Computational Materials Science 155 (2018) 331–339

Contents lists available at ScienceDirect

Computational Materials Science

journal homepage: www.elsevier.com/locate/commatsci

Two-way design of alloys for advanced ultra supercritical plants based on machine learning

Xiaobing Hu[a], Jincheng Wang[a,*], Yanyan Wang[a], Junjie Li[a], Zhijun Wang[a], Yingying Dang[b], Yuefeng Gu[b]

[a] State Key Laboratory of Solidification Processing, Northwestern Polytechnical University, Xi'an 710072, PR China
[b] Xi'an Thermal Power Research Institute, Xingqing Road 136, Xi'an, Shaanxi 710032, PR China

ARTICLE INFO

Keywords:
Machine learning
Ultra supercritical materials
Artificial neural network
Genetic algorithm
Alloy design

ABSTRACT

Materials for advanced ultra-supercritical (A-USC) power plants with steam temperatures of 700 °C and above are extremely needed in order to achieve high efficiency and low CO_2 emissions. Alloy design based on machine learning is of great importance to explore the space to decide connections between compositions and performances. In this work, we employed an artificial neural network (ANN) in the machine learning framework to compete a two-way design which is defined by predicting the target properties and designing alloys over the dataset consisted of experimental data. Combined with Genetic Algorithm (GA), the ANN model was optimized to improve the accuracy over 98% by training and testing the full dataset. Meanwhile, the model can find the global optimization values of two performances: yield strength and creep rupture life eventually. With a true accuracy of over 90%, we designed a group of compositions of Ni based superalloy to meet the requirements of microstructures and properties for A-USC plants. Further experimental validation was also conducted, which proved that our ANN model optimized by GA can be used to predict and design superalloys for A-USC.

1. Introduction

Energy with lower carbon dioxide (CO_2) emissions is increasingly necessary to protect the global environment. The adoption of ultra-supercritical (USC) power plants with increased steam parameters can significantly improve the efficiency, which reduces fuel consumption and the emission of environmentally damaging gases. With the development of advanced coal-fired power generation technology, the working temperature and pressure have been gradually improved, resulting in a much higher requirement for high temperature strength and creep resistance of materials used in generating units [1,2]. For example, materials for advanced ultra supercritical (A-USC) plants at 750 °C require that not only the creep rupture strength corresponding to the 100 thousand hours endurance life at 750 °C should be higher than 100 MPa, but also less pollution and lower costs. There only exist very few Ni based superalloys being used in the superheater and reheat pipe of the generator, such as Inconel 740/740H [3–5]. However, the cost of Inconel 740/740H alloys is very high because of the high content of Co element. Thus, designing new and low-cost Ni-based superalloys meeting the service requirements at 750 °C is absolutely essential.

The traditional scientific methods of intuition and trial and error cannot keep up with the rapidly increasing need of modern industry for new materials. It is now common for innovation in materials research and development to accelerate the process from discovery to the application of new materials. In the traditional methods of alloy design, it is difficult to accurately reflect the complex nonlinear relationship between the components and the properties of Ni based superalloys. Besides, the number of allowed candidate alloys can be over 10^6 when the gap of their composition interval is 10% and their concentrations are measured to 0.1%. It is a hard work to search proper alloys in this vast space *via* experiment methods.

Recently, as a new method of alloys design, machine learning (ML) was used to complete the target properties, reduce the design costly and avoid dangerous operation, which can recognize data patterns and obtain insights from the data without explicit programming [6–8]. Lots of successful examples of ML in material science have certified its feasibility in alloy-design [9–15]. The artificial neural network (ANN) technique inspired from the functioning of human brain is probably the most widely used algorithm in ML, and the process of this technique depends on the functional relationship of efficient input-output dataset [16–21]. However, two issues exist in the ANN algorithm: (1) the accuracy of prediction in alloy properties with composition is high but not

* Corresponding author.
E-mail address: jchwang@nwpu.edu.cn (J. Wang).

https://doi.org/10.1016/j.commatsci.2018.09.003
Received 24 July 2018; Received in revised form 31 August 2018; Accepted 1 September 2018

in the inverse process; (2) in the learning process, the ANN is easy to fall into a local optimization. Therefore, it is essential to develop an ANN model combined with other ML algorithms to solve more questions of material science by deep learning. Genetic Algorithm (GA) inspired from the procedure of the chromosome changing in nature can find the best object by searching in the global scope and participate in the optimization of other algorithms.

In this study, we combined the ANN and Genetic Algorithm (GA) to achieve the target of two-way design of alloys and realize the global optimization. Firstly, we competed the dataset and chose 5 key features. Secondly, we adjusted the accuracy of model prediction by training and testing the model, and verified that our model can find the global optimization simultaneously. In the end, a new Ni based alloy for A-USC units was designed to certify the feasibility of two-way alloy design with the target properties, which also were verified by experiments with the composition provided by our prediction.

2. Dataset

In order to train the network, firstly, a related dataset should be set up for the A-USC superalloy. The data are mainly collected from the reported results of international symposiums on Superalloys (1996–2016) and experimental results. The dataset contains 580 rows and 5 columns. Each row is called as an instance and 5 columns called as features in the machine learning are used to describe the properties of superalloys [22]. In other words, our dataset has 580 instances and 5 features. In order to prevent the ANN model from over-fitting or just memorizing the data, the dataset was divided into training and testing data set and the test data was used to evaluate the performance of the model. If the testing error is very high compared to the training error, the network is regarded as over-fitting the data, and the model needs to be trained again until a properly fitted network is established [20]. Then the trained and tested neural network may be presented with a verification data set to evaluate the accuracy of the network and finally presented with the actual data to be processed [23]. In the dataset, data from 430 rows were used for training the network, while from other 150 rows for testing. In addition, we selected 6 alloys which were not used earlier for the purpose of training or testing to complete the validation. From Table 1, we can see that the 5 features in first 5 instances are volume fractions of γ and γ′, misfit of γ/γ′, yield strength and high temperature creep rupture life which was calculated via logarithm, respectively. In this work, the alloy components contain Ni, Al, Co, Cr, Fe, Mo, Nb, Ti, W, Si, B, and C [24–30]. The element of Ni was selected as the matrix element, and Al and Ti as elements of precipitation strengthening. In addition, the elements of solid solution strengthening consist of Co, W, Mo and Nb, corrosion elements are Cr and Si, and elements of reinforcing structure stability are B and C. The element of Fe was added to reduce the cost. What's more, the mean grain size and the gamma prime phase size were included in our database because of their important effect on the thermal process. Ultimately, the input of the network was consisted of compositions of the 14 elements and the first two features in Table 1, while the predicted error of remaining three features as output. Here, the input was considered as independent variables while the output as dependent variables of the model.

Table 1
List of the first 5 instances with 5 features used.

γ content	γ′ content	Mismatch of γ/γ′	Yield Stress	LOG(creep rupture life)
52.93	29.24	0.0123	952.74	5.16
60.56	25.6	0.0343	909.43	5.06
56.69	26.07	0.0474	860.89	5.13
53.06	29.19	0.0642	876.76	5.15
50.53	30.39	0.0307	975.52	5.16

All the units of the first 3 features are vol%, and the remaining 2 features are MPa and h, respectively.

3. Establishment of an optimizing ANN model

Before establishing the ANN model, we used the dataset above to normalize the feature values so that they can fall within the range of 0 and 1 by the following function:

$$x' = \frac{x_i - x_{\min,i}}{x_{\max,i} - x_{\min,i}} \tag{1}$$

where the x' is a normalized feature and x_i refers to the original data. $x_{max,i}$ and $x_{min,i}$ are the maximum and minimum values of the corresponding feature. This process ensures that the features inputted into the model have the same numeric scale and can be equally treated.

Among various ANN models, the most popular one is the feed-forward back-propagation neural network (BPNN). Fig. 1(a) shows a schematic representation of the neural network. The network contains three layers: input layer, hidden layer and output layer. The values of input variables in the input layer are transformed into the hidden layer and then are calculated to realize the goal of corresponding output. In this process, the hidden layers and their neurons play roles of combination and transinformation. Once the combined effect on each hidden neuron is determined, the activation at this neuron is determined *via* a transfer function. There are some common transfer function such as Sgn function and Sigmoid function that are used to connect the neurons. As shown in the Eq. (2), Sgn function gains the output of 0 or 1 via inputting the variable x, where the value 0 means neurons inhibition while the value 1 means neurons excitation. Another common function, sigmoid function, compresses the data with huge range to an interval of 0 to 1. It should be noted that the sigmoid function is a generally used transfer function though there are many other nonlinear functions can do the same thing [31]. The function sigmoid(x) is expressed by $f_1(x)$:

$$\begin{aligned} f_1(x) &= \text{sigmoid}(x) = \tfrac{1}{1+e^{-x}} \\ f_2(x) &= \text{Sgn}(x) = \begin{cases} 1, & x \geqslant 0; \\ 0, & x < 0. \end{cases} \end{aligned} \tag{2}$$

The output data is also calculated by the activation function $f_1(x)$ between the hidden layer and the output layer. During the back propagation of the signals, the weights and bias at each neuron are modified to minimize the output error. This process is known as training. Some optimization algorithm are usually used to complete the process, viz. Levenberg-Marquardt (*trainlm*), Bayesian algorithm (*trainbr*), scaled conjugate gradient (*trainscg*), resilient (*trainrp*), gradient descent (*traingd*), gradient descent with momentum (*traindm*). Levenberg-Marquardt algorithm has been found to be very efficient for ANN modeling by various investigators [19,23]. After training the model successfully and confirming the parameters in the model, the network architecture can be frozen. In the process of predicting, we used a fitness optimization function shown as Eq. (3) to evaluate the accuracy of the model [32].

$$f(A, B, C) = \frac{1}{1 + 2 \times e^{|A_E - A|}} + \frac{1}{1 + 2 \times e^{|B_E - B|}} + \frac{1}{1 + 2 \times e^{|C_E - C|}} \tag{3}$$

where A, B and C are the predicted values, and A_E, B_E and C_E are the expectation values. It can be seen that when the prediction value equals to the expectation one, the function reaches its maximum value of 1, indicating the best prediction result. On the contrary, the closer you get to 0, the worse it gets. Fig. 1(b) shows a neuron model in the neural network consisting of the input x_i', output y', weight ω_i and threshold θ. Obviously, the calculated results of the network are determined by the value of weight ω_i and threshold θ. However, whether the prediction result of ANN is local minimum or global minimum is not certain due to the random of weight and threshold. In order to gain results close to global minimum, we used the Genetic Algorithm (GA) to optimize weight and threshold in the ANN model [18].

The ANN and GA are combined into a new network showed in Fig. 2. When we provide the ANN with a dataset, the network will learn