

Constitutive Modelling of INCONEL 718 using Artificial Neural Network

Olufunmini Abiri and Bhakisipho Twala

Institute of Intelligent Systems, University of Johannesburg, Auckland Park 2006, South Africa

olufunminiya@uj.ac.za

Abstract. Artificial neural network is used to model INCONEL 718 in this paper. The model accounts for precipitate hardening in the alloy. The input variables for the neural network model are strain, strain rate, temperature and microstructure state. The output variable is the flow stress. The early stopping technique is combined with Bayesian regularization process in training the network. Sample and non-sample measurement data were taken from the literature. The model predictions of flow stress of the alloy are in good agreement with experimental measurements.

1. Introduction

INCONEL 718 (IN718) is commonly used in power plants, gas turbines and aircraft engines. The microstructure of IN718 is composed of coherent ordered disk-shaped body centred tetragonal (bct) γ'' phase that comprises of Ni₃Nb and nickel [1] and the ordered fcc γ' precipitates. The volume fraction of the γ'' and γ' phases are 10 - 20% and 3 - 5% respectively depending on heat treatment and the bulk alloy composition [2]. Strengthening of the alloy is mainly due to the γ'' phase. Modelling of the flow stress of the alloy therefore need additional consideration of the microstructural features such as grain size, precipitate size and volume fraction of the precipitates. Physical and phenomenological models of the alloy are the dislocation density model of Fisk et. al. [1] and phenomenological model of [3] respectively. Artificial neural network (ANN) offers an alternative approach. A background of ANN in modelling materials behaviour can be found in [4]. ANN modelling involves using an arbitrary number of computational processing units that learn from observed material data. Thus, through use of these parallel computational units, it can capture complex nonlinear relationships between inputs and outputs [5].

This work is a first attempt to use ANN to model the flow stress of IN 718 accounting for the microstructure. In the ANN model development, early stopping techniques together with Bayesian regularization process [6,7] are employed. Flow stress measurement data taken from [1] are utilized in developing the ANN model. Strain (ϵ), strain rate ($\dot{\epsilon}$), temperature (T) and microstructure state of the material (Age) are used as input variables of the ANN network and the flow stress (σ) is the output variable. The ageing time variable represents the microstructural state of the alloy. Predicted flow stress results are compared with the experimental data of different microstructural state of the



alloy not used in the ANN model development. The results indicate good agreement with experiments and therefore show the reliability of the ANN constitutive model.

2. Artificial neural network

A human brain is a complex biological neural network with many interconnected set of neurons that facilitate daily activities of human life[8]. An ANN is a mathematical model involving a group of interconnected artificial neurons that tries to simulate the neural structure of the human brain. ANN are trained to learn patterns from data sample. This learning process, also called a training process, is used to determine the neural network parameters for the model. This is similar to curve fitting in linear regression. The most popular type of neural network is a multilayer perceptron (MLP) with back propagation [9]. An MLP has layers of units: inputs layer, hidden layer or layers and an output layer. Independent processing units called neurons connect the layers. The number of neurons in the input and output layers is determined by the number of input and output features respectively. The number of hidden layer or layers depends the training process. There are no scientific or statistical methods yet to determine the number of neurons in the hidden layer or layers. The training algorithms are typically improved using heuristic methods [10,11].

In ANN, the output mapping from the input is given by

$$f(x; \omega; w) = g_2 \left(\sum_{j=0}^N \omega_j g_1 \left(\sum_{i=0}^k w_{ji} x_i \right) \right) \quad (1)$$

where $g_1(\cdot)$ and $g_2(\cdot)$ are activation functions also known as transfer functions. w and ω are network weight parameters. The activation function mathematically defines the non-linear relationship between inputs and output of a neuron in a neural network. For the training set of data comprising a set of input x_n , $n = 1, \dots, N$ and a corresponding set of target vectors t_n , the objective of the model is to minimize the sum of square error function given by

$$e(w) = \frac{1}{2} \sum_n \left\| (y(x_n; w) - t_n) \right\|^2 \quad (2)$$

In Equation (2), t_n is the desired or target output for the n th neuron when the n th input pattern is presented to the network and $y(x_n, w)$ is the actual output of the n th output neuron when the input pattern is presented. This type of learning process is known as supervised learning [12] in which every input pattern has its associated target output. The goal is to find a vector of weights such as Equation (2) takes its smallest value.

Given an input value that is not in the training set, the trained network can predict the most likely output value. This capacity of the network to be able to determine the output for an input the network was not trained with is known as generalization. Bayesian regularization improves generalization of the model by automatically incorporating model uncertainty thereby imposing prior probability distribution on the model parameters. Using Bayesian regularization, the error function $e(w)$ given by Equation (2) is modified by adding a term. The new error function, which can be called augmented error function $e_{aug}(w)$ becomes

$$e_{aug}(w) = e(w) + \frac{1}{2} \mu \sum_{ij} \alpha_{ij}^2 \quad (3)$$

where μ and α_{ij} are the regularization parameters known as the weight decay constant and weight connection from node j to node i respectively.

3. The dataset

The chemical composition of INCONEL 718 is presented in Table 1. The dataset used in the development of the ANN model for the alloy during thermal treatment was taken from the

relevant flow stress curves in Fisk et. al. [1]. The dataset were obtained by digitalizing the curves. The curves were obtained from compression test data of the alloy at two microstructure states of half-aged and fully aged. The ageing stages are achieved by heating the alloy to 760°C and holding for 30 minutes and 5 hours for half-aged and fully aged states respectively. ε , $\dot{\varepsilon}$, T , σ and Age are used as variables in the ANN. A total of 242 dataset is obtained from the flow stress curves. This dataset is split into the training set, validation set and the test set according to percentages for the neural network model. The training set is assigned 80%, the validation set 10 % and the test set is assigned 10%.

Table 1. Chemical composition of IN718 (wt %)[1].

Ni	Cr	Fe	Nb	Mo	Ti	Al			
50.0-	17.0-	Bal	4.75-	2.80-	0.65-	0.20-			
55.0	21.0		5.50	3.30	1.15	0.80			
Co	C	Mn	Si	P	S	B	Cu	B	Cu
Max	Max	Max	Max	Max	Max	Max	Max	0.048	0.040
1.00	0.08	0.35	0.35	0.015	0.015	0.06	0.03		

Table 2. Range of the variables in the dataset

State variables	Minimum value
ε (%)	0 – 0.7
$\dot{\varepsilon}$ (s^{-1})	0.001, 0.01, 0.1, 1
T ($^{\circ}\text{C}$)	400, 600
σ (MPa)	0 - 1800
Age (Hour)	0.5, 5

4. Model development and simulation

The ANN structure for the flow stress prediction for alloy 718 accounting for microstructural state of the material is depicted in Figure 1.

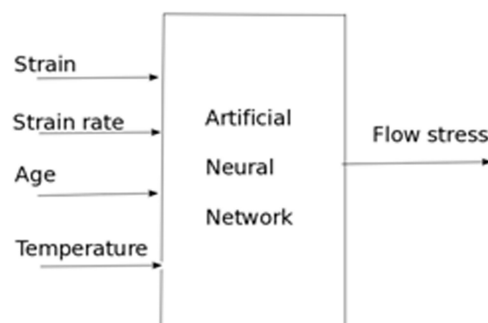


Figure 1. Schematic diagram of the ANN structure

Following the Kolmogorov theory [13], we are using a 4 – 9 – 1 feedforward neural network in Figure 1. The network input layer consists of 4 neuron nodes, the single hidden layer contains 9 nodes and the output layer has only one node. The activation function for the nodes in the hidden layer is a tangent sigmoid activation function (TANSIG) given by

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (4)$$

while the output layer used the linear activation function (PURELIN). The neural network package in the MATLAB software[14] was used to develop and execute the model. A learning rate of 0.01 was used. The number of epochs is 1000. An epoch can be defined as one presentation of all training examples to the network, followed by adjustment of the weights. The neural network was optimized according to the Bayesian regularization method. The mean square error of Equation (3) is used as the performance evaluation function. The 242 dataset from the flow stress curves in [1] is presented to the neural network. The dataset is pre-processed so that the input data will fall into the range $[-1, 1]$. The MATLAB “mapminmax” is used for this purpose. This normalization procedure ensures that larger values of the input data do not overwhelm the smaller input data. This helps to reduce the network error. After the normalization procedure, another MATLAB function, “divideint”, is used to divide the dataset into training, validation and test set. The output values were de-normalised.

5. Results and discussion

The performance curve after training the ANN is shown in Figure 2. The mean square error decreases as the number of epoch increases. The mean square error is 0.0000007, which is very small and close to the ideal value of zero. The target and ANN output after training the network for the sampled data is shown in Figure 3. The ANN model perfectly model the training data. Attempts was made to utilize the ANN model for predictions of input variables that were not in the training dataset. As Figure 4 shows, for the solution annealed IN718 at 400 °C and 0.001 s⁻¹, there is good agreement between measurement and ANN prediction. The solution annealed state was achieved by holding the alloy in a furnace at 950 °C for 1 hour [1].

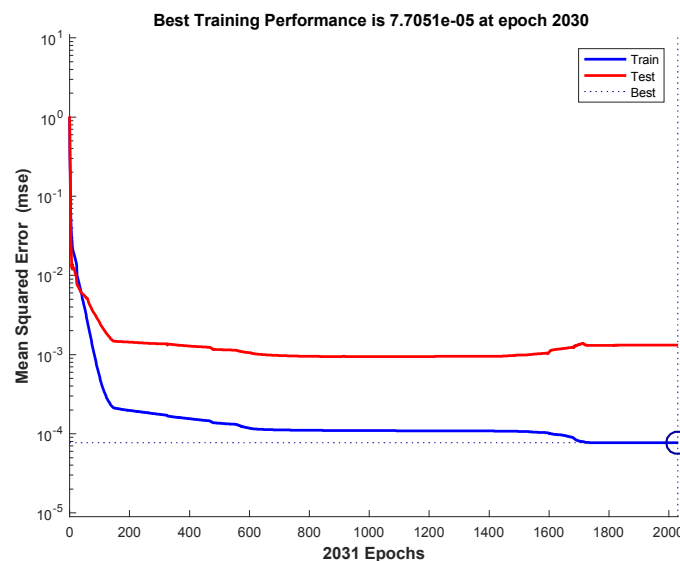


Figure 2. Performance curve for the ANN.

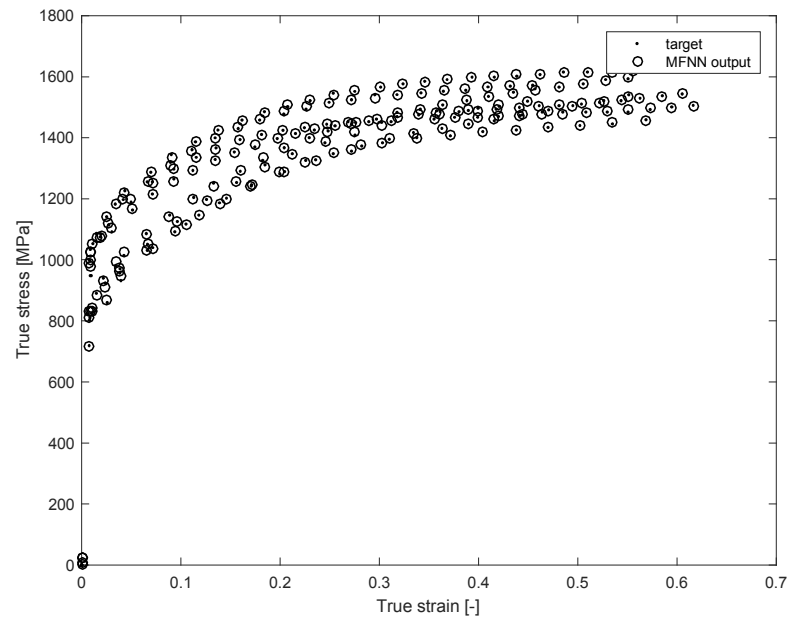


Figure 3. The ANN output versus measured flow stress for the training dataset.

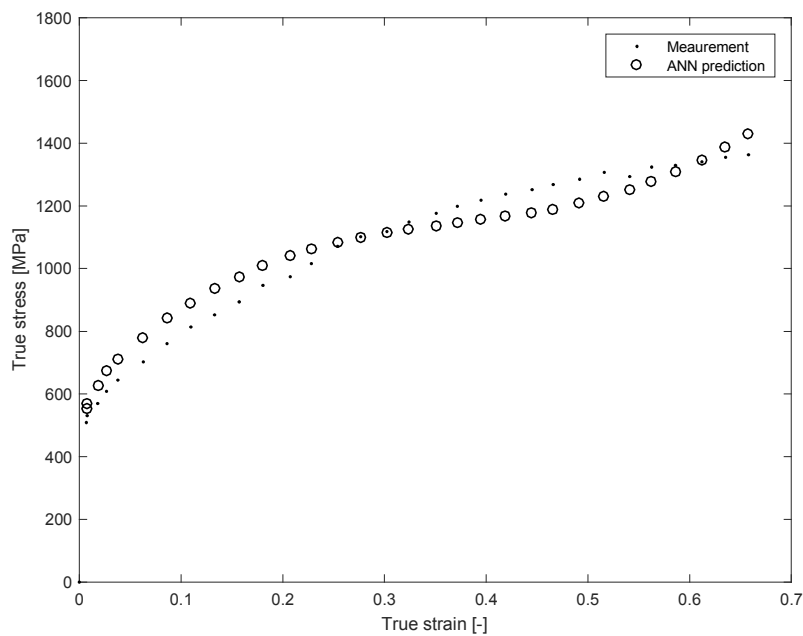


Figure 4. Comparison of ANN predicted and measured flow stress for at 400°C and 0.001 s^{-1} for solution annealed IN718.

6. Conclusions

Attempt has been made to use ANN for modelling IN718 thereby accounting for precipitate hardening of the alloy. Input variables for the ANN model are strain, strain rate, temperature and microstructural state of the alloy. The output variable is the flow stress. The ANN model predictions show good agreement with experimental measurement for the sampled data and non-sampled data. Further work will concentrate on obtaining more experimental data to use in training and thereby increase the reliability of the model for IN 718.

Acknowledgements

The authors gratefully acknowledge the financial support of the University of Johannesburg via the Global Excellence and Stature (GES) fellowship granted to the first author.

References

- [1] Fisk M, Ion JC, Lindgren L. Flow stress model for IN718 accounting for evolution of strengthening precipitates during thermal treatment. *Computational materials science* 2014;82:531-539.
- [2] Cruzado A, Gan B, Jiménez M, Barba D, Ostolaza K, Linaza A, et al. Multiscale modeling of the mechanical behavior of IN718 superalloy based on micropillar compression and computational homogenization. *Acta Materialia* 2015 10/1;98:242-253.
- [3] Iyer SK, Lissenden CJ. Multiaxial constitutive model accounting for the strength-differential in Inconel 718. *Int J Plast* 2003;19(12 SPEC ISS.):2055-2081.
- [4] Bhadeshia HH. Neural networks in materials science. *ISI Int* 1999;39(10):966-979.
- [5] Sheikh-Ahmad J, Twomey J. ANN constitutive model for high strain-rate deformation of Al 7075-T6. *J Mater Process Technol* 2007;186(1):339-345.
- [6] MacKay DJ. A practical Bayesian framework for backpropagation networks. *Neural Comput* 1992;4(3):448-472.
- [7] Duma S, Twala S and Marwala T. Predictive modeling for default risk using a multilayered feedforward neural network with Bayesian regularization. *Neural Networks (IJCNN), The 2013 International Joint Conference on: IEEE; 2013*.
- [8] Block H. The perceptron: A model for brain functioning. i. *Reviews of Modern Physics* 1962;34(1):123.
- [9] Bishop CM. *Neural networks for pattern recognition*. : Oxford university press; 1995.
- [10] An algorithm for fast convergence in training neural networks. *Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on: IEEE; 2001*.
- [11] Kalra A, Kumar S, Walia SS. ANN Training: A Survey of Classical & Soft Computing Approaches. *International Journal of Control Theory and Applications* 2016;9:715-736.
- [12] Bishop CM. *Neural networks for pattern recognition*. : Oxford university press; 1995.
- [13] Hecht-Nielsen R. Theory of the backpropagation neural network. *Neural Networks* 1988;1(Supplement-1):445-448.
- [14] Beale MH, Hagan MT, Demuth HB. *Neural network toolbox 7. User's Guide*, MathWorks 2010.