Based on Adam Optimization Algorithm: Neural Network Model for Auto Steel Performance prediction

Wang Pengtao

Research Institute of Technology, Shougang group co. LTD, Shijingshan District, Beijing, China

wangpengtao@shougang.com.cn  wptgoodluck@163.com

Abstract—Neural network is optimized using the Adam algorithm, which introduces the first-order and second-order momentum of gradient. At the same time, learning rate is dynamically adjusted to different training stages to avoid the network from slow converging or oscillating. The three-layer neural network model is trained with a 5000 samples data set for HSLA steel performance prediction. The result shows that: compared with the traditional SGD algorithm, Adam shows efficient and stable training characteristics. After 742 epochs with initial learning rate of 0.001, the mean square error (MSE) of training set and validation set are below 90. On a data set consisting of 50 new samples, the prediction error of the yield strength and tensile strength on test set are less than 19MPa and 22MPa respectively, and the elongation prediction error is less than 3.1%.

1. Introduction

In recent years, with the rapid developing of Chinese automotive industry, auto steel market has accelerated the pace of transformation from import to localization. At present, most of the high-end steels such as high-strength steel and galvanized outer plates have been produced domestically, having made great progress in both production and quality. High-strength steel is not only an important material for automobile light weight and passenger safety, but also could reduce manufacturing costs. As one of the most mature varieties of high strength steel, with high yield ratio, formability and excellent welding performance, high strength low alloy steel(HSLA) is widely used in body-in-white parts, such as anti-collision beams, sill reinforcement plates, etc.

Under the appropriate production process, the combinations of Nb, Ti, V and C, N elements in HSLA steel, form numerous fine and dispersed precipitates, which leads to strong precipitation strengthening and grain refinement. Although high-level production equipment could achieve stable control of the rolling process, behaviors of nucleation and growth of Ti (C, N) and Nb (C, N) precipitates with high sensibility to temperature and rolling, are still difficult to control precisely[1]. Disturbance caused by hot rolling, coiling, cold rolling, annealing and other processes will lead to large performance differences in HSLA products, especially at the head and tail of the strip. Usually, it needs to cut a certain length, causing metal waste. Therefore, a more accurate prediction of product performance under certain processes can not only reduce the metal loss, but also improve product performance stability. It could also be used to replace industrial trial production by fine-tuning of ingredients and processes[2].

Performance prediction is essentially a regression issue, requiring complicated screenings and calculations of large amounts of data, which are time-consuming and labor-intensive. In modern steel companies, industrial computers record a large amount of data every day, which contain important
information such as production equipment, processes, and product performance[3]. Artificial Neural Networks (ANN) have huge advantages in dealing with big data, multiparameter, and nonlinear problems. BP neural network is a multilayer feedforward neural network with signal forward propagation and error back propagation. Most of the ANNs used are BP neural networks or their optimization improvements[4, 5]. In this paper, a HSLA steel performance prediction model of BP neural network was built, using Python language, TensorFlow and Keras artificial intelligence framework.

2. neural network modeling

2.1.Layers
The theory has proved that choosing appropriate number of hidden layer neurons, a three-layer neural network with activation functions could approximate any continuous function indefinitely with arbitrary precision. Although increasing the number of hidden layers reduces fitting errors, it also complicates the network, thereby increasing training time and tendency to overfitting. It is easier to lower the fitting error by adding the number of neurons than hidden layers. Therefore, a three-layer neural network structure of the input layer, hidden layer and output layer is preferred[6, 7].

2.2.Input Feathers
There are many process parameters affect the performance of HSLA steel from smelting to product, which consist of hundreds of parameters in chemical composition, hot rolling, cold rolling and continuous annealing. If all these parameters are imported as input values to the model, it would become complicated and consume too much calculation and time. So, it is crucial to do feather selection first.

As high-quality automobile steel, solution strengthening elements like C, Si and Mn of HSLA could be precisely controlled. Except for some products(for example IF-P) added as alloying elements, P and S are strictly controlled as harmful elements to be reduced as much as possible. Ti and Nb are the most important alloy elements commonly used for fine grain and precipitation strengthening. Compared to Nb and Ti, V with large solid solution accumulation, weak strengthening effect, and high price, is rarely used in automotive steel. Al is a strong deoxidizing element having greater influence on inclusions. N is a harmful element, and fixed by Ti forming TiN with large size from smelting, which barely has strengthening effect. However, fluctuations of N resulting in an increase or decrease of effective Ti content, will affect performance stability[8, 9]. Therefore, elements of C, Si, Mn, P, S, Al, Nb, Ti and N are selected as the chemical composition input layer feathers.

Hot rolling process has a great influence on HSLA steel. Such as heating, initial and final rolling, reduction ratio, coiling, etc. All these parameters play important roles on the second phase precipitation behavior and will be inherited by the subsequent cold rolling and annealing processes. The cold rolling
and annealing processes directly determine the final performance of the product[10]. These parameters are listed in Table 1.

| Chemical element | Hot rolling | Cold rolling | Continuous annealing |
|------------------|-------------|--------------|----------------------|
| C, Si, Mn, P, S, Al, Nb, Ti, N | heating temperature, initial rolling temperature, final rolling temperature, reduction ratio, coiling temperature, hot rolling thickness | reduction ratio, cold rolling thickness | annealing temperature, slow cooling temperature, Rapid cooling temperature, over aging temperature, skin pass elongation, belt speed |

2.3. Output Layer neurons
The output features are the results after calculations. For cold-rolled HSLA product, three main mechanical properties of yield strength(Rp0.2), tensile strength(Rm) and elongation(A80) are selected as outputs.

2.4. Hidden layer neurons
For now, there is no clear theoretical method for selecting the number of hidden layer neurons. Usually, empirical formulas are used to select, where and are the number of input and output features, and is an integer between 1 to 10. From the text above, network has 23 input and 3 output features, for we choose the ceiling limit value 10. After adding them together, the sum is rounded to the right to calculate the number 15 as the number of hidden layer neurons.

2.5. Algorithm
Gradient descent(GD) algorithm is commonly used for neural network. Most deep learning models are basically trained using GD. It achieves global convergence for the convex function optimization problem and has a unique optimal solution. However, in practice, most models are complicated with numerous parameters and multidimensional structures. In the high-dimensional space matrices formed by these parameters, the possibility of reaching the global optimum is extremely low. Most of them are local optimal solutions similar to saddle points, where the conventional GD algorithm will be converging slowly and difficult to jump out.

Mini batch stochastic gradient descent (SGD) is often used to solve global optimal problems. Lots of research facts prove that SGD is very effective for multiparameter non-convex optimization. It introduces randomness in the direction of the gradient, while mini batch calculation ensures the stability of the training process, and also takes advantage of the matrix calculation. However, SGD is affected by the hyper-parameter: learning rate. If it is small, it will slow down the convergence rate; If it is too large, it will easily cause oscillations across the optimal solution [11, 12].

In order to achieve two goals of model optimization: fast convergence speed and global optimal solution, we use adaptive moment estimation algorithm (Adam). The first-order moment calculates the expectation of past and current gradient $E(g(t))$, so that each update does not differ too much from the previous, making the transition smooth and stable. The second-order moment, which is the expectation of the square of the past and current gradients $E(g(t)2)$, improves the ability to adapt to the environment[13].

2.6. Activation function
Activation function is a function that neurons receive input values, and outputs the results after calculating inside, the purpose of which is to introduce nonlinearity. If there is no nonlinear calculation,
no matter how many layers neural network has, only a linear map will be formed, that nonlinear problem cannot be solved[14]. Commonly used activation functions are as follows:

2.6.1. Sigmoid
Sigmoid is one of the earliest activation functions applied, the advantage of which is monotonously continuous and derivable in the entire domain, and value ranges in [0, 1]. However, it has two problems that its derivative is always less than 0.25, and when approaching to the limit value (-∞, or +∞), is infinitely to 0. In the multilayer network, the gradient becomes smaller, or even disappears, resulting in a slower convergence speed.

\[
f(x) = \frac{1}{1 + e^{-x}} \quad x \in (-\infty, +\infty)
\]  

(1)

2.6.2. Tanh
Tanh is an optimization of Sigmoid, which solves the problem that output is centered around 0. The convergence speed is improved, and the actual use effect is better than Sigmoid. But for the problem that the gradient disappears is unsolved. In addition, the calculation of the derivative of Tanh is more complex than Sigmoid.

\[
f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad x \in (-\infty, +\infty)
\]  

(2)

2.6.3. Relu
The advantage of Relu is that only simple mathematical operations of addition and multiplication are needed, and convergence speed is fast. But nondifferentiable point exists in the entire domain. For \( x < 0 \), the gradient is always 0, which makes some neurons may never be activated.

\[
f(x) = \begin{cases} 
  x & x \geq 0 \\
  0 & x < 0 
\end{cases}
\]  

(3)

2.6.4. Elu
Elu combines the advantages of Sigmoid and Relu, with soft saturation on the left and no saturation on the right. The linear part allows Elu to mitigate the disappearance of the gradient, while the soft saturation makes it more robust to noise. In addition, the average output is close to 0, speeding up convergence. Thus, we select Elu as the activation function.

\[
f(x) = \begin{cases} 
  x & x \geq 0 \\
  \alpha(e^x - 1) & x < 0 
\end{cases}
\]  

(4)
3.model training and prediction

3.1.Data set
The cold-rolled HSLA product we choose is produced by an auto steel production line of Shougang group. With total number of 8 strength grades (from 260MPa to 550MPa), consisting of 5050 samples as the original data set. 5000 samples are divided into two parts: training set and validation set, and each mini set has 100 samples. The remaining 50 samples are put into test set.

3.2.Data standardization
Because different input features vary greatly and have different dimensions. For example, chemical composition value is between 0.0001(w%) and 1.0(w%), annealing temperature ranges from 700℃ to 900℃, flat elongation is in 0%~2.0%. These will cause input values with different orders of magnitude to vary greatly in the stimulation of neurons, and it is easy to cause the distortion effect of different feature contributions, therefore, it is necessary to standardize the data.

That all input feathers including chemical compositions and process parameters are positive values, and maximum and minimum are relatively stable, is considered. So, we choose dispersion normalization for data normalization, controlling all input values range between 0 and 1, and eliminate dimension at the same time.

\[ x'_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \]  

\[(5)\]

3.3.Forward propagation
A three-layer neural network with \( i \) input features, \( j \) hidden layer neurons, and \( k \) output features, that all neurons in the hidden layer are connected to all input and output neurons. The input feature vector (\([x_1, x_2, ..., x_i]\)) is represented by \( X \), and the bias is 1. For the first hidden layer neuron, its connection weight vector (\([w_{11}, w_{21}, ..., w_{ji}]\)) is represented by \( W^j \), the weight of bias is \( b_{11} \). Neurons in the output layer are connected to each neuron in the hidden layer, the weight of the first neuron connection vect or (\([v_{11}, v_{21}, ..., v_{ji}]\)) is \( V^j \), and the weight of the bias is \( b_{21} \).

\[ \alpha_i = \sum_{j=1}^{i} (x_i \bullet w_{ji}) + b_{11} = X \bullet W^j + b_{11} \]  

\[(6)\]
\[ y_1 = f(\alpha_i) \]  

\[(7)\]
\[ z_1 = f(\beta_i) \]  

\[(8)\]
\[ \beta_i = \sum_{j=1}^{i} (y_j \bullet v_{ji}) + b_{21} = Y \bullet V^j + b_{21} \]  

\[(9)\]

\( \alpha_i \) : Input of the first neuron in the hidden layer.
\( y_1 \) : Output of the first neuron in the hidden layer.
\( \beta_i \) : Input of the first neuron in the output layer.
\( z_1 \) : Output of the first neuron in the output layer.

Similarly, for \( k \) output feature value \( z_k \), the following formulas holds

\[ \cdot \alpha_j = \sum_{i=1}^{j} (x_i \bullet w_{ij}) + b_{1j} = X \bullet W^j + b_{1j} \]  

\[(10)\]
\[ y_j = f(\alpha_j) \]  

\[(11)\]
\[ \beta_k = \sum_{j=1}^{k} (y_j \bullet v_{jk}) + b_{2k} = Y \bullet V^j + b_{2k} \]  

\[(12)\]
\[ z_k = f(\beta_k) \]  
(13)

\( \alpha_j \): Input of the \( j \) neuron in the hidden layer.

\( y_j \): Output of the \( j \) neuron in the hidden layer.

\( \beta_k \): Input of the \( k \) neuron in the output layer.

\( z_k \): Output of the \( k \) neuron in the output layer.

### 3.4. Backpropagation

For regression problems of product performance prediction, we choose mean square error loss function (Equation 2.4.1). Based on gradient descent strategy, adjust the connection weight \( v_{jk} \) and \( w_{ij} \) in the negative direction of gradient. The \( t \)th correction of the hidden and output layer connection weights \( v_{jk} \), the updated weights \( v_{jk}^{t+1} \), and the update amplitude \( \Delta v_{jk} \) can be expressed by equations to \( \frac{\partial L_k}{\partial v_{jk}} \) can be calculated by the chain derivation rule.

\[
L_k = \sum_{k=1}^{k} (d_k - z_k)^2
\]  
(14)

\[
\Delta v_{jk} = -\eta \frac{\partial L_k}{\partial v_{jk}}
\]  
(15)

\[
\frac{\partial L_k}{\partial v_{jk}} = \frac{\partial L_k}{\partial z_k} \cdot \frac{\partial z_k}{\partial \beta_k} \cdot \frac{\partial \beta_k}{\partial v_{jk}}
\]  
(16)

\[
v_{jk}^{t+1} = v_{jk}^t - \eta \frac{\partial L_k}{\partial v_{jk}}
\]  
(17)

Equally, The weight of the connection between the hidden layer and the output layer \( w_{ij} \), the updated weight \( w_{ij}^{t+1} \) and the update range \( \Delta w_{ij} \) can be updated using .

\[
\Delta w_{ij} = -\eta \frac{\partial L_k}{\partial w_{ij}}
\]  
(18)

\[
\frac{\partial L_k}{\partial w_{ij}} = \frac{\partial L_k}{\partial z_i} \cdot \frac{\partial z_i}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{ij}}
\]  
(19)

\[
w_{ij}^{t+1} = w_{ij}^t - \eta \frac{\partial L_k}{\partial w_{ij}}
\]  
(20)

### 3.5. Learning rate

Learning rate(lr) is one of the most important hyper-parameters in deep learning networks. There is no theoretical guidance for the setting of this parameter currently. The generally accepted idea is that the effect of dynamically adjusting the lr to adapt to different training stages is better than the static learning rate. A high lr in the early stage to speed up the training, and a low lr in the later stage to avoid frequent cross-domain oscillations when the model is close to the optimal[22]. According to a large amount of deep learning model experiences, the initial lr is generally not to be too large (such as 0.01 or 0.001). In this paper, the initial lr is set as 0.01, which is continuously optimized according to the mean square error loss changes with the number of iterations (loss curve).

### 3.6. Regularization

The fitting ability of the neural network is strong. After many iterations, the error on the training set is even close to 0, which easily leads to over-fitting. In order to improve the generalization ability of the
model, so that it also has a predictive ability on new data set, under the situation of data scale is difficult
to increase, the regularization method is usually used. The mathematical principle of regularization is to
add a penalty term after the loss function to limit the complexity of the parameter matrix and reduce the
risk of over-fitting[15].

Deep learning models often use L1 regularization and L2 regularization. The L1 method adds a
penalty term of the parameter L1 norm to the loss function, and generates a sparse matrix with some
parameters being 0. The L2 norm does not make the parameter matrix sparse, but it can make some
parameters close to 0. Combined with the production process characteristics of HSLA steel, although
different chemical elements and process parameters have different weights on the performance of the
product, they will not be completely unaffected. For example, fluctuations in the content of acid-soluble
aluminum and total aluminum will not have a significant impact on the strength of the material, but will
affect the morphology, size and number of inclusions, and indirectly affect the toughness of the material.
Therefore, we choose L2 regularization[16].

L2 Regularization:

\[ L = L_0 + \lambda \| w \|_2 \]  

where: 
- \( L \): Loss function after regularization. 
- \( L_0 \): Original loss function. 
- \( \lambda \): Regularization coefficient. 
- \( \| w \|_2 \): L2 norm of parameters.

3.7. Cross validation

For the artificial division of the training set and the test set, the model has good accuracy on a selected
data set, but the generalization ability on the new data set cannot be guaranteed. Cross-validation is
used to evaluate the prediction performance of the model, which can reduce the risk of overfitting and
improve the generalization ability. In practical applications, K- fold cross validation (K- CV) is often
used to divide the data set into equal K parts, of which K-1 part is used as the training set and 1 part is
the test set. And K- CV is performed K times, and the average error is calculated as the performance
evaluation index[17,18]. According to the 5000 sample size of the original data set, K value is 5, and
1000 samples for each subset.

3.8. Training termination condition

With the increase of the number of training iterations of the neural network, the mean square error loss
of the predicted and output value will continue to decrease. However, the gradient also tends to be
gentle, the magnitude of each update is getting smaller and smaller, which is affected by the learning
rate, until it repeatedly oscillates across the optimal solution and fails to converge. For example, when
the fluctuation of strength prediction error is less than 1MPa, the fitting error of the model is already
lower than the actual measurement error of the material, and there is a greater risk of overfitting, so it is
necessary to set the training termination condition.

According to the mechanical properties of the material, the value 1 is selected as the necessary
condition for continued optimization, and the number of verification is 20 times, indicating that if the
validation error cannot be optimized within the next 20 iterations cycle to reduce 1, the training is
terminated. Then, the current loss and iteration times are recorded.
Fig. 3. SGD and Adam convergence curves at different learning rates

4. Result analysis

4.1. SGD VS ADAM

In figure 4, SGD and Adam algorithms have different training characteristics under the same initial learning rate (lr). When lr is 0.01, the SGD model oscillates greatly within 132 iterations, the loss amplitude reaches more than 600, and cannot converge steadily. The Adam model only experienced minor fluctuations in the late training period and ended normally. When lr is lowered to 0.005, the loss curve of the SGD model changes suddenly at 125 iterations, and the training continues after the jump drops (shown in Figure 3c), indicating that the SGD algorithm can effectively jump out of the local optimal or saddle point. But a large oscillation with an amplitude exceeding 100 still occurs during the iteration. Compared with SGD, the Adam model is stable in training, which is completed after 280 iterations. The loss of the training and verification set reach 120 or less.
When lr is reduced to 0.001, SGD model training is stable, but the speed is significantly reduced. After 1722 iterations, the loss value is about 190. In contrast, the Adam model showed efficient and smooth convergence ability, and the loss reached 89 after 742 iterations, which is only half of the SGD, showing lower fitting and prediction errors.

Fig. 4. Adam convergence curves at lower learning rates

By comparing the SGD and Adam loss curves, it can be found that within a certain range, the initial lr has a great impact on both two algorithm models. A larger lr model has a fast convergence rate, and a poorer accuracy. Decreasing lr can improve prediction accuracy, but the convergence period is extended. For the SGD, the effect of lr is the most obvious. If it is unreasonable, the model will have two extreme cases of oscillation or long convergence period. Adam introduces first-order and second-order of gradient momentum, and adaptive learning rate, which is significantly less sensitive to lr.

As we continue to reduce Adam’s initial lr to 0.0001 and 0.00001, it can be seen from Figure 4 that, as lr decreases, the model update pace becomes slower, and the change of loss curve tends to be gentle. At the same time, the number of iterations is obvious increased to nearly 7000. However, the prediction error does not decrease significantly (Figure 4a). When the lr goes to 0.00001, the error increased to more than 200 (Figure 4b). The reason is that lower lr greatly reduces the parameter update step, causing the error to drop very slowly. When 20 consecutive updates have not reached 1, the termination condition is triggered and the training stops.

4.2 Test set prediction

The optimal parameters model under the initial learning rate(lr) of 0.001 is fixed, and the prediction is performed on the 50 samples test set. The graph is drawn with three mechanical performance prediction errors (predicted value - actual value). As shown in Figure 5, the predictions in yield strength(Rp0.2), tensile strength(Rm) and elongation(A80) are basically consistent with the actual value changes. The maximum errors of yield strength and tensile strength are respectively It is 19MPa and 22MPa, and the maximum elongation error is 3.1%.

In general, the yield ratio of microalloyed high strength steel is about 0.6~ 0.8. However, according to the predicted data, the prediction error of yield strength and tensile strength is at the same level. From the detection point of view, the measurement of yield strength needs to be calculated by adjusting the elastic modulus, and the tensile strength usually has no large error. So the precision error of yield strength prediction is equivalent to the tensile strength.
5. CONCLUSION
1) For both SGD and Adam algorithms, the initial learning rate is an important parameter that affects the convergence rate. But Adam has better adaptability to learning rate, and the convergence process is stable and efficient.

2) When the initial learning rate is large, the model converges quickly, but it is easy to cause shock. Lower the initial learning rate could improve stability and reduce errors, while the convergence time will increase significantly. However, learning rate is not as low as possible. By setting a low learning rate, the accuracy of the model is limited, and even decreases probably.

3) Through the Adam algorithm and the best parameters, the mean square error on the validation set is less than 90. The absolute value of the yield strength and tensile strength prediction errors are less than 19MPa and 22MPa, and the elongation error is less than 3.1%.

REFERENCES
[1] Mao X P, Sun X J, Kang Y L, Lin Z Y, Physical metallurgy for the titanium microalloyed strip produced by thin slab casting and rolling process, Acta Metallurgica Sinica, Vol. 42, pp. 1091-1095, Oct. 2006.
[2] Xiong H H, First-principles study on the precipitation and interface behavior of carbides in steel. Shanghai University, 2018.
[3] Liu Z Y, Cao G M, Zhou X G, The predicting technologies for microstructure and properties and their core roles in smart hot rolling processes, Steel Rolling, Vol. 36, pp. 1-7, Apr. 2019.
[4] Qiu X P, (2019) Neural Networks and Deep learning. Fudan University, online publishing.
[5] Sun R Y, (2019) Optimization for deep learning: theory and algorithms, unpublished.
[6] Funahashi K, Nakamura Y. Approximation of dynamical systems by continuous time recurrent neural networks, Neural networks, Vol. 6, pp. 801-806, 1993.
[7] Hornik K, Stinchcombe M, White H, Multilayer feedforward networks are universal approximators, Neural networks, Vol. 2, pp. 359-366, 1989.
[8] Fu L M, Shan A D, Wang W, Effect of Nb solute drag and NbC precipitate pinning on the recrystallization grain growth in low carbon Nb-microalloyed steel, Acta Metallurgica Sinica, Vol. 46, pp. 832-837, Jul, 2010.
[9] Fu J, Zhu J, Di L, Tong F, Study on the precipitation behavior of TiN in the microalloyed steels, Acta Metallurgica Sinica, Vol. 36, pp. 801-804, Aug, 2000.
[10] He X L, Yang G W, Mao X P, Yu C B, Effect of Nb on the continuous cooling transformation on rule and microstructure mechanical properties of Ti-Mo bearing microalloyed steel, Acta Metallurgica Sinica, Vol. 53, pp. 648-656, Jun, 2017.
[11] Ding C C, Tao W, Tao Q, A unified momentum method with triple-parameters and its optimal convergence rate, Journal of computer research and development, Vol. 57, pp. 1571-1580, 2020.
[12] Simon O. H, (2008) Neural Networks and Learning Machines. Prentice Hall.
[13] Kingma, Diederik, Ba, Jimmy, Adam: (2015) A Method for Stochastic Optimization. In: Proceedings of the 3rd International Conference on Learning Representations, ICLR.
[14] Goodfellow I J, Bengio Y, Courville A C, (2016) Deep learning. MIT Press.
[15] Zou H, Hastie T, Regularization and variable selection via the elastic net, Journal of the Royal Statistical Society: Series B (Statistical Methodology), Vol. 67, pp. 301-320, 2005.
[16] Zou H, Hastie T, Regularization and variable selection via the elastic net, Journal of the Royal Statistical Society: Series B (Statistical Methodology), Vol. 67, pp. 301-320, 2005.
[17] Ron K, A study of cross-validation and bootstrap for accuracy estimation and model selection, Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, pp. 1137-1145, 1995.
[18] Felipe A. C. V, Raphael T. H, Valder S J, Multiple surrogates: how cross-validation errors can help us to obtain the best predictor, Structural & Multidisciplinary Optimization, Vol. 39, pp. 439-457, 2009.