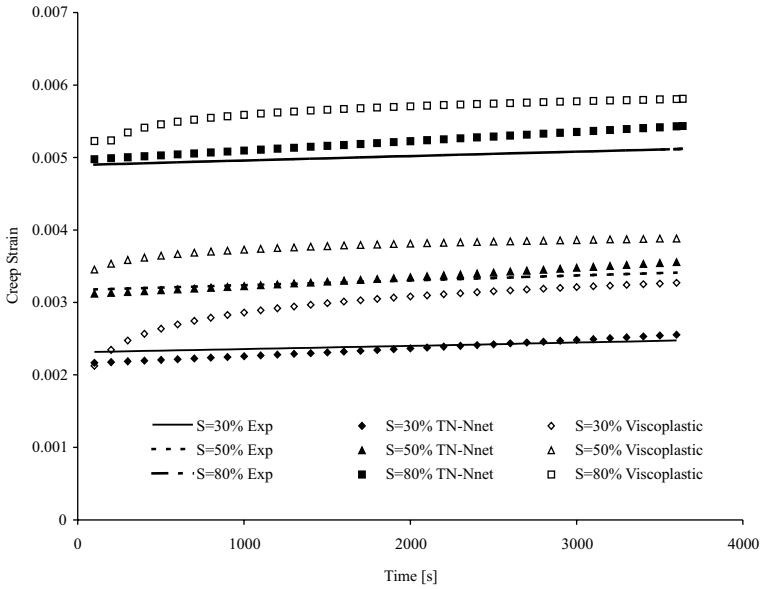

Fig. 14. Validation of explicit viscoplastic and neural networks model (TN) for the creep evaluation at $T = 65$ °C and stress levels 30%, 50%, and 80% of the composite strength at this temperature.

network model predicted more accurate results at different stress–temperature conditions especially under conditions of relatively high temperature. Simulating the creep behavior of the composite at temperatures (65 °C) closer to the glass transition temperature of the composite (86 °C), revealed the discrepancy of the explicit viscoplastic model to capture the actual behavior of the composite in the vicinity of the glass transition region.

Moreover, in building the neural network creep model, only one type of data is required, that is creep data at different thermomechanical histories, while empirical viscoplastic model requires both tensile tests data together with load relaxation data, and of course creep data still required to verify the performance of the model.

7. Concluding remarks

In this study, the viscoplastic behavior of a carbon-fiber/thixotropic-epoxy matrix composite was investigated. The creep behavior of the composite was captured by a neural network formulation. The neural network model was built directly from the experimental results obtained via creep tests performed at various stress–temperature conditions. The optimal structure of the neural network was achieved through the universal approximation theory and the dimensionality of the creep problem; (stress, temperature, and time). The neural network model was trained to predict the creep strain based on the stress–temperature–time values. The performance of the neural model is represented by the mean squared error between the neural network prediction and the experimental creep strain results. To minimize this error, several optimization techniques were examined. The minimization of the error when carried out by the TN method outperforms both the steepest descent and CG methods in terms of convergence rate and accuracy.

Using neural network with TN training algorithm, the prediction of the creep strain was very satisfactory compared to the experimental results.

TN method is proven efficient for solving a large-scale optimization problem. It achieved a desired quadratic convergence rate while limiting the waste of oversolving the model at points far from the solution, where the steepest descent method usually fails. The TN method was easily customized to fit the neurocomputational creep model, eliminating the randomness encountered for choosing an optimal learning rate and momentum for the steepest descent method. Simulating the behavior of the composite at temperatures closer to the glass transition temperature of the composite revealed the discrepancy of the explicit viscoplastic model to capture the actual creep behavior of the composite. The TN-neural network model has succeeded in effectively capturing the creep evolution of the composite at any state belonging to the 3-dimensional space of stress–temperature–time.

The work presented here establishes a foundation of a new application of neural networks combined with advanced optimization algorithms to materials science, from which a great deal of additional development and improvements can be pursued.

Acknowledgements

M. Al-Haik and H. Garmestani are grateful to the Structural Research Center, Florida Department of Transportation, for providing the materials used in this research.

I.M. Navon would like to acknowledge support from the NSF grant no. ATM-0201808 managed by Dr. Linda Peng whom he would like to thank for her support. He also acknowledges support from the School of Computational Science and Information Technology.

References

[1] M. Al-Haik, H. Garmestani, Durability study of a polymeric composite material for structural applications, J. Polym. Comp. 22 (6) (2001) 779–792.
[2] M. Al-Haik, Durability of a polymer matrix composite, a neural networks Approach, Ph.D. Thesis, Department of Mechanical Engineering, Florida Sate University, 2002.
[3] M. Al-Haik, M.R. Vaghar, H. Garmestani, M. Shahawy, Viscoplastic analysis of structural polymer composites using stress relaxation and creep data, Compos. Part B—Engrg. 32 (2001) 165–170.
[4] R. Battiti, Accelerated backpropagation learning: Two optimization methods, Complex Syst. 3 (1989) 331–342.
[5] J. Corum, Durability of Polymer Matrix Composites for Automotive Structural Applications: A State of the Art Review, Technical Report ORNL-6869, Oak Ridge National Laboratory, 1995.
[6] P.S. Dembo, T. Steihaug, Truncated Newton algorithms for large-scale unconstrained optimization, Math. Program. 26 (1983) 190–212.
[7] D.A. Dillard, Viscoelastic Behavior of Laminated Composite Materials, Elsevier Applied Science Publishers Ltd., England, 1991.
[8] R. Fletcher, Practical methods of optimization, second ed., John Wiley and Sons, 1987.
[9] R. Fletcher, C. Reeves, Function minimization by conjugate gradients, Computer J. 7 (1964) 149–154.
[10] L. Franke, H.J. Meyer, Predicting the tensile strength and creep-rupture behavior of pultruded glass reinforced polymer rods, J. Mater. Sci. 27 (1992) 4899–4908.
[11] T. Furukawa, G. Yagawa, Implicit constitutive modeling for viscoplasticity using neural networks, Int. J. Numer. Meth. Engrg. (1998) 195–219.
[12] H. Garmestani, M.S. Al-Haik, S. Downey, Durability analysis of an autoclave structural polymer composite using stress relaxation and creep data, in: Proc. CANCOM'2001, Quebec, Canada, 2001, pp. 136–141.
[13] T.S. Gates, Matrix dominated stress/strain behavior in polymeric composites: effects of hold time, nonlinearity and rate dependency, in: E.T. Camponeschi, Jr. (Ed.), Composite Materials: Testing and Design, vol. 11, ASTM STP 1206, American Society for Testing and Material, 1993, pp. 177–189.
[14] T.S. Gates, Effects of elevated temperature on the viscoplastic modeling of graphite/polymeric composites, in: C.E. Harris, T.S. Gates (Eds.), High Temperature and Environmental Effects on Polymeric Composites, ASTM STP 1174, American Society for Testing and Material, 1991, pp. 201–221.
[15] T.S. Gates, C.T. Sun, Elastic/viscoplastic constitutive model for fiber reinforced thermoplastic composites, AIAA 29 (3) (1991) 457–463.

[16] T.S. Gates, J.L. Chen, C.T. Sun, Micromechanical characterization of nonlinear behavior of advanced polymer matrix composites, in: C.R. Saff, R.B. Deo (Eds.), Composite Materials: Testing and Design, vol. 12, ASTM STP 1274, American Society for Testing and Materials, 1995, pp. 295–319.
[17] P.E. Gill, W. Murray, Newton-type methods for unconstrained and linearly constrained optimization, Math. Program. 28 (1974) 311–350.
[18] J. Ghaboussi, J.H. Garrett, X. Wu, Knowledge-based modeling of materials behavior with neural networks, J. Engrg. Mech—ASCE 117 (1) (1991) 132–153.
[19] J.C. Haplin, Introduction to Viscoelasticity, Technomic Publishing Co., 1968.
[20] E.W. Hart, H.D. Solomon, Load relaxation studies of polycrystalline high purity aluminum, Acta. Metall. Mater. 21 (1973) 295–307.
[21] S. Haykin, Neural Networks: A Comprehensive Foundation, second ed., Prentice Hall, New Jersey, 1999.
[22] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural Networks 4 (1991) 251–257.
[23] K. Hornik, M. Stinchocombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks 2 (1989) 359–366.
[24] E.M. Johansson, F.U. Dowla, D.M. Goodman, Backpropagation learning for multi-layer feedforward neural networks using the conjugate gradient method, Int. J. Neural Syst. 2 (1991) 291–302.
[25] F.X. Le Dimet, I.M. Navon, D.N. Daescu, Second order information in data assimilation, Mon. Weather Rev. 130 (2002) 629–648.
[26] C.J. Li, A. Ray, Neural network representation of fatigue damage dynamics, Smart Mater. Struct. 4 (1995) 126–133.
[27] M.C. Liu, E. Krempl, A uniaxial viscoplastic model based on total strain and overstress, J. Mech. Phys. Solids 27 (1979) 377–391.
[28] A. Mukherjee, S.N. Biswas, Artificial neural network in prediction of mechanical behavior of concrete at high temperature, Nucl. Engrg. Des. 178 (4) (1997) 1–11.
[29] S.G. Nash, Newton-type minimization via the Lanczos method, SIAM J. Numer. Anal. 21 (1984) 770–788.
[30] S.G. Nash, Preconditioning of truncated Newton methods, SIAM J. Sci. Stat. Comp. 6 (1985) 599–616.
[31] S.G. Nash, J. Nocedal, A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization, SIAM J. Optimiz. 1 (1991) 358–372.
[32] S.G. Nash, A. Sofer, Block truncated-Newton methods for parallel optimization, Math. Program. 45 (1989) 529–546.
[33] S.G. Nash, A. Sofer, Assessing a search direction within a truncated Newton method, Oper. Res. Lett. 9 (1990) 219–221.
[34] S.G. Nash, A. Sofer, Linear and Nonlinear Programming, McGraw-Hill, 1996.
[35] I.M. Navon, X. Zou, M. Berger, P.K. Phua, T. Schlick, F.X. Le Dimet, Numerical experience with limited memory quasi-Newton and truncated Newton methods, in: K. Phua (Ed.), Optimization Techniques and Applications, World Scientific Publishing Co., 1992, pp. 33–48.
[36] I.M. Navon, X. Zou, M. Berger, P.K. Phua, T. Schlick, F.X. Le Dimet, Testing for reliability and robustness of optimization codes for large scale optimization problems, in: K. Phua (Ed.), Optimization Techniques and Applications, World Scientific Publishing Co., 1992, pp. 445–480.
[37] J. Nocedal, S. Wright, Numerical Optimization, Springer-Verlag, 1999.
[38] B.D. Parker, Second order backpropagation, in: IEEE First International Conference on Neural Networks, vol. 2, 1987, pp. 593–600.
[39] D. Pertez, Y. Weitsman, Nonlinear viscoelastic characterization of FM-73 adhesive, J. Rheology 26 (3) (1966) 245–261.
[40] J.C. Principe, N.R. Euliano, W.C. Lefebvre, Neural and Adaptive Systems: Fundamentals through Simulations, John Wiley and Sons Inc., New York, 2000.
[41] D. Robertson, S. Mall, Micromechanical analysis for thermoviscoplastic behavior of unidirectional fibrous composites, Comp. Sci. Tech. 50 (1994) 483–496.
[42] R. Schalkoff, Artificial Neural Networks, McGraw-Hill Inc., New York, 1997.
[43] R.A. Schapery, A method of viscoelastic stress analysis using elastic solutions, J. Franklin Inst. 279 (74) (1965) 268–289.
[44] R.A. Schapery, An engineering theory of nonlinear viscoelasticity with applications, Int. J. Solids. Struct. 2 (1966) 407–425.
[45] R.A. Schapery, Stress Analysis of Viscoelastic Composite Materials, Technomic Publishing Co., 1968.
[46] E. Shin, R. Morgan, Performance evaluation of glass fiber-epoxy composites for durable goods applications, in: Proceedings of the 10th Annual ASM/ESD Advanced Composites Conference, 1994, pp. 545–554.
[47] T. Schlick, A. Fogelson, TNPACK: A truncated Newton minimization package for large-scale problems: I. algorithm and usage, ACM Trans. Math. Soft. 18 (1992) 46–70.
[48] A. Skinner, J. Broughton, Neural networks in computational materials science: Training algorithms, Model. Simul. Mater. Sci. Engrg. 3 (1995) 371–390.
[49] C.T. Sun, J.L. Chen, A micromechanical model for plastic behavior of fibrous composites, Comp. Sci. Tech. 40 (1991) 115–129.
[50] C.T. Sun, I. Chang, Modeling of elastic-plastic behavior of IDF and continuous fiber reinforced AS-4/PEEK composites, Comp. Sci. Tech. 43 (1992) 339–345.
[51] Z. Wang, I.M. Navon, X. Zou, F.X. Le Dimet, Truncated-Newton optimization algorithm in meteorology applications with analytic Hessian vector products, Comput. Optim. Appl. 4 (1995) 241–262.

[52] Z. Wang, K.K. Droegemeier, L. White, I.M. Navon, Application of a new adjoint Newton algorithm to the 3-d ARPS storm scale model using simulated data, Mon. Weather Rev. 125 (1997) 2460–2478.
[53] P. Werbos, Backpropagation past and future, in: IEEE International Conference of Neural Networks, vol. 1, 1988, pp. 343–353.
[54] X. Wu, Neural network based material modeling, Ph.D. Thesis, Department of Civil Engineering, University of Illinois at Urbana-Champaign, 1991.
[55] I.C. Yeh, Modeling of strength of high-performance concrete using artificial neural networks, Cement Concrete Res. 28 (12) (1999) 1797–1808.
[56] X. Zou, I.M. Navon, M. Berger, P.K. Phua, T. Schlick, F.X. Le Dimet, Numerical experience with limited memory quasi-Newton methods and truncated Newton methods, SIAM J. Optimiz. 3 (1993) 582–608.