Materials Transactions, Vol. 61, No. 12 (2020) pp. 2276 to 2283

Estimation of Texture-Dependent Stress-Strain Curve and *r*-Value of Aluminum Alloy Sheet Using Deep Learning[*1]

Kohta Koenuma[1], Akinori Yamanaka[2,*2], Ikumu Watanabe[3] and Toshihiko Kuwabara[2]

[1]*Graduate School of Engineering, Department of Mechanical Systems Engineering, Tokyo University of Agriculture and Technology, Tokyo 184-8588, Japan*
[2]*Division of Mechanical Systems Engineering, Institute of Engineering, Tokyo University of Agriculture and Technology, Tokyo 184-8588, Japan*
[3]*Research Center for Structural Materials, National Institute for Materials Science, Tsukuba 305-0047, Japan*

The deformation of an aluminum alloy sheet is affected by its underlying crystallographic texture and has been extensively studied using the crystal plasticity finite element method (CPFEM). Numerical material test based on the CPFEM enables the quantitative estimation of the stress-strain curve and Lankford value (*r*-value), which depend upon the texture of aluminum alloy sheets. However, the application of CPFEM-based numerical material test to the optimization of aluminum alloy texture is computationally expensive. In this paper, we propose a method for rapidly estimating the stress-strain curves and *r*-values of aluminum alloy sheets using deep learning with a neural network. We train the neural network with the synthetic crystallographic texture and stress-strain curves calculated through the numerical material tests. To capture the features of synthetic texture from a {111} pole-figure image, the neural network incorporates a convolution neural network. Using the trained neural network, we estimate the uniaxial stress-strain curve and in-plane anisotropy of the *r*-value for various textures that contain Cube and S components. The results indicate that the application of a neural network trained with the results of numerical material test is a promising method for rapidly estimating the deformation of aluminum alloy sheets. [doi:10.2320/matertrans.P-M2020853]

(Received March 12, 2020; Accepted July 20, 2020; Published November 6, 2020)

Keywords: *numerical analysis, tension test, texture, crystallite orientation, flow stress, deep learning*

1. Introduction

The basic information needed to understand and predict the deformation behavior of aluminum-alloy sheets during plastic forming is the stress-strain curve and Lankford value (*r*-value), which strongly depend upon the crystallographic texture formed during the manufacturing process.[1)] Therefore, a method to estimate the texture-dependent stress-strain curves and *r*-values of aluminum alloy sheets is required for their design and development. However, as the texture of aluminum alloy sheets involves various types of preferred crystal orientations, time-consuming material testing is required to experimentally clarify the relationship between the texture and stress-strain curve.

Numerical simulation based on the crystal plasticity finite element method (CPFEM) is effective for estimating the texture-dependent stress-strain curves of aluminum alloys.[2,3)] The results of a benchmark test held at NUMISHEET 2018[4)] demonstrated the potential of CPFEM in predicting the stress-strain curves and the in-plane anisotropy of *r*-value of the 5000-series aluminum-alloy sheets. These results indicated the potential of CPFEM-based numerical simulation as a numerical biaxial tensile test[5)] for estimating the stress-strain curves of aluminum alloys as an alternative to experimental material testing. In addition to the experimental material testing, an effective method to obtain the stress-strain curves, *r*-values, and other mechanical responses from the texture of aluminum alloy sheets would be ideal.

For several years, researchers have explored machine learning techniques efficiently to estimate the physical responses of a material. Although there are several approaches using machine learning, the neural network approach was extensively applied in the field of material engineering in the latter half of the 1990s.[6–8)] Yoshitake *et al.* proposed the use of neural networks as an effective method to estimate the fatigue crack growth rate from information on the chemical composition and crystal grain-size of nickel-base superalloy.[6)] Bhadeshia summarized the methods through which neural networks have been applied in materials science and concluded that these were very effective tools for recognizing the material features and predicting the material properties based on their features.[8)] However, it was difficult to handle large quantities of complex data because the techniques related to machine leaning were immature and the computational performance was poor.

Recently, the drastic improvements in computational performance and the development of new machine learning algorithms such as deep learning have facilitated the application of neural networks to complex classification and nonlinear regression problems involving multidimensional information such as digital images.[9–11)] In the field of materials engineering, Adachi *et al.*[12)] proposed a method for estimating the stress-strain curves using microstructural information such as the grain-size of steels and constructed an integrated software.[13)] As an example of the application of machine learning for aluminum alloys, Sheikh *et al.*[14)] developed a method to accurately estimate the flow stress depending on the strain rate and temperature, in the cold plastic forming of A5083 aluminum alloy. Yuan *et al.*[15)] demonstrated that the random forest method[16)] can predict the texture-dependent stress-strain curve and post-deformation texture of the material using the stress-strain curve of copper calculated through the viscoplastic self-consistent (VPSC) method as training data. Onoshima *et al.*[17)] showed that the stress-strain curve of A1145 aluminum alloy can be

[*1]This Paper was Originally Published in Japanese in J. JSTP **61** (2020) 48–55.
[*2]Corresponding author, E-mail: a-yamana@cc.tuat.ac.jp

accurately reproduced through deep learning to estimate the parameters of crystal plasticity constitutive equations used for numerical material test. However, to the best of the authors' knowledge, the application of deep learning with a neural network to estimate the stress-strain curves and r-values directly from the texture of aluminum alloy sheets has not been proposed.

In this study, we propose a method to rapidly estimate the stress-strain curves and the in-plane anisotropy of r-value from the texture of aluminum alloy sheets using deep learning with a neural network. The proposed method is equivalent to the use of deep learning for regression analysis of the nonlinear relationship between the texture of aluminum alloy sheets and their stress-strain curves or r-values. To perform this nonlinear regression, vast quantity of training data are required for training the neural network. The training data in this study includes the textures of aluminum alloy sheets along with their corresponding stress-strain curves and r-values. It is difficult to acquire this training data within a short period through experimental material testing. Therefore, we generate numerous stress-strain curves and r-values calculated in advance by numerical material test[3] for application as training data for the neural network.

Moreover, we describe the training of the neural network to estimate the stress-strain curves and the in-plane anisotropy of r-value using various types of texture. We verify the proposed method by demonstrating that our estimation results agree well with the numerical material test results which are assumed as the true values.

2. Preparation of Training Data through Numerical Material Test

To prepare the training data for deep learning, we conducted several numerical material tests based on the CPFEM. These tests require the initial crystal orientation as the input data. In this section, we describe the applied methodology for the numerical material tests and the generation of the initial crystal orientation data including the preferred texture orientation of the aluminum alloy sheets.

2.1 Numerical material test using the crystal plasticity finite element method

The numerical material tests used for generating training data were based on the CPFEM with the homogenization method, which has been previously described.[3] For details on the numerical material tests, see Ref. 3). In this section, we describe the work hardening model used in the numerical material tests.

The resolved shear stress $\tau^{(\alpha)}$ in slip system α is given by the following equation:

$$\tau^{(\alpha)} = P_{ij}^{(\alpha)}\sigma_{ij}, \tag{1}$$

where σ_{ij} is the Cauchy stress tensor and $P_{ij}^{(\alpha)}$ is the Schmid tensor for slip system α ($\alpha = 1, 2, \ldots, 12$). For the shear-strain rate in slip system α, we use the following exponential model proposed by Peirce *et al.*:[18,19]

$$\dot{\gamma}^{(\alpha)} = \dot{\gamma}_0 \frac{\tau^{(\alpha)}}{g^{(\alpha)}}\left|\frac{\tau^{(\alpha)}}{g^{(\alpha)}}\right|^{\frac{1}{m}-1}, \tag{2}$$

where $\dot{\gamma}_0$ and m are the reference shear strain rate and strain rate sensitivity index, respectively; $g^{(\alpha)}$ is the critical resolved shear stress in slip system α, and its time evolution (i.e., work hardening) is expressed by the following equations:

$$g^{(\alpha)} = \tau_0 + \int_t \dot{g}^{(\alpha)} dt, \tag{3}$$

$$\dot{g}^{(\alpha)} = \sum_{\beta} h^{(\alpha\beta)}|\dot{\gamma}^{(\beta)}|, \tag{4}$$

where τ_0 is the initial critical resolved shear stress; $h^{(\alpha\beta)}$ is the hardening coefficient matrix which represents the contribution of slip system α to the hardening of slip system β, and is expressed by the following equation:[20]

$$h^{(\alpha\beta)} = qh(\gamma) + (1-q)h(\gamma)\delta_{\alpha\beta}, \tag{5}$$

where δ_{ij} is Kronecker's delta, and q is a coefficient that represents the level of latent hardening. The value of $h(\gamma)$ is obtained from the following equation related to the accumulated shear strain:

$$h(\gamma) = h_0 nC\{C(\gamma_{\mathrm{int}} + \gamma)\}^{n-1}, \tag{6}$$

where h_0, n, C, and γ_{int} are the initial work hardening coefficient, hardening index, hardening coefficient, and initial shear strain, respectively, and γ is the accumulated shear strain.

To perform numerical material test, it is necessary to determine the parameters in eqs. (2)–(6). Although this study does not target specific materials, we use the parameters determined based on the true stress-strain curves obtained through the uniaxial tensile testing of A5182-O aluminum alloy sheet which is the test material in Ref. 3). The sheet includes Cube-texture components; the other physical property values and parameter values are as indicated in Ref. 3).

2.2 Generation of synthetic crystallographic textures

In order to generate training data for the neural network used for deep learning, we created synthetic crystallographic textures that include multiple preferred crystal orientations and applied them as the basis for generating the initial crystal orientation data used in the numerical material tests. The synthetic textures, which are a type of "pseudo texture" generated by numerical calculation, were created as described below.

The crystallographic texture in aluminum-alloys includes several preferred orientations such as the Cube, Goss, S, and copper components.[21] Although it would be desirable to create a synthetic texture that contains the various preferred orientations, instead of creating a synthetic texture that includes all the preferred orientations, we created a texture with only two preferred orientations: the Cube texture, which is a typical recrystallization texture of aluminum alloys, and the S texture, which is a rolling texture, as well as random textures. The purpose of this study is to use deep learning to rapidly estimate the stress-strain curves and r-values of aluminum alloy sheets from the texture information, and demonstrate that the estimated stress-strain curves agree with the numerical material test results. Hence, we validate the proposed method for a simple synthetic texture containing the Cube and S textures.

For generating the synthetic texture using the above-mentioned procedure, it is assumed that the crystal-orientation distribution follows the 3D Gaussian probability density function f shown below, where the average value is considered to be the ideal Cube and S orientations in Bunge's Euler angular space, i.e., $(\phi_1, \phi, \phi_2) = (0, 0, 0)$ and $(59°, 37°, 63°)$.

$$f(\phi_1, \phi, \phi_2) = \frac{1}{(\sqrt{2\pi})^3 \xi_i^3} \exp\left(-\frac{\phi_1^2 + \phi^2 + \phi_2^2}{2\xi_i^2}\right), \quad (7)$$

where ξ_i^2 ($i =$ Cube or S) is the variance that indicates the variation in crystal orientation with respect to the ideal Cube or S texture component.

We generated various synthetic textures by varying the volume fractions of the preferred orientations included in the synthetic texture and the variance given by eq. (7). In this study, the volume fractions of the Cube and S textures (denoted by V_{Cube} and V_S, respectively) were varied in increments of 10% in the range of 10 and 90%. When the sum of V_{Cube} and V_S was less than 100%, the remaining orientations were randomly created using uniform random numbers. Based on Wu *et al.*,[22] we changed the variances of the Cube and S textures (denoted by ξ_{Cube}^2 and ξ_{S}^2, respectively) every $3\,\mathrm{deg}^2$ in the range of 2–$11\,\mathrm{deg}^2$.

The initial crystal orientation data for the numerical material tests were generated using eq. (7) according to the following steps. Henceforth, we refer to this sequence of steps as "sampling."

Step 1 Generate three real numbers in the range of 0–1 using uniform random numbers, and then convert them to real numbers (a, b, c) that follow a normal distribution with a mean $= 0$ and variance $= 1$ through the Box-Muller method.[23)]

Step 2 Multiply the real numbers obtained in Step 1 by the standard deviation ξ_i and 360° to obtain the random crystal orientation $(\phi_1{}', \phi', \phi_2{}')$. The results are in degrees (°) (unit).

Step 3 Compute the initial crystal orientation data (ϕ_1, ϕ, ϕ_2) by adding the Euler angle of the ideal orientation to the crystal orientation $(\phi_1{}', \phi', \phi_2{}')$ obtained in Step 2. For example, for the S texture, we obtain a crystal orientation of $(\phi_1, \phi, \phi_2) = (\phi_1{}' + 59°, \phi' + 37°, \phi_2{}' + 63°)$.

Step 4 Generate the initial crystal orientation data that obeys the probability density function f expressed by eq. (7), by repeating Steps 1–3 the same times as the desired number of initial crystal orientations.

The initial crystal orientation data will differ slightly depending on the random numbers used in Step 1 when using the above procedure to generate the initial crystal orientation data, even if we do not change the values of volume fraction and variance used to generate the initial crystal orientation data. Specifically, the greater is the variance ξ_i^2 in eq. (7), the greater is the difference in the initial crystal orientations generated by sampling, and the greater is the variation in the results of the numerical material tests. Therefore, in this study, the number of sampling is given by eq. (8), for a probability density function f representing a given synthetic texture,

$$N_{sample} = \frac{V_{\mathrm{Cube}} \xi_{Cube}^2 + V_{\mathrm{S}} \xi_S^2}{2}. \quad (8)$$

For example, when a synthetic texture is $V_{\mathrm{Cube}} = 0.5$, $\xi_{\mathrm{Cube}}^2 = 8\,\mathrm{deg}^2$, $V_S = 0.1$, and $\xi_{\mathrm{S}}^2 = 5\,\mathrm{deg}^2$, the sampling number is 4 according to eq. (8). Therefore, for this synthetic texture, the numerical material test was performed four times, and all the results were included in the training data. When this procedure was applied to all the volume fractions and variances, a total of 1,468 initial crystal orientations were generated.

In order to obtain the in-plane anisotropy of r-values for all the initial crystal orientations, three types of numerical material tests were performed at tensile directions of 0°, 45°, and 90°, relative to the rolling direction (RD) of the sheet. Therefore, a total of $1{,}468 \times 3 = 4{,}404$ numerical material tests were performed, and the results of all the calculations were used as training data.

3. Deep Learning with Neural Network

This section describes the neural network used in this study as well as the deep learning method using the training data generated by the method described in Section 2.

3.1 Generation of {111} pole-figure image of synthetic texture

In this study, we entered the texture information into the neural network, and obtained the estimates of the stress-strain curves and in-plane anisotropy of the r-values as the output. The format for entering the texture information into the neural network is an important factor that determines the accuracy of the stress-strain curve and r-value estimates. The method adopted in this study involved entering a {111} pole-figure image into the neural network. Recent neural network-based deep learning techniques for the image recognition exhibit extremely powerful discrimination capabilities when images are used as the input.[24)] This suggests that the usage of image data representing textural features as input to the neural network is a suitable method for estimating the stress-strain curves with high accuracy.

The {111} pole-figure images of the synthetic textures, which are the input to the neural network, were generated as follows. First, for the initial crystal orientation data generated using the method described in Section 2.2, the positions of the poles on the {111} pole figures were determined. Further, to form the pole-figure image, the pole figure was divided into $N \times N$ subregions, where each subregion corresponds to one image pixel. The luminance value of each pixel was then set according to the number of poles included in the pixel. In this study, each pixel luminance value L was set to $255n/a$, where n is the number of poles contained in the pixel and a is a constant set to 10. In approximately 0.01% of the synthetic texture images used as training data in this study, there were pixels whose luminance values exceeded 255. In these cases, the pixel luminance value was corrected to 255. Using the above-mentioned method, we represented the degree of orientation of the texture in terms of the individual pixel luminance, as depicted in Fig. 1(a).

The {111} pole-figure images were formatted to PNG for the ease of compression. In order to reduce the memory required for deep learning, the image resolution was always set to $2^{\mathrm{n}} \times 2^{\mathrm{n}}$ pixels, i.e., a power of two on each axis.

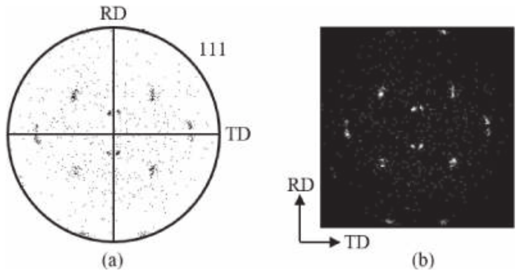

Fig. 1 (a) Example of {111} pole figure of a synthetic crystallographic texture and (b) the corresponding image used for the input data to the neural network.

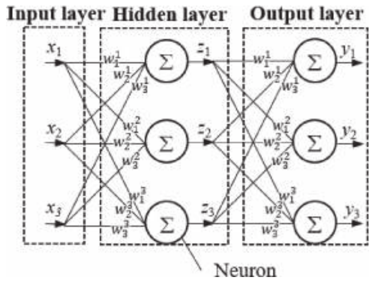

Fig. 2 Schematic diagram of a neural network.

Before the {111} pole-figure image was entered into the neural network, it was converted into a black-and-white inverted image, as shown in Fig. 1(b). This step was performed to render the image data similar to the format of the MNIST dataset,[25] which includes a set of sample images often used in deep-learning image recognition; it is noted that even if the data were not converted into black-and-white inverted images, the results of training the neural network would not be affected.

3.2 Neural network applied for deep learning

Neural networks, which are mathematical models used to implement deep learning, are inspired by the biological mechanism of the system of neurons in the brain.[26] Figure 2 displays the schematic of a simple neural network; the circles in the figure indicate the modeled neurons. A neural network comprises an input layer that receives input data, hidden layers that extract the features of the data received by the input layer, and an output layer that generates output data.

A weight is assigned to each input value x_i for each neuron, and the weights are then are summed as shown in the equation below. This equation depicts the output of a neuron in the hidden layer shown in Fig. 2.

$$z_j = \sum_{i=1}^{3} w_i^{(j)} x_i, \tag{9}$$

where $w_i^{(j)}$ is the weight assigned to the input value x_i to the j^{th} neuron. In neural network training, the weights $w_i^{(j)}$ are determined such that the relationships between the training data inputs and outputs are reproduced. To determine the appropriate weight $w_i^{(j)}$ for multiple inputs, optimization algorithms such as the gradient descent are used to reduce the difference between the true output values contained in the training data and the neural-network estimated outputs.[27,28] In this study, Adam[29] was used as the optimization algorithm. A layer in which all the neurons are connected to the preceding and following layers, as shown in Fig. 2, is called a fully connected layer. Various types of neural-network connection methods have been proposed, in addition to the fully connected layers used in deep learning.[14,30,31]

In this study, we supplemented the fully connected layers with convolutional layers, pooling layers, and dropout layers.[30] A neural network that uses convolutional layers is called a convolutional neural network (CNN). CNN combines convolutional and pooling layers in succession to compress the input image and capture the image features.[30] The dropout layers assist in optimizing the weights and prevent overlearning by ignoring certain neurons during neural-network training.[31]

In this study, the above-described layers were used to construct the neural network shown in Fig. 3. The role of each layer in our network is as follows:

Layer 1 is a convolutional layer that receives a $2^n \times 2^n$ pixel {111} pole-figure image and converts it into 16 images of 25×25 pixels that capture the features of the input pole-figure image.

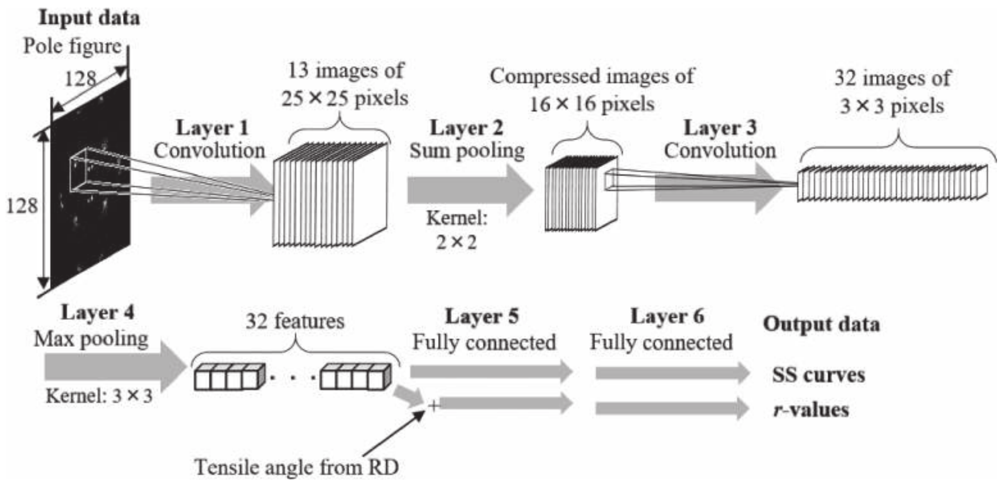

Fig. 3 Neural network constructed in this study. This neural network is composed of two pairs of convolutional and pooling layers and two fully connected layers.

Layer 2 is a sum pooling layer that calculates the sum of the 2×2-pixel regions (kernels) in each of the 16 images generated by Layer 1 in order to compress them into 16×16-pixel images representing the luminance values. Layer 3 is a convolutional layer that converts the image compressed by Layer 2 into 32 images of 3×3 pixels to capture the features of the pole-figure image. Layer 4 converts the {111} pole-figure input into a 32-dimensional feature value by calculating the maximum value of each 3×3-pixel region (kernel) in each of the 32 images generated by the convolution layer in Layer 3. Layer 5 improves the regression accuracy for nonlinear training data using the max pooling layer and a fully connected layer. Layer 6 is a fully connected layer for outputting the stress-strain curve or r-value corresponding to the feature value of the {111} pole-figure in the input. The stress and strain are normalized to values between zero (minimum) and unity (maximum). The normalized stress values and their corresponding strain values are then output along with the maximum and minimum stress and strain values used for normalization.

To estimate the in-plane anisotropy of r-value, the value indicating the angle of the tensile direction relative to the RD is added to the 32-dimensional feature value generated by the max pooling layer in Layer 4. This enables the estimation of the r-value for any tensile direction.

Individual neural networks are trained separately to estimate the stress-strain curves and the in-plane anisotropy of r-value from the texture images.

The greater the resolution of the {111} pole figures, the longer it takes to train and apply the neural network. In this study, the estimation accuracy was verified by training neural networks under three conditions, namely $n = 6$, 7, and 8. It followed that no further improvement of the estimation accuracy was observed when the neura network was trained at resolutions more than $n = 7$. Therefore, we set the resolution of the {111} pole-figure images to $n = 7$, i.e., 128×128 pixels.

3.3 Validation of the trained neural network

We used the trained neural network to estimate the stress-strain curves and the in-plane anisotropy of r-values from the texture information, and then verified its accuracy by comparing the results with the numerical material test results.

As explained in Section 2.2, the initial crystal orientation data generated during sampling can vary depending on the random numbers, even if we do not change the values of volume fraction and variance used to generate the synthetic texture. Therefore, when performing numerical material tests to generate verification data, we used five initial crystal orientation data obtained by sampling five times, using the probability density function f representing a given synthetic texture. When estimating using the trained neural network, we used the same probability density function f to generate the fifty {111} pole-figure images used as inputs for estimating the stress-strain curves.

4. Estimation Results Using Neural Network

4.1 Estimation of the stress-strain curves

Figure 4 depicts the {111} pole figures representing the three types of initial crystal orientation generated using V_{Cube} set to 10%, 50%, and 90%, respectively. Here, ξ^2_{Cube}, V_S, and ξ^2_{S} were set to $5\,\mathrm{deg}^2$, 10%, and $5\,\mathrm{deg}^2$, respectively.

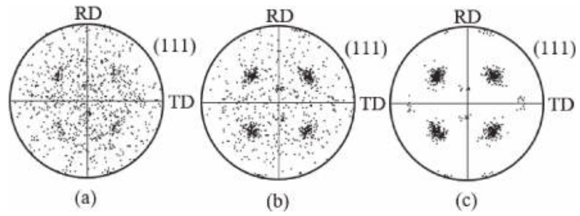

Fig. 4 {111} pole figure of the synthetic crystallographic texture used as the input data for the numerical material test and the trained neural network. The volume fraction of the Cube texture, V_{Cube}, for each texture is (a) 10%, (b) 50%, and (c) 90%, respectively. The volume fraction of the S texture, and variances of the Cube and S textures are fixed at $V_{\mathrm{S}} = 10\%$, $\xi^2_{\mathrm{S}} = 5\,\mathrm{deg}^2$, and $\xi^2_{\mathrm{Cube}} = 5\,\mathrm{deg}^2$, respectively.

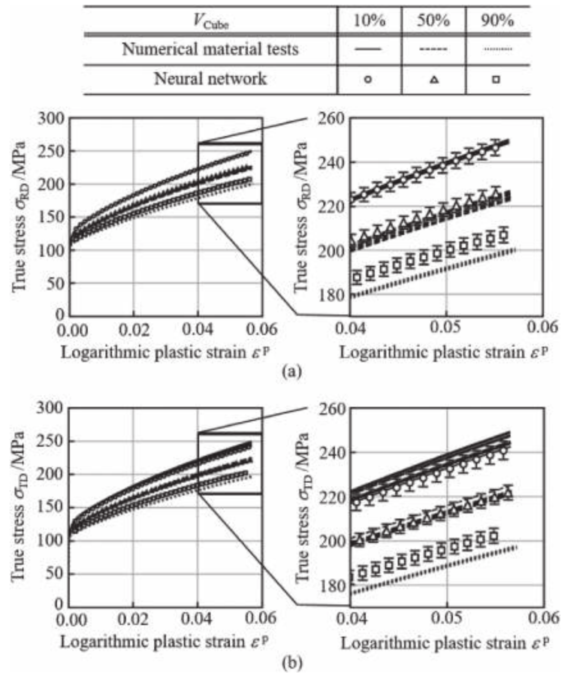

Fig. 5 True stress-logarithmic plastic strain curves for the tensile direction of (a) RD and (b) TD calculated by numerical tensile tests and estimated by the trained neural network using the different textures shown in Fig. 4.

Figure 5(a) displays the true stress-logarithmic plastic strain curves calculated by numerical material tests and estimated by the trained neural network in the tensile direction of RD using the initial crystal orientation shown in Fig. 4; the error bars in the figure indicate the variance of the estimated value depending on the {111} pole figure-images input to the neural network. The flow stress calculated by numerical material test decreases as V_{Cube} increases. In the numerical material tests, even the volume fraction and variance in the Cube texture were identical to those in the S texture, the flow stress varied due to minor differences in the initial crystal orientation. Similar to the numerical material test results, in the stress-strain curves estimated by the trained neural network, the flow stress decreased with the increase in V_{Cube}. At V_{Cube} values of 10% and 50%, the values estimated by the neural network agreed well with those calculated by numerical material tests. On the other hand, when $V_{Cube} = 90\%$, the neural network overestimates the flow stress compared to that calculated by the numerical material test.

Figure 5(b) depicts the estimated results of the true stress-logarithmic plastic strain curves, when the transverse direction (TD) is the tensile direction. Using the trained neural network, it was possible to estimate the decrease in flow stress associated with the increase in V_{Cube}, similar to the results in Fig. 5(a). However, the estimation accuracy when $V_{Cube} = 10\%$ was less than that shown in Fig. 5(a).

Figure 6 shows the {111} pole figures representing the three types of initial crystal orientation generated when the variance of the Cube texture (ξ^2_{Cube}) for each texture is 2 deg^2, 5 deg^2, and 8 deg^2, respectively. Here, V_{Cube}, V_S, and ξ^2_S were set to 90%, 10% and 5 deg^2, respectively.

Figure 7(a) displays the true stress-logarithmic plastic strain curves calculated by numerical material tests and estimated using the trained neural network in the tensile direction of RD with the initial crystal orientation shown in Fig. 6. The flow stress calculated by numerical material test decreases as ξ^2_{Cube} increases. On the other hand, the stress-strain curves estimated by the trained neural network exhibits an opposite trend. However, when ξ^2_{Cube} is 2 deg^2 and 5 deg^2, the results agree well with the numerical material test results. When ξ^2_{Cube} is 8 deg^2, the stress-strain curves estimated by the neural network are greater than those obtained from the numerical material tests.

Figure 7(b) shows the estimated results of the true stress-logarithmic plastic strain curves when the tensile direction is TD. The flow stress calculated by numerical material test decreases as ξ^2_{Cube} increases, similar to the results shown in Fig. 7(a). When ξ^2_{Cube} is 8 deg^2, the estimates by the neural network are greater than the numerical material test results.

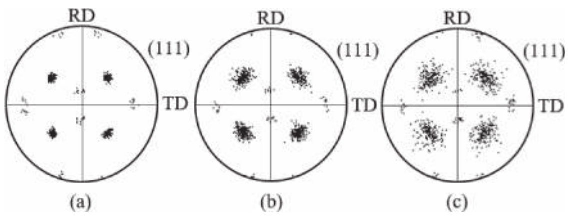

Fig. 6 {111} pole figure of the synthetic crystallographic texture used as the input data for the numerical material test and the trained neural network. The variance of the Cube texture ξ^2_{Cube} for each texture is (a) 2 deg^2, (b) 5 deg^2, and (c) 8 deg^2. The volume fraction of the S texture and variances of the Cube and S textures are fixed at $V_{Cube} = 90\%$, $V_S = 10\%$, and $\xi^2_S = 5$ deg^2, respectively.

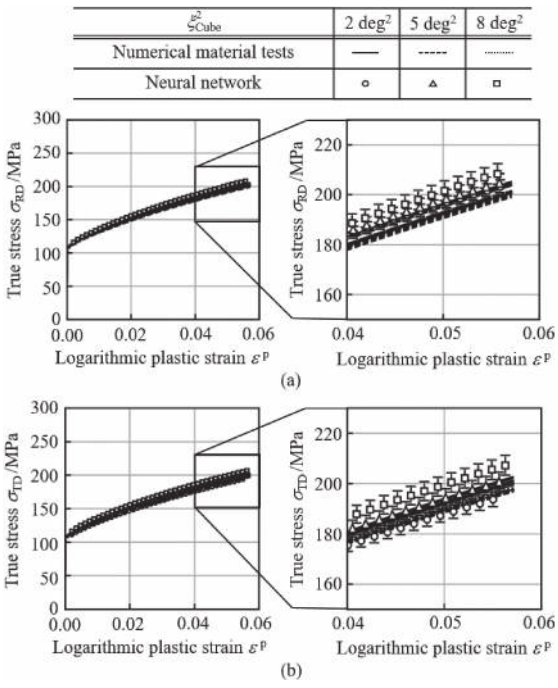

Fig. 7 True stress-logarithmic plastic strain curves for the tensile direction of (a) RD and (b) TD calculated by numerical tensile tests and estimated by the trained neural network using the different textures shown in Fig. 6.

4.2 Estimation of in-plane *r*-value

This section presents the results of using the trained neural network to estimate the in-plane anisotropy of the r-value from the texture information. Table 1 lists the volume fraction and variance of the Cube and S texture components of the synthetic textures used for estimating the r-value.

Figure 8 depicts the changes in the r-value associated with the increase in the logarithmic plastic strain calculated by numerical material test and estimated by the trained neural network. The values of r_{90} and r_{45} obtained by numerical material test decrease with the increase in the logarithmic plastic strain. On the other hand, r_0 increases with the increase in the logarithmic plastic strain. For all the synthetic textures, when the logarithmic plastic strain is 0.05, the greatest value is at r_0, followed by r_{90} and r_{45}. Comparing the results of Cases 3, 4, and 5, as V_{Cube} increases, r_{45} gradually approaches zero. Moreover, the higher the volume fraction of the S orientation relative to the volume fraction of the Cube orientation, the greater is the difference between r_0 and r_{90}.

The above figures indicate that the estimates by the trained neural network can grasp the tendency of results of the numerical material tests. Furthermore, as the results of the numerical materials tests are within the range of the r-value variation estimated by the trained neural network, it is established that the proposed method can be applied for estimating the anisotropy of r-values with the same accuracy as that of the numerical material tests.

Table 1 Volume fraction and variance of Cube and S textures used for generating synthetic crystallographic textures.

Case	V_{Cube}	ξ^2_{Cube}	V_S	ξ^2_S
1	14%	2 deg^2	32%	2 deg^2
2	86%	3 deg^2	12%	10 deg^2
3	20%	10 deg^2	64%	4 deg^2
4	21%	5 deg^2	76%	4 deg^2
5	70%	9 deg^2	25%	9 deg^2
6	39%	8 deg^2	56%	10 deg^2

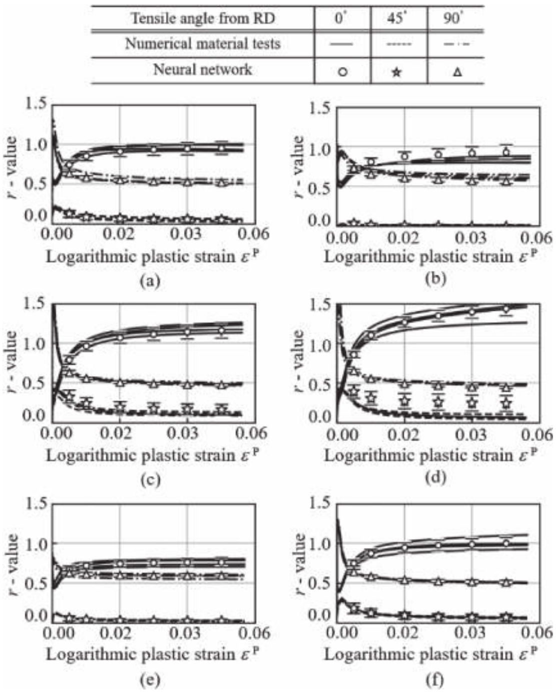

Fig. 8 Variation of r-value with logarithmic plastic strain calculated by the numerical material tests and estimated by the trained neural network for cases 1–6 shown in Table 1. The error bars show the error included in the estimation by the trained neural network.

5. Discussion

5.1 Improvement of the stress-strain curve estimation accuracy

We used the trained neural network to estimate the stress-strain curves based on the volume fraction and variance of the Cube texture. The results demonstrated that the stress-strain curves calculated through numerical material tests were accurately estimated. As mentioned in Section 1, performing numerous numerical material tests and accumulating training data is time consuming. However, if the neural network training data can be accumulated faster, for example, by using parallel computation to increase the speed of numerical material test, it should be possible to estimate the texture-dependent stress-strain curve within a short period. On the other hand, the estimation accuracy of the stress-strain curves is clearly reduced with increasing V_{Cube} and ξ^2_{Cube}. Hereafter, we discuss methods to improve this decrease in estimation accuracy.

In this study, we expressed textural differences by increasing the pixel luminance value with a high degree of integration in the {111} pole-figure image. On the other hand, because the luminance value in each pixel is expressed as a value between 0–255, the variance is relatively small, and differences in the texture cannot be clearly recognized. This may be the cause for the decrease in the estimation accuracy of the neural network when the Cube texture component is more than 80%. Therefore, the estimation accuracy can be improved by modifying the method described in section 3.1 for calculating the pixel luminance value in the {111} pole-figure image, using a pole figure other than the {111} pole figure as the input data to the neural network, or a texture representation other than pole figures. In addition, when the variance of the texture is increased, the estimation accuracy may be improved by generating more training data.

5.2 Improvement of estimation accuracy of in-plane anisotropy of *r*-value

We used a trained neural network to estimate the in-plane anisotropy of r-value for six types of synthetic texture. The results showed that for most textures, it was possible to estimate the r-values calculated by numerical material test. However, there were errors in the r-value estimated by the neural network. The causes for these errors and the solutions are discussed below.

Variations in the estimates of the r-values by the neural network were caused because only few r-values were calculated by numerical material tests, rendering these values strongly dependent on minor differences in the initial crystal orientation. As noted in Section 4.2, even if we do not change the values of volume fraction and variance of the ideal orientations constituting the synthetic texture, the initial crystal orientation used as input information for the numerical material tests will differ slightly depending on the random numbers. Therefore, the numerical material test results will vary depending on the difference. To solve this problem, it is necessary to increase the initial number of crystal orientations, in the numerical material test.

As mentioned above, issues remain to be solved in future research. From the results of this study, however, it is demonstrated that the use of {111} pole figures as input data to a CNN-based neural network for capturing the textural features of aluminum alloy sheets is highly promising, and that the stress-strain curves and the in-plane anisotropy of r-value can be estimated with high accuracy. Although regression estimation by deep learning with a neural network is essentially an interpolated estimate of the training data, it is advantageous from an engineering perspective, for example, in reducing the number of material tests that would otherwise be required.

6. Conclusion

In this study, we proposed the new method using deep learning with the neural network to rapidly estimate the texture-dependent stress-strain curves and r-values of aluminum alloy sheets. The neural-network training data were obtained through numerical material tests based on CPFEM, with synthetically generated texture data as the input information. We stored the results of the numerical material tests and used them as input for training the neural network, which could estimate with high accuracy the stress-strain curves and the in-plane anisotropy of r-value from the {111} pole figures of the textures.

The challenge for future research is to improve the estimation accuracy of the neural network by improving the training data, for example, by supplementing the training data with actual experimental data or numerical material test results based on a greater number of initial crystal orientations than those applied in this study.

Acknowledgments

This research was performed with the support of the Grant-in-Aid for Scientific Research (B) (17H03425), which we gratefully acknowledge here. We also thank Ryunosuke Kamijo for the valuable discussions.

REFERENCES

1) G. Wasserman and Y. Masuda: J. JILM **17** (1967) 3–11.
2) S. Li, P.V. Houtte and S.R. Kalidindi: Model. Simul. Mater. Sci. Eng. **12** (2004) 845.
3) K. Hashimoto, A. Yamanaka, J. Kawaguchi, K. Sakura and T. Kuwabara: J. JILM **65** (2015) 196–203.
4) Benchmark Problems and Results: The 11th Int. Conf. on Numerical Simulation of 3D Sheet Metal Forming Processes, ed. by T. Hama, M. Kuroda, T. Kuwabara, S. Takahashi and A. Yamanaka, (2018) pp. 129–170.
5) I. Watanabe and K. Terada: Int. J. Mech. Sci. **52** (2010) 343–355.
6) S. Yoshitake, V. Narayan, H. Harada, H.K.D.H. Bhadeshia and D.J.C. Mackay: ISIJ Int. **38** (1998) 495–502.
7) H. Fujii, D.J.C. Mackay and H.K.D.H. Bhadeshia: ISIJ Int. **36** (1996) 1373–1382.
8) H.K.D.H. Bhadeshia: ISIJ Int. **39** (1999) 966–979.
9) A. Ziletti, D. Kumar, M. Scheffler and L.M. Ghiringhelli: Nat. Commun. **9** (2018) 2775.
10) S.M. Azimi, D. Britz, M.M.F. Engstler and F. Mücklich: Sci. Rep. **8** (2018) 2128.
11) N. Haghdadi, A. Zarei-Hanzaki, A.R. Khalesian and H.R. Abedi: Mater. Des. **49** (2013) 386–391.
12) Y. Adachi, K. Shinkawa, A. Okuno, S. Hirokawa, S. Taguchi and S. Sadamatsu: Tetsu-to-Hagané **102** (2016) 47–55.
13) Y. Adachi, Y. Matsushita, I. Murakami and I.J. Inoue: Shisutemu/Seigyo/Joho **61** (2017) 188–193.
14) H. Sheikh and S. Serajzadeh: J. Mater. Process. Technol. **196** (2008) 115–119.
15) M. Yuan, S. Paradiso, B. Meredig and S.R. Niezgoda: Integr. Mater. Manuf. Innov. **7** (2018) 214–230.
16) L. Breiman: Mach. Learn. **45** (2001) 5–32.
17) S. Onoshima and T. Oya: Proc. Manuf. **15** (2018) 1833–1840.
18) D. Peirce, R.J. Asaro and A. Needleman: Acta Metall. **30** (1982) 1087–1119.
19) J. Pan and J.R. Rice: Int. J. Solids Struct. **19** (1983) 973–987.
20) J.W. Hutchinson: Proc. R. Soc. London, Ser. A **319** (1970) 247–272.
21) A. Rollett, F.J. Humphreys, G.S. Rohrer and M. Hatherlyt: *Recrystallization and Related Annealing Phenomena, Second Edition*, (Elsevier, Amsterdam, 2004).
22) P.D. Wu, S.R. MacEwen, D.J. Lloyd and K.W. Neale: Mater. Sci. Eng. Ser. A. **364** (2004) 182–187.
23) G.E.P. Box and M.E. Muller: Ann. Math. Stat. **29** (1958) 610–611.
24) C.C. Aggarwal: *Neural Networks and Deep Learning: A Textbook*, (Springer, Berlin, 2018).
25) Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Proc. IEEE **86** (1998) 2278–2324.
26) J. Duchi, E. Hazan and Y. Singer: J. Mach. Learn. Res. **12** (2011) 2121–2159.
27) M.D. Zeiler: arXiv:1212.5701 (2012).
28) D.P. Kingma and J.L. Ba: arXiv:1412.6980 (2014).
29) N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov: J. Mach. Learn. Res. **15** (2014) 1929–1958.
30) S. Iofee and C. Szegedy: arXiv:1502.03167 (2015).
31) T. Okatani: *Shinsou Gakushu*, (Kodansha Ltd., Tokyo, 2015).