

Incorporating Neural Network Material Models within Finite Element Analysis for Rheological Behavior Prediction

B. Scott Kessler, A. Sherif El-Gizawy, and Douglas E. Smith

University of Missouri-Columbia
E2411 Engineering Building East
Columbia, MO 65211

Phone: (573) 882-9569, Fax: (573) 884-5090

ABSTRACT

The accuracy of a finite element model for design and analysis of a metal forging operation is limited by the incorporated material model's ability to predict deformation behavior over a wide range of operating conditions. Current rheological models prove deficient in several respects due to the difficulty in establishing complicated relations between many parameters. More recently, artificial neural networks (ANN) have been suggested as an effective means to overcome these difficulties.

To this end, a robust ANN with the ability to determine flow stresses based on strain, strain rate, and temperature is developed and linked with finite element code. Comparisons of this novel method with conventional means are carried out to demonstrate the advantages of this approach.

INTRODUCTION

Finite element modeling of manufacturing processes has been gaining wider acceptance over the last several years. Modeling prior to the start of actual production can save considerable time, effort, and money. While modeling may provide these benefits, it must be kept in mind that finite element software can only provide accurate simulations of a "real" process if appropriate material models are utilized.

In this paper a novel material model is presented and compared to conventional models. For lack of an exact mathematical model, an intelligent algorithm, the ANN, will be used to map relationships between the hot forging parameters and the flow stress of the material. The ANN learns the patterns and offers robust and adaptive processing capabilities by implementing learning and self-organization rules.

In the present work, an ANN is generated and trained based on physical testing of 6061 aluminum as found in published literature [1]. This trained network is then used as the material model, which when linked with a commercial finite element code, provides a model capable of more accurately reflecting actual experience. Much of this results from a robust ANN's ability to predict outputs between, and to some degree, outside the bounds established by a training set. For this application, values of strain, strain rate, and temperature not matching the family of curves used for training can be submitted to the network, and intermediate values of flow stress found.

The conventional modeling approach requires experimental data curves to be fit to some form of hardening law and intermediate values interpolated by some means. While this may not be particularly difficult, the curve fitting process itself can be exceptionally tedious and in many cases does not produce accurate fits of the data. The ANN is much simpler to implement. Set up the network, train it, submit input values, and output is generated.

To these ends, initially, a review of conventional material models and their limitations will be presented. The various factors leading to difficulties in addressing real problems will be summarized. The development and use of artificial neural networks will be covered with the specific aim of developing an unconventional material model for linking with finite element code.

Conventional and ANN-based material models are then developed for 6061 aluminum using published data. The training of the ANNs is accomplished using MATLAB's Neural Network Toolbox. The conventional model is used directly with the commercial finite element code, ABAQUS. The ANN-

based model requires the generation of Fortran code that is linked by means of a subroutine within ABAQUS.

BACKGROUND

Conventional Material Modeling

Many manufacturing operations require material deformation well beyond the elastic limit and as such require elasto-plastic models. For the most part, the models differ in their approach to describing the plastic portions of deformation (i.e. hardening behavior). Typically, finite element codes treat the elastic and plastic portions of strain separately as shown in Eq. (1).

$$\varepsilon = \varepsilon_e + \varepsilon_p = \frac{\sigma}{E} + \varepsilon_p \quad (\text{EQ 1})$$

Several significant difficulties arise when attempting to model material behavior beyond yield. As shown in Fig. 1, real materials may exhibit difficult to characterize behavior once plastic deformation begins to take place. While many materials display strain flattening under a narrow set of conditions, other behaviors result when outside that range. Strain softening and/or oscillating flow stresses may occur due to dynamic recrystallation, recovery, or other somewhat poorly understood phenomenon.

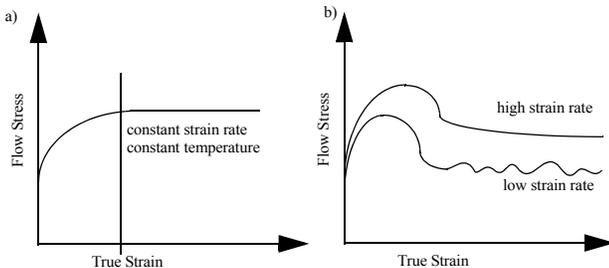


Figure 1. a) Strain flattening and b) strain softening behavior [2].

Typically, forging processes assume that flow stresses, σ_f , follow some form of the power law. Again treating elastic and plastic strains separately, the Ramberg-Osgood relation (Eq. (2)) incorporates the power law treatment for plastic portion of deformation. The reference stress, K , and strain hardening exponent, n are determined through curve fitting a stress/strain (σ_f/ε) diagram obtained from compression testing the material to be modeled.

$$\varepsilon = \frac{\sigma}{E} + \left(\frac{\sigma_f}{K}\right)^{\frac{1}{n}} \quad (\text{EQ 2})$$

If the material is being hot worked (i.e., many forging processes) the strain rate is substituted for strain as given by Eq. (3).

$$\dot{\varepsilon} = \left(\frac{\sigma_f}{C}\right)^{\frac{1}{m}} \quad (\text{EQ 3})$$

To more accurately model behavior at elevated temperature the kinetic rate equation Eq. (4) is frequently employed:

$$\dot{\varepsilon} = A\sigma_f^n e^{-\frac{Q}{RT}} \quad (\text{EQ 4})$$

where ε is the true strain rate as before, A a constant, Q is the activation energy, R the universal gas constant, and T the temperature is Kelvin.

Sellars and Tegart [3], suggest a somewhat more complicated form Eq. (5) based on deformation as a thermally activated process:

$$\dot{\varepsilon} = A(\sinh \alpha \sigma_f)^n e^{-\frac{Q}{RT}} \quad (\text{EQ 5})$$

where an additional constant α is required. Other more complicated forms of the power law have been suggested to more adequately model behavior [4][5]. Severe difficulties arise in fitting the above equations to actual test data. For each of the above cases, compression tests are performed and the constants determined through curve fitting. Constant strain rate testing must be employed using several different strain rates and temperatures [6][7].

When constitutive models are used, one of the above equations is fit to empirical data. Obviously, some are easier to fit than others. Eq. (5) requires that four constants be determined, and depending on the methods used, considerable errors may be generated [8]. In general, the fits are only properly obtained using steady state stresses.

Neural Networks

Over the last decade, several artificial intelligence tools such as artificial neural networks (ANN), fuzzy logic, and genetic algorithms (GA) have been introduced and applied in the field of manufacturing process engineering [9][10]. They provide for more accurate models than the available analytical ones. More recently, artificial neural networks (ANN) have been proposed to describe the material flow stress under the considered processing conditions [11][12][13][14].

The general idea behind artificial neural networks is to emulate the signal processing scheme used by nature. Several dendrites accept input that a given neuron processes before exiting at the axon. The axon then in-turn transmits a signal to another neuron's dendrites. In this way information is processed or modified appropriately as it pass through the nervous system. Artificial neural networks have performance characteristics similar to biological neural networks and are based on the following assumptions [15]:

- Information processing occurs at many simple elements called neurons.
- Signals are passed between neurons over connection links.
- Each connection link has an associated weight, which, in a typical neural net, multiplies the signal transmitted.
- Each neuron applies an activation function (usually nonlinear) to its net input (sum of weighted input signals) to determine its output signal.

A schematic of a simple multilayer artificial neural network is shown in Fig. 2. Each of the inputs is connected to each of the first hidden layer neurons and each of the first hidden layer neurons connects to each of the second hidden layer neurons. Finally, the second hidden layer combines to form a single output.

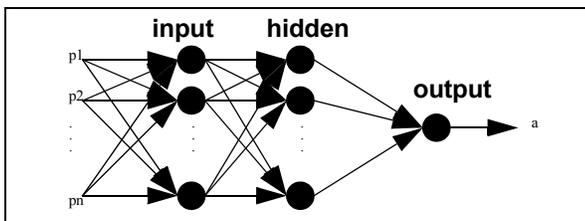
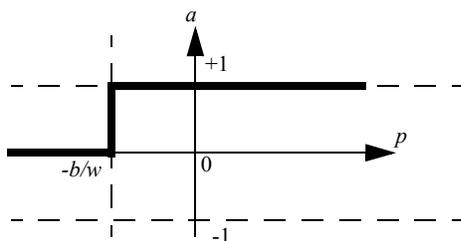


Figure 2. Schematic of a simple artificial neural network architecture.

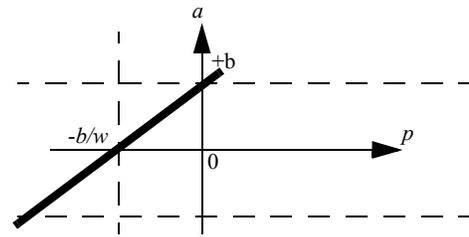
The ability of a network to mirror behavior results from the number of neurons, the number of layers, their interconnectedness, and by the transfer or activation functions chosen. Transfer functions vary from simple McCulloch-Pitts neurons [16] as given by Eq. (6) and Fig. 3 to those provided by Fig. 4, Fig. 5, and Eq. (7).

$$a = \text{hardlim}(n) = \begin{cases} 1 & \text{if } n \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (\text{EQ 6})$$



$$a = \text{hardlim}(wp+b)$$

Figure 3. Hardlim transfer function within MATLAB, where \mathbf{W} is the weight(s), \mathbf{p} the input(s), and \mathbf{b} the bias.

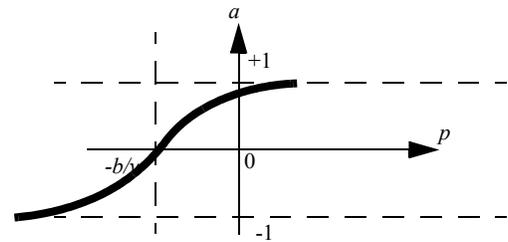


$$a = \text{purelin}(wp+b)$$

Figure 4. Linear transfer function, single-input purelin neuron.

Another commonly employed transfer function, tan-sigmoid, is provided by Eq. (7) and is shown graphically in Fig. 5.

$$a = \frac{2}{1 + e^{-2n}} - 1 \quad (\text{EQ 7})$$



$$a = \text{tansig}(wp+b)$$

Figure 5. Tan-sigmoid transfer function, single-input tansig neuron.

As an example, the material model required for forging requires the determination of flow stress as it depends on strain, strain rate, and temperature; Eq. (8) shows the matrix form for the hidden layer with strain, strain rate, and temperature as inputs.

$$\text{tansig} \left(\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ \dots & \dots & \dots \\ w_{s1} & w_{s2} & w_{s3} \end{bmatrix} \begin{bmatrix} \varepsilon \\ \dot{\varepsilon} \\ T \end{bmatrix} + \begin{bmatrix} b_{11} \\ b_{12} \\ \dots \\ b_{1s} \end{bmatrix} \right) = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_s \end{bmatrix} \quad (\text{EQ 8})$$

The subscript s refers to the particular neuron (i.e., s equals the number of neurons per input value in a layer). The hidden layer values, a , are then fed into the output layer as shown in Eq. (9) which results in a single value for the flow stress. The dot product in this case functions the same as a linear transfer function.

$$\sigma = [w_{2_1} \ w_{2_2} \ \dots \ w_{2_s}] \cdot \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_s \end{bmatrix} + b_2 \quad (\text{EQ 9})$$

Feedforward Backpropagation

Feedforward backpropagation (FBP) networks are commonly utilized for function approximation. The presentation above describes the feedforward portion of the network. Backpropagation refers to the particular method of adjusting or correcting the weights and biases to produce a network output consistent with the training set. This is accomplished by presenting a training set consisting of inputs that result in a known output. The weights and biases are initialized, the inputs present and an output determined. The output is compared against the target and the error determined. This error is used to adjust the weights and biases starting with the last layer and working backwards through the network. This procedure is repeated until an acceptable error is achieved. Several backpropagation schemes are available. The Levenberg-Marquardt (LM) algorithm, a variation of Newton's methods is typically efficient, from a computational perspective, and has proven to provide reasonable results [17].

Modeling Difficulties

If the number of neurons is excessive, or training carried out for too many epochs, the network may produce wild swings developing a greater number of inflections than the data producing overfitting (see Fig. 6).

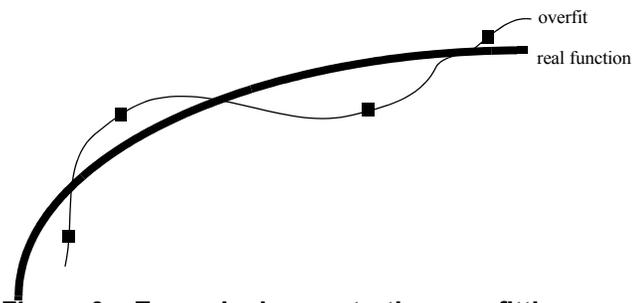


Figure 6. Example demonstrating overfitting.

Convergence, another issue, can result when training produces a network that may have found a local minimum, but not a global minimum. The learning rate specified and many times the initial conditions (i.e., starting values of the weights and biases) can produce a non-ideal solution as multi-layer networks may have many local minima.

If prior knowledge about possible relations between particular input variables and expected output exists, it is many times beneficial to include those relations through modification of the inputs. In the case of flow stresses, it is known that the log of the

stress may relate linearly to the log of the strain rate. So, it may make sense to provide the network with log strains or strain rate in addition to or instead of strains and strain rates.

Additionally scaling or normalizing the input variables and/or training values may enable the transfer functions to better handle the data, though with the added complication of re-processing the output from the trained network.

Bayesian Regularization

Bayesian regularization (BR) addresses overfitting, or improving generalization, by attempting to reduce the model complexity to the minimum necessary for reasonable performance. Bayesian methods automatically incorporate the principle of "Occam's razor" which states that the simplest model that adequately fits the data should be preferred over more complex representations [18][19][20].

Bayesian regularization adds a term to the performance function, or squared errors, used when compared targets to ANN output. If E_D represents the squared error and E_W the sum of squares of the weights, then a modified performance index can be developed Eq. (10).

$$F = \beta E_D + \alpha E_W \quad (\text{EQ 10})$$

where α and β are objective function parameters. As β grows larger and α grows smaller, then network errors are forced to be smaller. If the reverse is true, training attempts to minimize the squared weights. Decreasing the values of the weights aids in smoothing the network response and should improve generalization.

6061 ALUMINUM MODEL DEVELOPMENT

The material models that follow, conventional and ANN, are both developed from published literature (Prasad, et al., 1997). For comparison purposes, an FEA model of the simple compression of a billet, 0.015 m tall by 0.010 m diameter, was carried out at 450°C. The compression is performed between two rigid dies with a coefficient of friction of 0.15 at the interface. For purposes of simplifying the model, the process was considered isothermal without adiabatic heat rise. Compression models for strain rates 0.01, 0.1, and 1.0 through strains of approximately 0.5 were generated for conventional and ANN material models for comparison.

The flow stress curves that follow, Fig. 7 and Fig. 8, provide data for temperatures of 300 and 550°C. The curves were digitized to provide numerical input for the ANN training. In addition, the same literature provides tabular flow stress data for several strains and strain rates at 450°C as shown in Table 1.

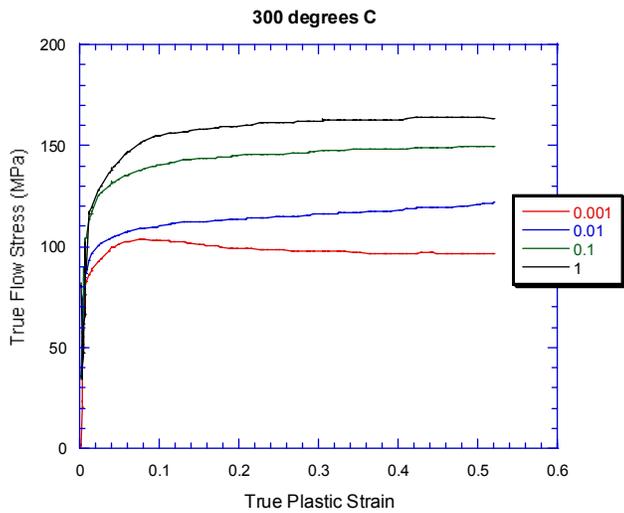


Figure 7. 6061 aluminum flow stress as a function of strain for 300°C.

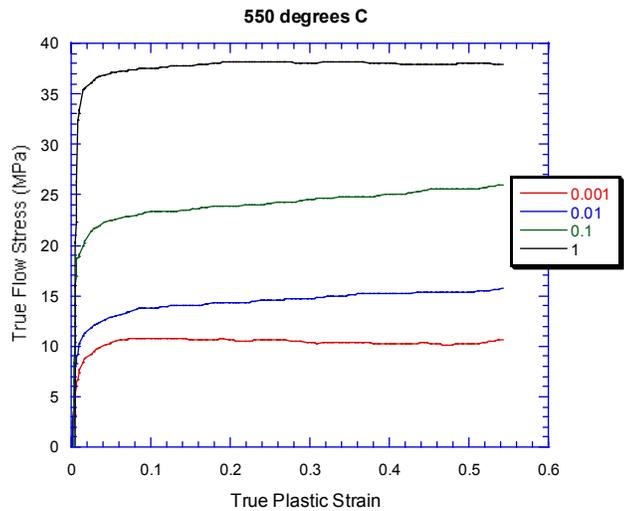


Figure 8. 6061 aluminum flow stress as a function of strain for 550°C.

Table 1: Flow stress (MPa) at 450°C.

Rate/Strain	0.1	0.2	0.3	0.4	0.5
0.001	19.6	19.5	19.8	19.9	20.4
0.01	30.2	30.9	31.5	32.1	32.6
0.1	42.3	43.0	44.4	45.1	45.6
1.0	60.5	63.0	64.5	64.4	64.3

Conventional Material Model

ABAQUS provides two conventional material models; one using the power law approach and another by directly inputting tabular values for strain, strain rate, and flow stress as provided above. ABAQUS determines strain rate dependence using the overstress power law as given by Eq. (11), a slight variation on Eq. (3). The values for the coefficient, 3.83, and exponent 0.33, were determined based on Fig. 9 using 0.3 strain as generated from Table 1.

$$\dot{\epsilon}_{plastic} = D \left(\frac{\sigma_{flow}}{\sigma_{static\ yield}} - 1 \right)^p \text{ for } (\sigma_{flow} \geq \sigma_{static\ yield}) \quad (\text{EQ 11})$$

For implementation of either the overstress power law or the tabular based model, ABAQUS requires estimation of static yield stress and elastic modulus, in this case at 450°C. Values of 15 MPa, and 54 GPa were used for static yield and modulus, respectively.

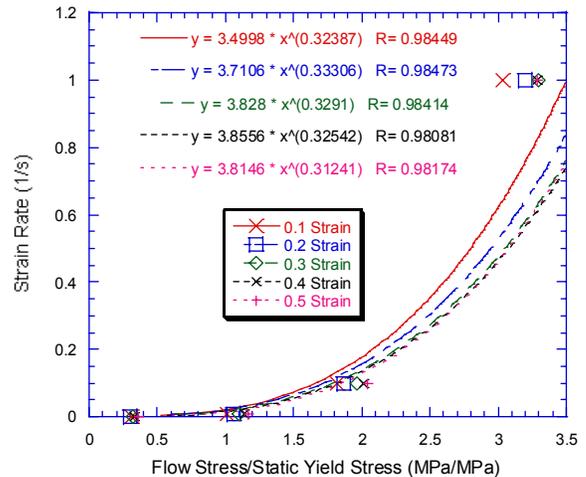


Figure 9. Curve fits for all strains at 450°C.

Neural Network Development

As a starting point, values extracted from the true stress versus true plastic strain curves, Fig. 7 and 8, were used for training several networks with varying numbers of neurons. The initial training set is comprised of 1008 total data points, split evenly for 300°C and 550°C, with equal numbers for points for each strain rate (i.e., 126 points for each separate curve, four curves per temperature). An additional set of tabular training data consisting of flow stresses at strains of 0.1, 0.2, 0.3, 0.4, and 0.5 for strain rate of 0.001, 0.01, 0.1, and 1 for temperatures of 300, 350, 400, 450, 500, and 550°C were also supplied for training once deemed necessary.

Table 2 below shows various conditions of the data as modified prior to presentation to a network. Table 3 shows a sample of the results obtained for selected attempts at training using differing data sets, algorithms, and networks architectures

along with the "best" results obtained from five training attempts under each condition, basically determined through inspection of the curves produced.

The figures that follow, show the progression of increased network performance as the data is modified due to anticipated relationships and as the network becomes more complex. Bayesian Regularization is also employed in a effort to prevent overfitting and establish the appropriate number of parameters.

The first set of figures (Fig. 10 through 13) show the performance of the 50 neuron LM trained network which produces reasonable results for the training set as demonstrated by the very low mean squared error and the R value of 1. When presented with intermediate values of strain rate, wild swings occur, indicating a lack of robustness.

The 8-3 neurons BR trained network (Fig. 14 through 19) produces better response for the training set and intermediate values of strain and strain rate, but not for intermediate temperatures.

The final resulting network of 15 input neurons followed by a second hidden layer of 3 neurons using BR training (Fig. 20 through Fig. 26) has the ability to almost perfectly match the targets when queried using the training input values. It also produces very accurate results when values of strain from 0 to 0.5 at 0.001 intervals are supplied. When intermediate values of strain rate (0.005, 0.05, 0.5, and 5) are submitted, the curves produced appear to reflect that which would be anticipated based on experience. The final trial, establishes the network output for temperatures of 375 and 450°C at 0.001, 0.01, 0.1, and 1.0 values of strain rate.

Table 2: Training sets.

Data Set	Data Condition
A	$\ln(\epsilon), \ln(\dot{\epsilon}), \ln(\sigma), \frac{1}{T}$
B	normalized values of $\ln(\epsilon), \ln(\dot{\epsilon}), \ln(\sigma), \frac{1}{T}$
C	normalized values of $\ln(\epsilon), \ln(\dot{\epsilon}), \ln(\sigma), \frac{1}{T}$, plus tabular data

Table 3: Training attempts.

Data Set	BP Algorithm	Neurons	Epochs	MSE	Parameters	R
A	LM	50	50	0.000511	NA	1.000
B	BR	8-3	500	0.103	51.4/63	1.000
C	BR	15-3	4575	0.287	88.57/112	0.999

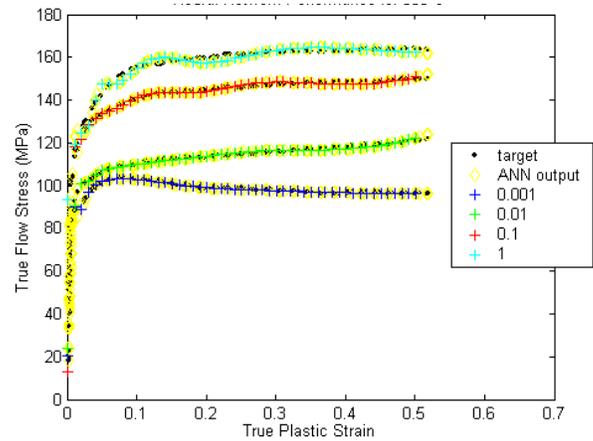


Figure 10. 50 neuron LM network output using data set A for 300°C.

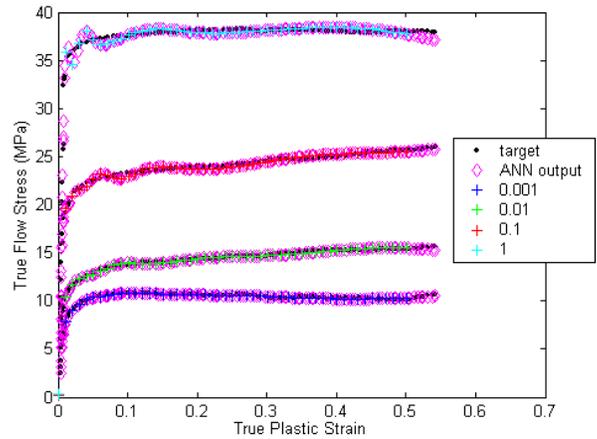


Figure 11. 50 neuron LM network output using data set A for 550°C.

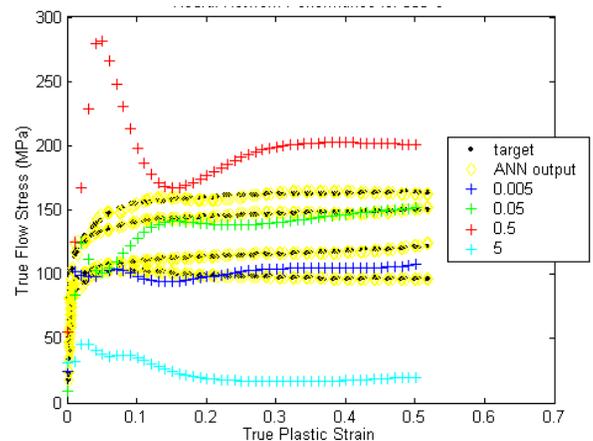


Figure 12. 50 neuron LM network output using data set A for 300°C at intermediate strain rate values.

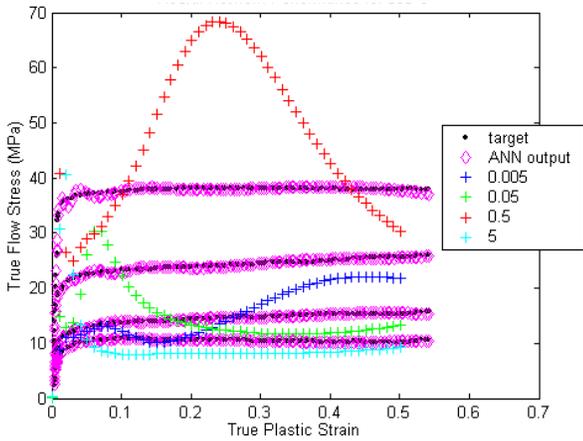


Figure 13. 50 neuron LM network output using data set A for 300°C at intermediate strain rate values.

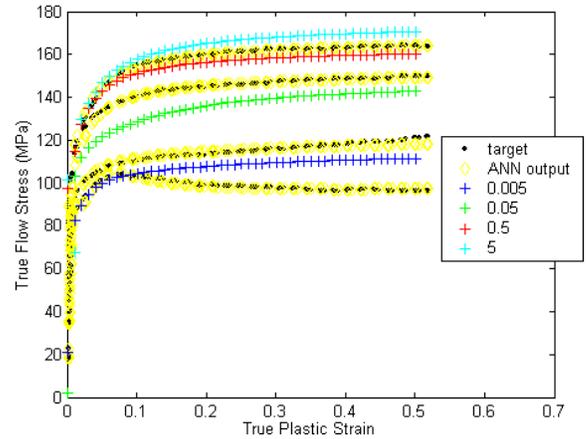


Figure 16. 8-3 BR neuron network output using data set B for 300°C at intermediate strain rate values.

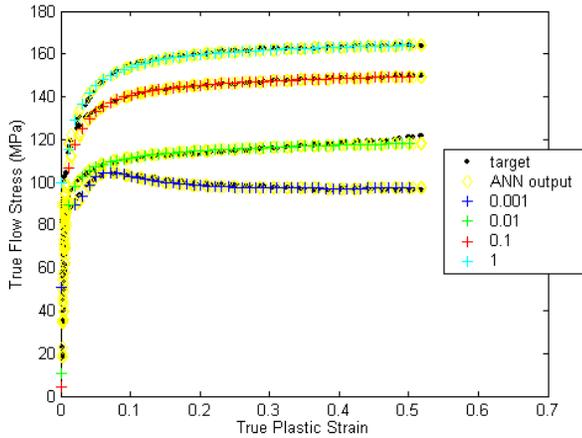


Figure 14. 8-3 BR neuron network output using data set B for 300°C.

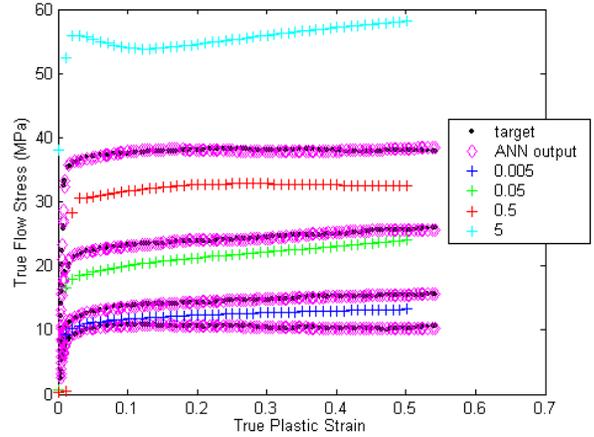


Figure 17. 8-3 BR neuron network output using data set B for 550°C at intermediate strain rate values.

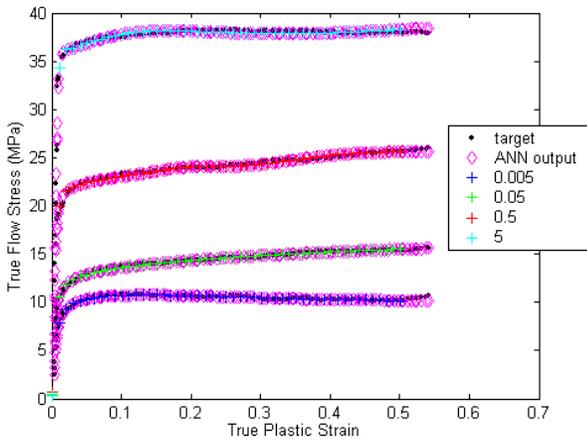


Figure 15. 8-3 BR neuron network output using data set B for 550°C.

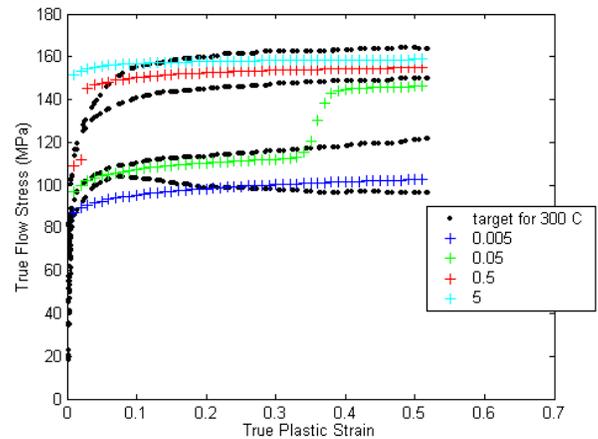


Figure 18. 8-3 BR neuron network output using data set B for 375°C at intermediate strain rates.

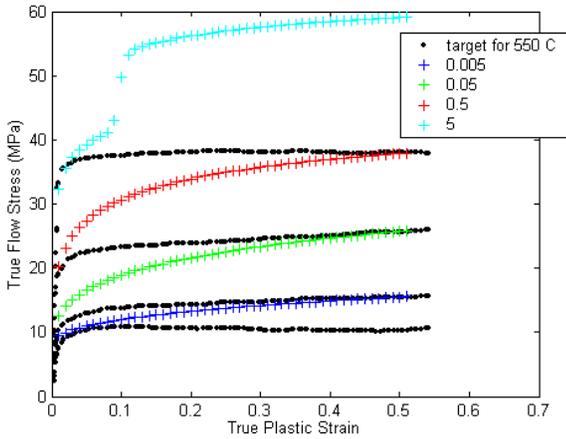


Figure 19. 8-3 BR neuron network output using data set B for 450°C at intermediate strain rates.

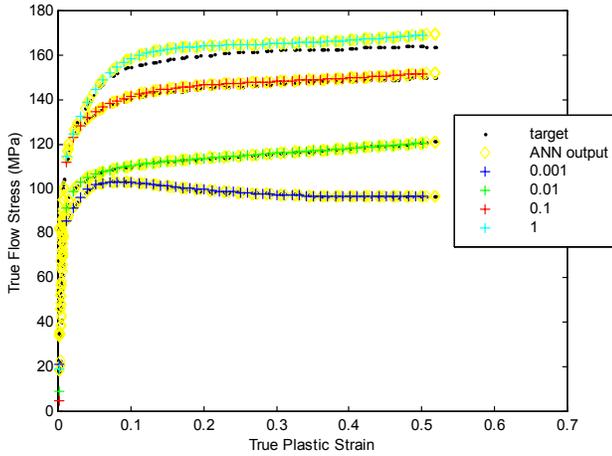


Figure 20. 15-3 BR neuron network output using data set C for 300°C.

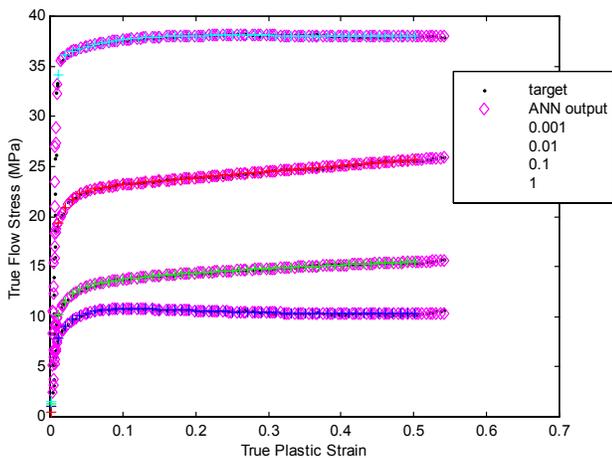


Figure 21. 15-3 BR neuron network output using data set C for 550°C

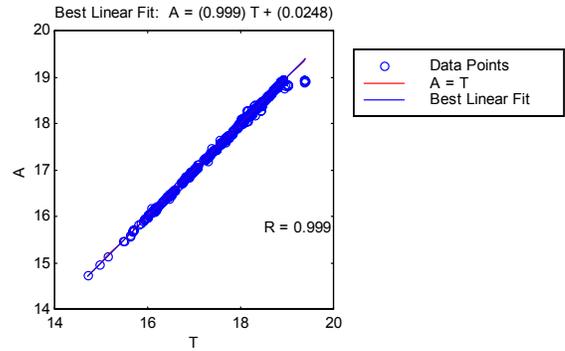


Figure 22. Linear regression for the 15-3 BR neuron network.

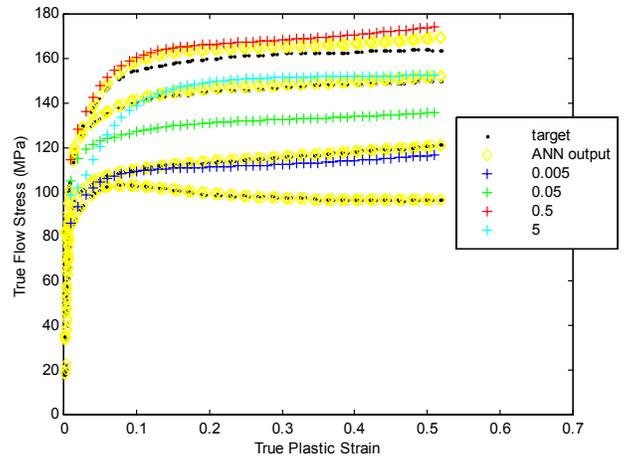


Figure 23. 15-3 BR neuron network output using data set C for 300°C at intermediate strain rates.

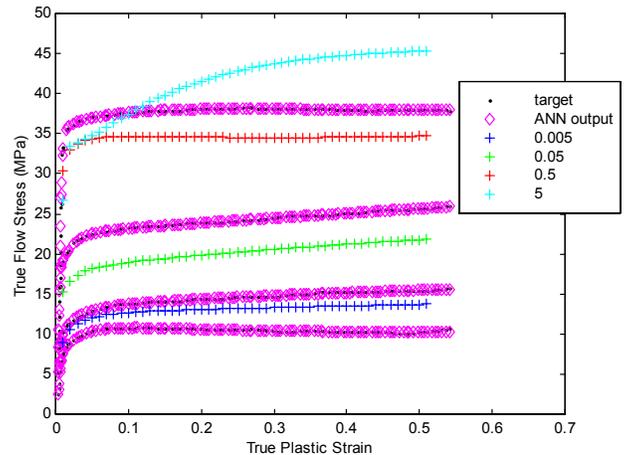


Figure 24. 15-3 BR neuron network output using data set C for 550°C at intermediate strain rates.

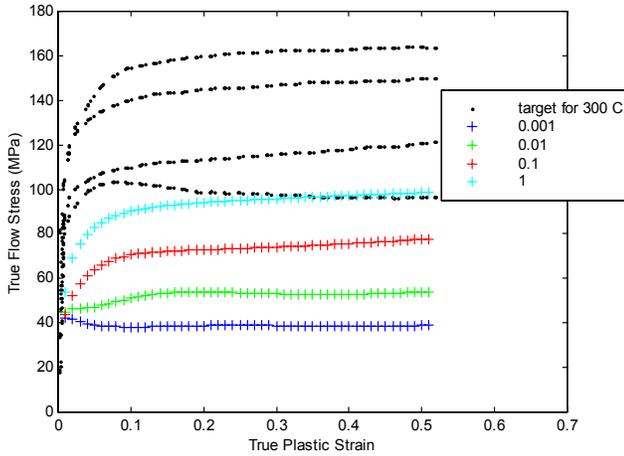


Figure 25. 15-3 BR neuron network output using data set C for 375°C.

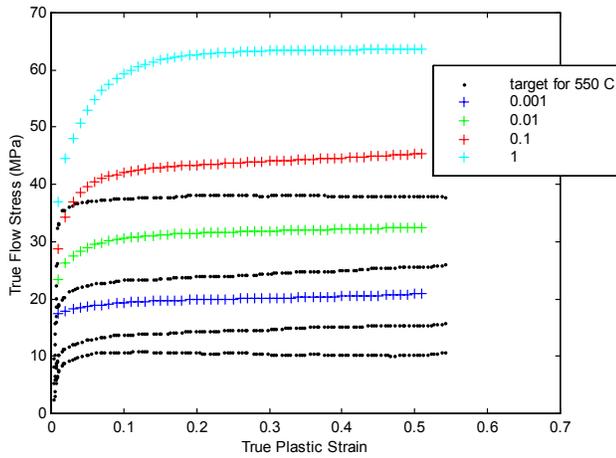


Figure 26. 15-3 BR neuron network output using data set C for 450°C.

ANN IMPLEMENTATION WITHIN FINITE ELEMENT COMPARISON WITH CONVENTIONAL MATERIAL MODELS

The weights and biases from the 15-3 neural network developed above were input into a FORTRAN program that produces the feedforward portion of the ANN. ABAQUS, through its VUMAT capability calls the program, which includes definitions for the yield surface (based on Von Mises stresses), the flow rule, and the evolution law (i.e. hardening behavior).

Figure 27 through Fig. 29 provide a comparison of a finite element model using the conventional power law and the tabular data based approach as provided by ABAQUS with the linked ANN material model approach. The figures show true stress strain curves generated by each modeling approach along with experimental test results from the published literature.

The figures clearly demonstrate that the ANN material model

possesses a superior ability to mirror experimental results. The power law model only produces reasonable results once significant strain has occurred and for only a rate of 0.1. This results from the curve fit straying from the actual values at lower and higher rates. The tabular based model provides a more reasonable approximation, but it should be noted that values of static yield stress and elastic modulus had to be estimated for input into the model. The tabular model also yields a less smooth or rounded curve as might be expected when examining experimental data at 300 or 550°C. The ANN model does not require yield estimates as it has the ability to supply flow stress values for the entire range of strain.

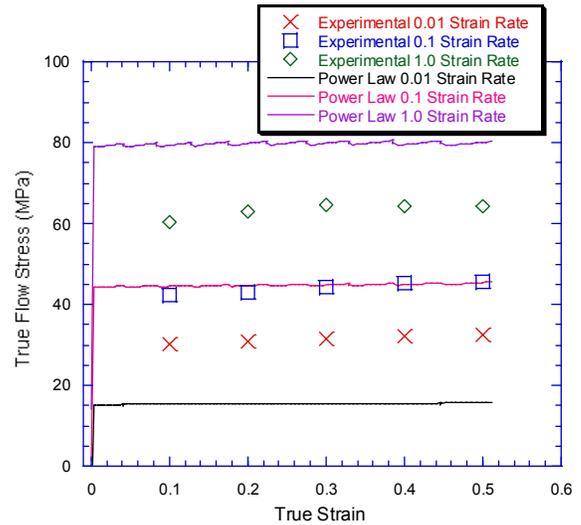


Figure 27. Compression at 450°C for power law model compared with experimental data.

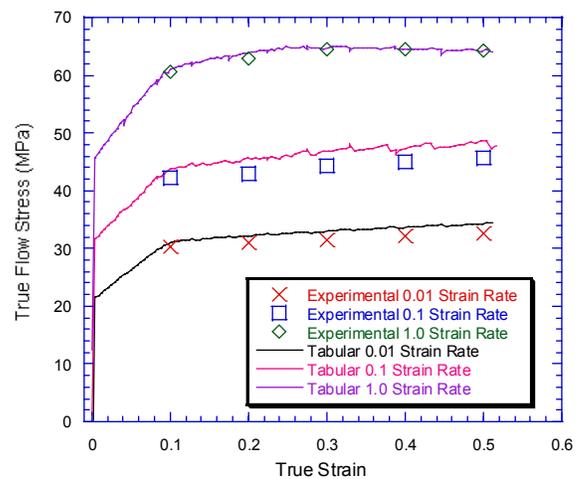


Figure 28. Compression at 450°C for tabular model compared with experimental data.

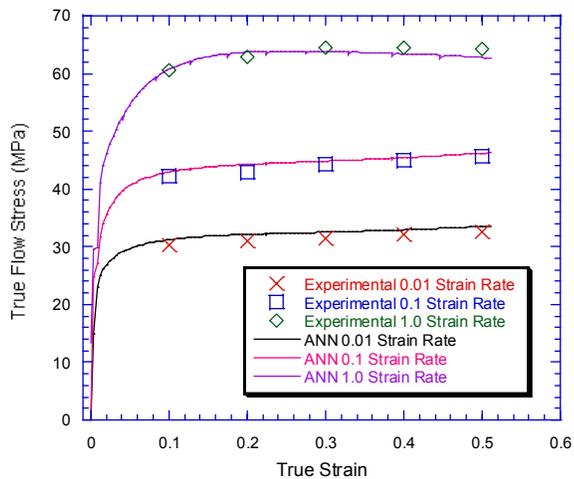


Figure 29. Compression at 450°C for ANN model compared with experimental data.

CONCLUSIONS

The above work demonstrates the ability of an ANN material model, when implemented within a commercial finite element code, to produce virtual models more closely matching experimental experience as compared to conventional material modeling methods.

REFERENCES

- [1] Prasad, Y.V.R.K., Sasidhara, S., Gegel, H.L., and Malas, J.C., 1997, *Hot Working Guide: A Compendium of Processing Maps*, ASM International, p 104-105.
- [2] Lange, Kurt, 1985, *Handbook of Metal Forming*, McGraw-Hill, New York, p 3.21.
- [3] Sellars, C.M. and Tegart, W.J.McG., 1963, *Mem. Sci. Rev. Metall.*, **63**, p 86.
- [4] Laasroui, A., and Jonas, J.J., 1966, "Prediction of Steel Flow Stresses at High Temperatures and Strain Rates," *Metall. Trans. A*, **22**, p 1545-1558.
- [5] Brown, S.B., Kim, K.H., and Anand, L., 1989, "An Internal Variable Constitutive Model for Hot Working of Metals," *Int. J. Plast.*, **5**, p 95-130.
- [6] "Standard Test Methods of Compression Testing Metallic Materials at Room Temperature," ASTM E9-89a, *Annual Book of ASTM Standards*.
- [7] "Standard Practice for Compression Tests of Metallic Materials at Elevated Temperatures with Conventional or Rapid Heating Rates and Strain Rates," ASTM E209-00, *Annual Book of ASTM Standards*.
- [8] Zhao, Dan, 2000, "Testing for Deformation Modeling," *ASM Handbook*, ASM International, **8**, p 798-809.
- [9] Rao, K.P., and Prasad, Y., 1990, "Neural Network Approach

to Flow Stress Evaluation in Hot Deformation," *Journal of Materials Processing Technology*, **53**, p 552-566.

- [10] Hodgson, P.D., Kong, L.X., and Davies, C., 1999, "The Prediction of the Hot Strength in Steels with an Integrated Phenomenological and Artificial Neural Network Model," *Journal of Materials Processing Technology*, **87**, p 131-138.
- [11] Enemuoh, E.U., and El-Gizawy, A. Sherif, 2002, "A Robust Neural Network Model for Online Prediction of Process-Induced Damage in Drilling Epoxy Composites," *Proceeding 3rd CIRP Int. Sem. On Intelligent Computation in Mfg. Eng.*, Naples, Italy, p 183-188.
- [12] El-Gizawy, A.S., Hwang, J.Y., and Brewer, D.H., 1990, "A Strategy for Integrating Product and Process Design of Aerospace Components," *Manufacturing Review*, ASME, **3**, p 178-186.
- [13] Enemuoh, E.U., and El-Gizawy, A. Sherif, 2002, "A Robust Neural Network Model for Online Prediction of Process-Induced Damage in Drilling Epoxy Composites," *Proceeding 3rd CIRP Int. Sem. On Intelligent Computation in Mfg. Eng.*, Naples, Italy, p 183-188.
- [14] Bariani, P.F., Bruschi, S., and Negro, T.Dal, 2002, "Neural Networks to Describe Superalloys Rheological Behavior Under Hot Working Conditions," *Proceeding 3rd CIRP Int. Sem. On Intelligent Computation in Mfg. Eng.*, Naples, Italy, p 585-588.
- [15] Fausett, L., 1994, *Fundamentals of Neural Networks, Architectures, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs, NJ, p xiii.
- [16] McCulloch, W. and Pitts, W., 1943, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, **5**, p 115-133.
- [17] Hagan, Martin T., Demuth, Howard B., and Beale, Mark, 1996, *Neural Network Design*, PWS Publishing, p 12-19.
- [18] Jeffreys, H., 1939, *Theory of Probability*, Oxford University Press, Oxford.
- [19] Gull, S.F., 1988, "Bayesian inductive inference and maximum entropy," *Maximum Entropy and Bayesian Methods in Science and Engineering, Vol. 1: Foundations*, G.J.Erickson and C.R. Smith, eds., Kluwer, Dordrecht, p 53-74.
- [20] MacKay, David J.C., 1992, "Bayesian Interpolation," *Neural Computation*, Massachusetts Institute of Technology, **4**, p. 415-447.