Estimating Input Coefficients for Regional Input–Output Tables Using Deep Learning with Mixup

Shogo Fukui

*Faculty of Economics, Yamaguchi University, Japan.

Corresponding author(s). E-mail(s): sfukui@yamaguchi-u.ac.jp;

This version of the article has been accepted for publication, after peer review but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: https://doi.org/10.1007/s10614-024-10641-1. Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms.

Abstract

Input–output tables provide important data for the analysis of economic states. Most regional input–output tables in Japan are not publicly available; therefore, they have to be estimated. Input coefficients are pivotal in constructing precise input–output tables; thus, accurately estimating these input coefficients is crucial. Non-survey methods have previously been used to estimate input coefficients of regions as they require fewer observations and computational resources. However, these methods discard information and require additional data.

The aim of this study is to develop a method for estimating input coefficients using artificial neural networks with improved accuracy compared to conventional non-survey methods. To prevent overfitting owing to limited data availability, we introduced a data augmentation technique known as mixup. In this study, the vector sum of data from multiple regions was interpreted as the composition of the regions and the scalar product of regional data was interpreted as the scaling of the region. Based on these interpretations, the data were augmented by generating a virtual region from multiple regions using mixup.

By comparing the estimates with the published values of the input coefficients for the whole of Japan, we found that our method was more accurate and stable

than certain representative non-survey methods. The estimated input coefficients for three Japanese cities were considerably close to the published values for each city.
This method is expected to enhance the precision of regional input–output table estimations and various quantitative regional analyses.

Keywords: regional input–output table, deep learning, non-survey method, data augmentation, mixup

1 Introduction

Input–output tables record the flow of products and services by the industry for a given region and period. These tables are crucial in various quantitative analyses, such as economic spillover effects and general equilibrium analyses. In general, input–output tables are not available for all regions and periods because the compilation thereof requires vast amounts of primary data and significant effort. The only input–output table for Japan that is strictly derived from primary data is that of the entire country; it is not available region-wise. Although estimated input–output tables are available for prefectures and certain cities, numerous smaller administrative units (i.e., cities, towns, and villages) do not publish their input–output tables. Therefore, estimation of input–output tables becomes necessary to analyze a small region using these tables.

The estimation methods for input–output tables can be classified as survey, non-survey, or hybrid methods. Non-survey and hybrid methods are commonly used in small regions. Richardson (1985) summarized specific methods that are commonly used to estimate regional input–output tables. The survey method derives an input–output table by synthesizing primary data that are obtained from firm and consumer surveys related to the economy of the target region. This method is highly accurate because it uses information from individual firms and consumers, but it requires vast amounts of primary data for estimation. Consequently, survey methods are often used for entire countries but rarely for small areas such as cities. In comparison, non-survey methods are generally simpler and more cost-effective for estimating input–output tables as they require fewer data inputs. Hybrid methods estimate the table by combining another survey or dataset with the results of a non-survey estimation.

Accurate estimation of the input coefficients is pivotal in constructing precise input–output tables. The input coefficient refers to the transfer of output between industries (intermediate input) divided by the gross output of each industry. These coefficients are summarized in an input coefficient matrix. By estimating the input coefficient matrix, it is possible to derive the intermediate inputs, which are calculated as the product of the input coefficients and gross output. This contributes significantly to the estimation of the complete input–output table. Even if the full input–output table cannot be estimated, economic spillovers are calculated by estimating the input coefficient matrix.

Numerous studies have presented non-survey methods for estimating the input coefficients of regional input–output tables. The location quotient (LQ) and RAS

methods are representative non-survey methods for estimating the input coefficients for a region.

According to Isserman (1977), the LQ is defined as "the ratio of an industry's share of the economic activity of the economy being studied to that industry's share of another economy." The LQ method uses LQs to estimate the input coefficients of the target region from the coefficients of the reference region. The estimation of input coefficients using LQs includes derivatives such as cross-industry LQ (CILQ), Round's semilogarithmic LQ (RLQ), and Flegg's LQ (FLQ) (Flegg & Tohmo, 2013, 2016). An early study that considered estimating regional input coefficients using the LQ method was conducted by Schaffer and Chu (1969). Bonfiglio and Chelli (2008) compared the estimation accuracy of methods using LQ and demonstrated that the accuracy of FLQ and the augmented FLQ (AFLQ) could exceed that of other methods. Morrissey (2016) analyzed the current state of industrial specialization and clustering by estimating input coefficients using the LQ method for the two largest regions in Ireland. Lamonica and Chelli (2018) verified the estimation accuracy, variation, and bias of each variant of LQ based on the input–output table for countries. New methods such as FLQ+ by Flegg, Lamonica, Chelli, Recchioni, and Tohmo (2021) have also been developed in recent years.

The RAS method, which was originally introduced in the study of Stone (Bacharach, 1970), estimates an input coefficient matrix by iteratively adjusting the initial coefficients based on the total intermediate demands, total intermediate inputs, and total gross outputs of each industry. [1] The RAS method is highly accurate in estimating input coefficient matrices. For instance, Hewings (1977) successfully applied the RAS method to estimate the input coefficient matrix for Kansas in 1965 using the input coefficient matrix for Washington in 1963 as a basis. Several improvements to the RAS method have also been proposed. Lenzen, Moran, Geschke, and Kanemoto (2014) extended the RAS method to ensure that the sign of the input coefficients is not preserved in the iteration. Junius and Oosterhaven (2003) introduced the generalized RAS (GRAS), which can use input coefficient matrices that contain negative values as initial values. Several improvements to the GRAS method have been presented (Lemelin, 2009; Lenzen, Wood, & Gallego, 2007; Temursho, Oosterhaven, & Cardenete, 2021). [2] The estimation of regional input coefficient matrices using the RAS method has also been improved by numerous researchers. For example, Hiramatsu, Inoue, and Kato (2016) attempted to improve the estimation accuracy of the inter-regional input–output table of Japan using RAS with a real-code genetic algorithm. Holý and Šafr (2023) developed a multidimensional RAS and applied the method to estimate regional input–output tables in the Czech Republic.

Estimating input coefficients using non-survey methods relies on significant assumptions. Extensive debates have arisen regarding the practicality of these assumptions and the estimation accuracy in non-survey methods. Round (1983) provided a critical perspective on the theoretical aspects of non-survey methods. In terms of the empirical problems of non-survey methods, Riddington, Gibson, and Anderson (2006) estimated the economic spillovers of tourism expenditures for a group of small regions

[1] The total intermediate demands and total intermediate inputs of each industry are equal to the row and column sums, respectively, of the intermediate input matrix.

[2] Lahr and De Mesnard (2004) and Lenzen et al. (2014) summarized several other extensions of RAS.

in Scotland and demonstrated that estimates using the simple LQ and CILQ may lead to erroneous conclusions. Szabó (2015) highlighted that the non-survey method has both theoretical and empirical shortcomings, but argued that non-survey methods are necessary for estimating the input coefficients in data-deficient regions.

The non-survey method faces challenges in terms of reduced estimation accuracy owing to limited data availability and instability caused by data selection. Although the construction of an input–output table requires vast amounts of primary data, non-survey methods attempt to estimate the input coefficients using a relatively smaller dataset. In all variants of LQ, the quotients of the target region are obtained for a few variables, such as the total outputs of industries. In FLQ+ (Flegg et al., 2021), which is a relatively new derivation of LQ, a process for estimating additional coefficients exists, but the essential estimation method remains the same. As noted previously, the RAS method uses only the values based on input–output tables. Owing to the limited data that are used for estimation, a significant amount of data regarding the local economy is inevitably disregarded. Such discarding of information can negatively impact the estimation accuracy. In addition, both LQ and RAS must use input coefficients and other data from outside the time and region that are being predicted. For example, Holỳ and Šafr (2023) estimated intermediate inputs with multidimensional RAS when the total intermediate demands and total intermediate inputs of the target regions were known. In actual estimation, information on the input–output tables of the regions is often limited; therefore, in applying RAS, the total intermediate demands and total intermediate inputs of the regions must also be estimated. The accuracy of these methods relies heavily on the data that are used for the estimation.

In addition to these non-survey methods, other methods are available for estimating input coefficients using regression. Limited previous studies on estimation using regression, rather than LQ and RAS, exist. Gerking (1976) proposed a method to calculate input coefficients as estimates of partial regression coefficients through regression analysis, with the intermediate input as the objective variable and the gross output as the explanatory variable. Gerking's method aims to avoid the effects of measurement errors in the intermediate input and gross output data on the estimation results. Applying Gerking's method directly is difficult in small regions where these data are generally unavailable. Papadas and Hutchinson (2002) constructed an artificial neural network (ANN) as a forecasting model for the input coefficients, obtained forecasts of input coefficients for 1992 based on 1984 data from the United Kingdom, and attempted to compare them with those of RAS. In their study, the ANN was configured with input coefficients as the target variable. The ratio of intermediate demand to gross output in the input source industry and the ratio of intermediate input to gross output in the input target industry were used as the two explanatory variables. The model was trained with 49 observations in seven industries in the input–output table. However, the model was limited to two explanatory variables and one hidden layer, and a single model was applied to all 49 input coefficients. These limitations may have hindered the achievement of predictive accuracy beyond RAS.

Although various restrictions and drawbacks exist in predicting input coefficients using regression, it can alleviate certain issues that are associated with current mainstream non-survey methods. When input coefficients are predicted using regression

methods, various variables representing the local economic state can be included as explanatory variables. Therefore, more information can be included in the estimation than in LQ or RAS and higher prediction accuracy can be expected. Furthermore, unlike LQ and RAS, regression methods do not require additional data. Therefore, regression methods can avoid the instability in estimation accuracy that is associated with the estimation or selection of additional data. Recent developments have emerged in deep learning methods, which involve the estimation and utilization of multilayer ANNs for prediction. The introduction of deep learning into the forecasting of input coefficients using regression is expected to improve the prediction accuracy. The effectiveness of ANN-based forecasting in economics and finance has been demonstrated in several studies (Abbasimehr, Shabani, & Mohsen, 2020; Law, Li, Fong, & Han, 2019; Ramyar & Kianfar, 2019). However, when the dataset that is available for model estimation is small, overfitting can reduce the forecast accuracy. The available data for estimating regional input coefficients are generally limited in size. Therefore, overfitting is inevitable when deep learning is applied directly to predict regional input coefficients.

To mitigate the effects of overfitting, data augmentation, which involves manipulating the original data to increase their size, is used extensively in machine learning, particularly for data types such as images, text, and audio. Several studies have demonstrated the effectiveness of data augmentation (Dao et al., 2019; Wu, Zhang, Valiant, & Ré, 2020).

The objective of this study is to develop a deep learning method to predict more accurate and stable input coefficients than in conventional non-survey methods. To achieve this, we attempt to prevent overfitting by extending the mixup of H. Zhang, Cisse, Dauphin, and Lopez-Paz (2018) to regional data and augmenting the dataset that is used for model estimation. [3] First, for the input–output table and various macroeconomic variables, the data for virtual regions are augmented with the mixup from the data for prefectures and specific cities for which estimated input–output tables are available. We generate the data for the input coefficients and each explanatory variable used in the model estimation from these data of virtual regions. Next, we use these data to estimate an ANN with the input coefficients as the objective variable. Finally, the input coefficients are predicted using the ANN for the entire Japan and three Japanese cities.

The remainder of this paper is organized as follows. Section 2 describes the input coefficient prediction methodology using an ANN with mixup. Section 3 presents the results of predicting the input coefficients using our method. The results are discussed in Section 4, and the conclusions and limitations of this study are presented in Section 5.

[3] In a paper published in Japan in 2021, the author attempted to estimate the input coefficients by applying a partial mixup, focusing only on the additivity of regional data, and demonstrated that it could predict the coefficients for two regions in Japan with the same level of accuracy as RAS. However, the additivity-specific mixup can only use limited data as explanatory variables to ensure the vicinity of regions with different economic sizes. Building on the previous research, this study introduces a full mixup that considers scalar products for regional data, allowing more types of information to be included as explanatory variables. In addition, we attempt to make more accurate predictions by adding a new procedure for the establishing regional vicinity when training the model.

2 Extension of mixup to regional input coefficient prediction

Data augmentation is used to process the original data to generate new data. In general image recognition, machine learning is performed using the numerical information of the image as the explanatory variable and the label as the objective variable. A model that is trained using machine learning can tend to be overfitted if the amount of data used for training is small. Overfitting causes the trained model to fit the training data excessively well, resulting in poor prediction accuracy for unknown data. Data augmentation diversifies the training dataset by generating new data from the existing data. For image data, a new image is generated by slightly rotating and scaling the image in the original data; however, the label remains the same as that in the original image. In this manner, data augmentation creates new image–label pairs, thereby increasing the data size.

We review the procedure for data augmentation using a mixup based on H. Zhang et al. (2018). For the i-th individual in the data of size n ($i = 1, \ldots, n$), $\boldsymbol{x}_i$ is the value of the explanatory variable, y_i is the value of the objective variable, and the original training data are $(\boldsymbol{x}_i, y_i)$. In the mixup, a new individual $(\bar{\boldsymbol{x}}, \bar{y})$ is generated from two individuals $(\boldsymbol{x}_A, y_A)$ and $(\boldsymbol{x}_B, y_B)$ that are randomly selected from the training data, as follows:

$$\bar{\boldsymbol{x}} = \lambda \boldsymbol{x}_A + (1 - \lambda)\boldsymbol{x}_B \tag{1}$$

$$\bar{y} = \lambda y_A + (1 - \lambda) y_B, \tag{2}$$

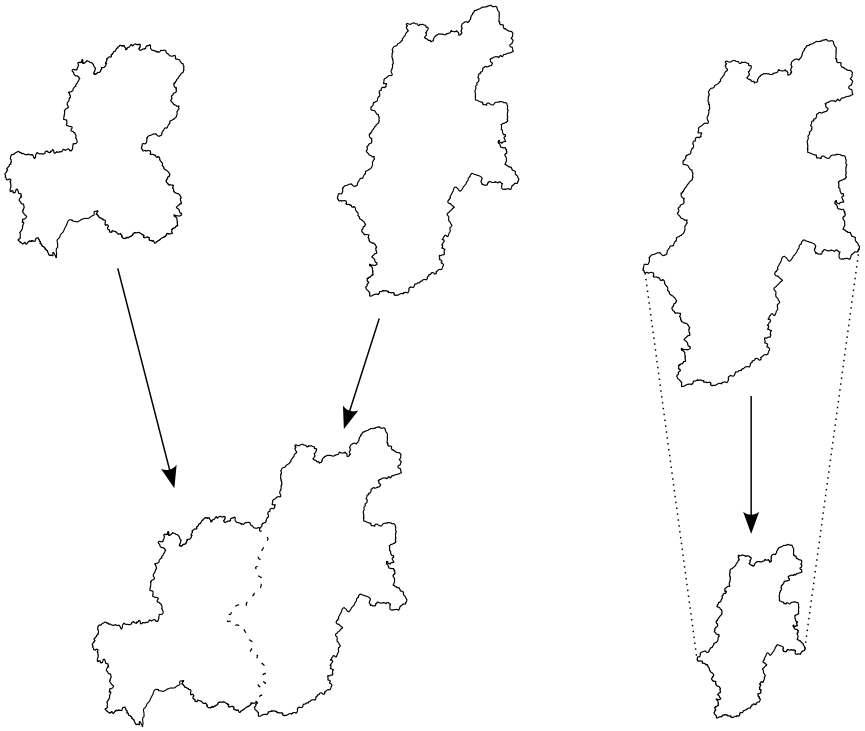
where $\lambda \in [0, 1]$ and $\lambda \sim \text{Beta}(\alpha, \alpha)$. When a mixup is performed, the random number that is generated from this beta distribution is used as λ.

Let $\boldsymbol{x}_i$ be the numerical density of each pixel in the grayscale image and y_i be the label of that image. [4] In this case, let us suppose that two images $(\boldsymbol{x}_1, y_1)$ and $(\boldsymbol{x}_2, y_2)$ are randomly selected. The mixup produces a new composite image by diluting $\boldsymbol{x}_1$ to $\lambda \times 100\%$ and $\boldsymbol{x}_2$ to $(1 - \lambda) \times 100\%$. The label of the new image is set as a linear combination with weights $(\lambda, 1-\lambda)$ of the labels of the two original images. According to H. Zhang et al. (2018), a mixup is equivalent to vicinal risk minimization (Chapelle, Weston, Bottou, & Vapnik, 2000) with a certain generic vicinity distribution for each observation $(\boldsymbol{x}_i, y_i)$. In addition, L. Zhang, Deng, Kawaguchi, Ghorbani, and Zou (2021) demonstrated that the loss function in a mixup contains regularization terms.

Data augmentation generates data based on prior information or knowledge. In conventional data augmentation, which transforms single-image data, the prior knowledge is the invariance of the image data; that is, the labels remain unchanged with respect to image rotation and scaling. In a mixup, the prior knowledge is that "linear interpolation of feature vectors should lead to linear interpolation of the associated targets" (H. Zhang et al., 2018).

We apply the mixup to quantitative regional macroeconomic variables for the regions. Mixup can only be applied to specific data types. For example, it is difficult to adapt the prior knowledge that a mixup assumes for qualitative values such as gender

[4] If the images are categorized by one-hot encoding, y_i is a vector.



(a) Composition (b) Scaling

Fig. 1 Images showing the composition and scaling of regions. Maps of Nagano and Gifu prefectures in Japan are shown as examples, which were created using Matplotlib and GeoPandas in Python based on the Digital National Land Information (administrative area data) of the Ministry of Land, Infrastructure, Transport and Tourism (https://nlftp.mlit.go.jp/ksj/index.html)

and occupation and rating scales such as technical support satisfaction. However, operations such as vector sums and scalar products of variables can be meaningful for numerous quantitative regional macroeconomic variables. For instance, the total population of North American countries such as the United States, Canada, and Mexico is equal to that of North America. The same holds for quantitative variables such as income and number of establishments. That is, if $\boldsymbol{r}_k$ is a vector that consists of the quantitative economic variables for region $k(k = 1, \ldots, K)$, the vector sum of the quantitative economic variables for each region, $\sum_k \boldsymbol{r}_k$, corresponds to the hypothetical composite of these regions (Figure 1a).

The scalar product of a quantitative variable implies the scaling of values. If the North American population is r, half of it can be calculated as $0.5 \times r$. For a vector $\boldsymbol{r}$

comprising quantitative economic variables, $\lambda \boldsymbol{r}$ multiplies all quantitative variables in $\boldsymbol{r}$ by λ. This operation implies the hypothetical expansion or contraction of a region (Figure 1b).

A new virtual region can be generated using a linear interpolation that combines a set of regions by scaling them. The observation vector of the virtual region is generated using the following equation:

$$\bar{\boldsymbol{r}} = \sum \lambda_k \boldsymbol{r}_k, \tag{3}$$

where $\boldsymbol{r}_k$ $(k = 1, \ldots, K)$ is the vector of observations and λ_k, $(k = 1, \ldots, K)$ is a constant that scales them for region k.

The variables in our model were calculated from the observation of a virtual region that was obtained using the linear interpolation described above. The values contained in $\bar{\boldsymbol{r}}$ become the values of the quantitative variables. When indices or ratios are used as variables, the quantities that are the sources of the indices and ratios are included in $\boldsymbol{r}_k$, and the values of these variables are calculated after linear interpolation. For example, when using the unemployment rate as an explanatory variable, the number of unemployed individuals and labor force are incorporated into $\boldsymbol{r}_k$. Subsequently, the unemployment rate can be calculated based on data from a virtual region obtained through linear interpolation. For a competitive import type input–output table, the intermediate inputs of a region include inputs from other regions. However, as explained in the Appendix, the sum of the intermediate inputs for a group of regions is equal to the intermediate inputs of the combined group of these regions measured as a single region. The scalar product of the intermediate inputs of a region is equal to its intermediate inputs when the region is scaled by the scalar value. These properties also hold for the gross output. [5] Therefore, it is feasible to calculate the input coefficient of a virtual region obtained by linear interpolation as the ratio of the intermediate input to the gross output.

We make the following assumption to derive prior knowledge, which is a prerequisite for the mixup:

Assumption. *Let the input coefficient $a_{i,j}$ from industry i to j be the objective variable and $\boldsymbol{x}^* = (x_1, \ldots, x_m)'$ be the explanatory variable. Subsequently, $a_{i,j} = f^*_{i,j}(\boldsymbol{x}^*)$ holds uniquely for all regions.*

This is similar to the usual econometric model assuming one regression equation for all observed individuals, which means that the input coefficient $a_{i,j}$ is determined by only one function $f^*_{i,j}$ for all regions, including the virtual region generated by linear interpolation. The values in the input–output table are calculated from various primary data based on predefined rules. In principle, these rules are the same for all regions; therefore, making the above assumption for input coefficient predictions is natural. With this assumption, if we denote the values of the primary data as $\boldsymbol{x}^*$ and the rule for calculating the values of the input–output table from the primary data as the function $\boldsymbol{F}^*$, the input–output table can be expressed as $\boldsymbol{F}^*(\boldsymbol{x}^*)$. For the input coefficients, if $f^*_{i,j}$ is the rule for computing $a_{i,j}$ from the primary data,

$$a_{i,j} = f^*_{i,j}(\boldsymbol{x}^*). \tag{4}$$

[5] As shown in the Appendix, the sum of the gross output of a group of regions is equal to the gross output of the regions when they are considered as a single region.

Based on the above assumption, if the primary data vector of a hypothetical region v generated by linear interpolation is $\boldsymbol{x}_v^*$, its input coefficient can be calculated as $f_{i,j}^*(\boldsymbol{x}_v^*)$. However, the input coefficient $a_{i,j}^v$ for region v is already obtained by the mixup. Thus, $a_{i,j}^v = f_{i,j}^*(\boldsymbol{x}_v^*)$. Eq. (4) is the prior knowledge in this analysis. That is, the mixup can be applied to estimating the input coefficients by changing the prior knowledge of the original mixup, as follows: For a virtual region obtained by linear interpolation, the feature vectors should lead to an associated target.

The regional composition and scaling in the above mixup do not represent actual regional mergers and divisions. If two cities merge, the values of the economic variables after the merger should differ from the sum of the values before the merger owing to changes in the economic structure. Alternatively, even if the area of a city is divided into two equal parts, this does not necessarily translate to equal populations in those parts. Our mixup approach involves composition and scaling, where the data values are derived from the synthesis of individual areas as well as from the scaling of an area by a constant factor. These operations do not assume any changes in the economic structure.

In this study, $f_{i,j}^*$ is approximated using a multilayer ANN. In general, the primary data required to compute input–output tables are not sufficiently measured in small areas such as cities. In this case, the original explanatory variable $\boldsymbol{x}^*$ must be replaced by another variable $\boldsymbol{x}$ that is derived from the available data. Similarly, the actual function $f_{i,j}^*$ is approximated by the function $f_{i,j}(\boldsymbol{x})$. Because this study aims to predict the input coefficients in a small region with high accuracy, a multilayer ANN is established as $f_{i,j}(\boldsymbol{x})$.

3 Empirical analysis for Japan

Following data augmentation by mixup for some prefectures and ordinance-designated cities in Japan, deep learning was performed using the input coefficients as the objective variable. We verified the prediction accuracy of the trained model for the input coefficients for all of Japan and predicted the input coefficients for certain cities in Japan. This section details the forecasting methodology and results.

The data used in this analysis are presented in Table 1. [6] All input–output tables used in the analysis are of the competitive import type. Data from 2015 were used to predict the input coefficients for 2015. For the data that were used as explanatory variables, where the data for 2015 were unavailable, we substituted data from the most recent available year, such as 2013 or 2014. The index c attached to some variables in Table 1 means industry. For each variable with c, there were 619 industries in the minor classification ($c = 1, \ldots, 619$) and 17 industries in the large classification ($c = 1, \ldots, 17$). Data with non-numeric values were excluded before training the model. The regions for which all of these data are currently available were included in the analysis (Table 2). Regions with outliers were excluded from the analysis even if all data were available. Japan is divided into 47 prefectures, each of which is further

[6] All data used in this study are publicly available from public institutions and can be obtained from the sources listed in Table 1. Links to the input–output table for each region were compiled on the Pacific Rim Association for Input–Output Analysis (PAPAIOS) website (http://www.gakkai.ne.jp/papaios/en/io_j.html). The formatted datasets are available from the corresponding author upon request.

Table 1 Dataset used in this study. The minor and large classifications refer to the industry classifications of the Economic Census

Variable name	Data	Source
$\mathrm{SFirm}_{c,k}$	Number of establishments (minor classification)	2014 Economic Census for Business Frame
$\mathrm{SEmp}_{c,k}$	Number of employees (minor classification)	Ibid.
$\mathrm{VA}_{c,k}$	Added value (large classification)	2012 Economic Census for Business Activity
$\mathrm{Sales}_{c,k}$	Sales (large classification)	Ibid.
$\mathrm{Firm}_{c,k}$	Number of establishments (large classification)	Ibid.
Income_k	Taxable income	Statistical Observations of Prefectures 2015 and Statistical Observations of Municipalities 2015
TP_k	Taxpayer	Ibid.
PopLF_k	Population in labor force	Ibid.
Unemp_k	Number of unemployed persons	Ibid.
$\mathrm{Pop15}_k$	Total population (15 and over)	Calculated by the author from Statistical Observations of Prefectures 2015 and Statistical Observations of municipalities 2015
$A_{i,j,k}$	Intermediate input (12 industries)	Calculated by the author from the input–output table for each prefecture and city.
$Y_{j,k}$	Gross output (12 industries)	Ibid.

divided into several cities, towns, and villages. The names of the cities in Table 2 are followed by the prefectures in which they are located. Among the regions in the table, the targets of the forecast (Japan, Gujo City in Gifu Prefecture, Sapporo City in Hokkaido, and Okayama City in the Okayama Prefecture) were excluded to obtain the values of the input coefficients and explanatory variables in Table 3 using mixup. The input coefficients were calculated after recompiling the input–output tables for each region to ensure that there were 12 industry sectors, as shown in Table 4. For an input coefficient $a_{i,j}$, i and j indicate the values of the order in Table 4. For example, $a_{1,2}$ represents the input coefficients from agriculture to mining.

The data for the model estimation were augmented by the mixup based on the data in Table 1 for prefectures and some cities for which the estimated input–output table is publicly available, excluding the regions to be inferred. Because most of the target areas are prefectures, as shown in Table 2, if a mixup is performed directly on these data, the generated data are likely to be concentrated close to the prefectures. Consequently, the trained ANN reflects the prefectures rather than all of Japan or its cities.

In this analysis, the sizes of all regions were scaled based on a single variable before the mixup and the generated regions were converted to the size of the region to be predicted. Prior to performing the mixup, a scalar product with (1/Pop15) was calculated for the observed values of each prefecture and city. This transformation ensured that these areas were scaled so that the population aged 15 years and above

Table 2 Target areas for analysis. The city names are followed by the name of the prefecture to which the city belongs

For training	
Prefectures	Hokkaido, Aomori, Iwate, Miyagi, Akita, Yamagata, Fukushima, Ibaraki, Tochigi, Gunma, Saitama, Chiba, Tokyo, Kanagawa, Niigata, Toyama, Yamanashi, Nagano, Gifu, Shizuoka, Aichi, Mie, Shiga, Kyoto, Osaka, Hyogo, Wakayama, Shimane, Okayama, Hiroshima, Yamaguchi, Tokushima, Kagawa, Ehime, Kochi, Fukuoka, Saga, Nagasaki, Kumamoto, Oita, Miyazaki, Kagoshima
Cities	Saitama (Saitama), Yokohama (Kanagawa), Kawasaki (Kanagawa), Fukuoka (Fukuoka)
For inference	
	Japan, Gujo (Gifu), Sapporo (Hokkaido), Okayama (Okayama)

Table 3 Variables used for analysis

Name	Definition
Input coefficient	$= A_{i,j,k}/Y_{j,k}$
Number of establishments (minor classification)	$= \mathrm{SFirm}_{c,k}$
Composition ratio of number of establishments (minor classification)	$= \mathrm{SFirm}_{c,k}/\sum_c \mathrm{SFirm}_{c,k}$
Number of employees (minor classification)	$= \mathrm{SEmp}_{c,k}$
Composition ratio of number of employees (minor classification)	$= \mathrm{SEmp}_{c,k}/\sum_c \mathrm{SEmp}_{c,k}$
Added value (large classification)	$= \mathrm{VA}_{c,k}$
Added value per firm (large classification)	$= \mathrm{VA}_{c,k}/\mathrm{Firm}_{c,k}$
Sales (large classification)	$= \mathrm{Sales}_{c,k}$
Sales per firm (large classification)	$= \mathrm{Sales}_{c,k}/\mathrm{Firm}_{c,k}$
Taxable income	$= \mathrm{Income}_k$
Taxable income per taxpayer	$= \mathrm{Income}_k/\mathrm{TP}$
Population in labor force	$= \mathrm{PopLF}_k$
Labor force population ratio	$= \mathrm{PopLF}_k/\mathrm{Pop15}_k$
Unemployment rate	$= \mathrm{Unemp}_k/\mathrm{PopLF}_k$

had a value of one. After performing a mixup on these data, the generated virtual regions were expanded such that the value of the population aged 15 years and above was close to the level of those of the regions to be forecasted. For the projection of the input coefficients for the entire Japan, a scalar product was calculated for each observation obtained by the mixup, with the Pop15 of Japan as a constant. To predict the input coefficients for a city, we first established a uniform distribution with the minimum and maximum of Pop15 in all Japanese cities as the lower and upper bounds, respectively. Then, each time the values of a virtual area were created in the mixup, a scalar product was obtained for these values using a random number that was generated from the uniform distribution as a constant. This transformed dataset was used for training.

In this mixup, two to five regions were randomly selected to generate data for a virtual region. Unlike the original mixup, the current mixup is not limited to two observations. In the original mixup, λ is assumed to follow a beta distribution $B(\alpha, \alpha)$, whereas in this method, $(\lambda_1, \ldots, \lambda_K)$ is assumed to follow a Dirichlet distribution. In

Table 4 Industry classification

Order	Industry
1	Agriculture (agriculture, forestry, and fisheries)
2	Mining
3	Manufacturing
4	Construction
5	Energy (electricity, gas, and water)
6	Trade
7	Finance (finance, insurance, and real estate)
8	Transportation (transportation and postal)
9	Communication (information and communication)
10	Public business
11	Services
12	Other industry

addition, the K parameters of the Dirichlet distribution are assumed to share the same value, α. The number of regions ($= K$) for the mixup is a random number that is generated from a discrete uniform distribution with a lower bound of two and an upper bound of five. Generating data from numerous regions can improve the extrapolation accuracy as the data differ from the original set of regions. However, in the preliminary analysis, when the mixup was performed for numerous regions, the accuracy of the predictions made by the trained model tended to decrease. This may be because the features of the generated data are homogenized when numerous regions are composited. Therefore, to prevent the number of target regions from becoming excessively large, the maximum value was set to five and the number of regions was randomly selected. During the mixup, prefectures and cities in an inclusion relationship were not selected simultaneously. For instance, Sapporo was included in Hokkaido, so we did not select a group of regions that included these two regions. This is because obtaining the sum of these areas for values that include transfers, such as the intermediate input and gross output, is infeasible.

The values of the input coefficients and their explanatory variables were calculated using a dataset generated by the mixup. These variables are summarized in Table 3. When training the model, the principal component scores of the explanatory variables were calculated and used as inputs. We used 50 principal component scores from the highest cumulative contribution ratio. Some input coefficients were sufficiently small such that the derivative calculated by backpropagation could approach zero. Therefore, the following transformation, which is similar to standardization, was performed on the input coefficients during deep learning:

$$\begin{aligned} \hat{a}_{i,j} &= \frac{a_{i,j} - a_{\mathrm{L}}}{a_{\mathrm{U}} - a_{\mathrm{L}}} \\ a_{\mathrm{L}} &= \max(0, a_{i,j}^{\min}) \\ a_{\mathrm{U}} &= \min(1, a_{i,j}^{\max} + 0.5(a_{i,j}^{\max} - a_{i,j}^{\min})), \end{aligned} \tag{5}$$

where $a_{i,j}^{\min}$ and $a_{i,j}^{\max}$ are the minimum and maximum values of $a_{i,j}$ in the training data, respectively. This transformation uses $a_{i,j}^{\max} + 0.5(a_{i,j}^{\max} - a_{i,j}^{\min})$ as the candidate

for the maximum value. This is because our transformation tended to be slightly more accurate than the transformation obtained using $a_{i,j}^{\max}$ as the candidate for the maximum value in a preliminary analysis for the entire Japan. The trained model then predicted $\hat{a}_{i,j}$. By transforming $\hat{a}_{i,j}$ inversely, we obtained the estimated value of $a_{i,j}$.

The multilayer ANN used for this training is shown in Figure 2. The multilayer ANN had a relatively standard shape, with a fully connected layer with 512 nodes and a batch normalization layer as a single pair; ten of these pairs were connected as an hidden layer. In the output layer, $\hat{a}_{i,j}$ was obtained by feeding a linear combination of the hidden layer outputs into a sigmoid function. In the fully connected layers of the hidden layer, the activation function was set as an exponential linear unit (ELU) and the initial values of the parameters were determined using the method in He, Zhang, Ren, and Sun (2015).

For training of the ANN, L2 regularization with a hyperparameter value of 0.01 was applied to all weight parameters of the fully connected layers. The learning rate varied exponentially and cyclically, with minimum and initial values of 0.0001, a maximum value of 0.01, and a step size of 50 (Smith, 2017). When training the model, the parameters were estimated using stochastic gradient descent with the mean squared error as the loss function and the mini-batch size was set to 32. The parameters were updated using the Nesterov accelerated gradient (NAG) with a momentum of 0.9 to accelerate the optimization. [7] The step size for optimization was set to a variable value (the maximum number was 200) with early stopping. Specifically, the optimization process was halted when the mean squared error from the validation data increased for ten consecutive steps. The model obtained before the increase in the mean squared error was then selected as the final training result. [8]

The specific analytical procedure is described below. The data for prefectures and cities in the "For training" row of Table 2 were the initial data in our analysis. After transforming the original data using a scalar product with (1/Pop15) as a constant, a mixup was used to generate data for 50,000 regions. The value of α of the Dirichlet distribution was set to one according to the prediction accuracy in the preliminary analysis. This setting renders the Dirichlet distribution a multivariate uniform distribution. The generated data were scaled by the population value of individuals aged 15 years or older in the target regions to obtain data for the virtual regions close to the targets. From these data, the values of the input coefficients and explanatory variables were calculated. The input coefficients were transformed using Eq. (5). The principal component scores computed from the explanatory variables were used as inputs to the model. Among the data generated by the mixup, 40,000 instances were used for training, 10,000 were used for testing, and 20% of the training data was used for validation. The ANN shown in Figure 2 was trained to obtain a prediction model for the input coefficients using the training data. Thereafter, we verified the prediction accuracy of the trained ANN with the test data. For each region to be predicted, $\hat{a}_{i,j}$ was calculated from the prediction model using the principal component scores of the explanatory variables as inputs. The predicted value of the input coefficient ($a_{i,j}$)

[7] The NAG was proposed in a 1983 paper authored by a Russian scholar named Yurii E. Nesterov. For more information on the NAG, refer to Botev, Lever, and Barber (2017).

[8] These settings in this model training are based on Géron (2019).

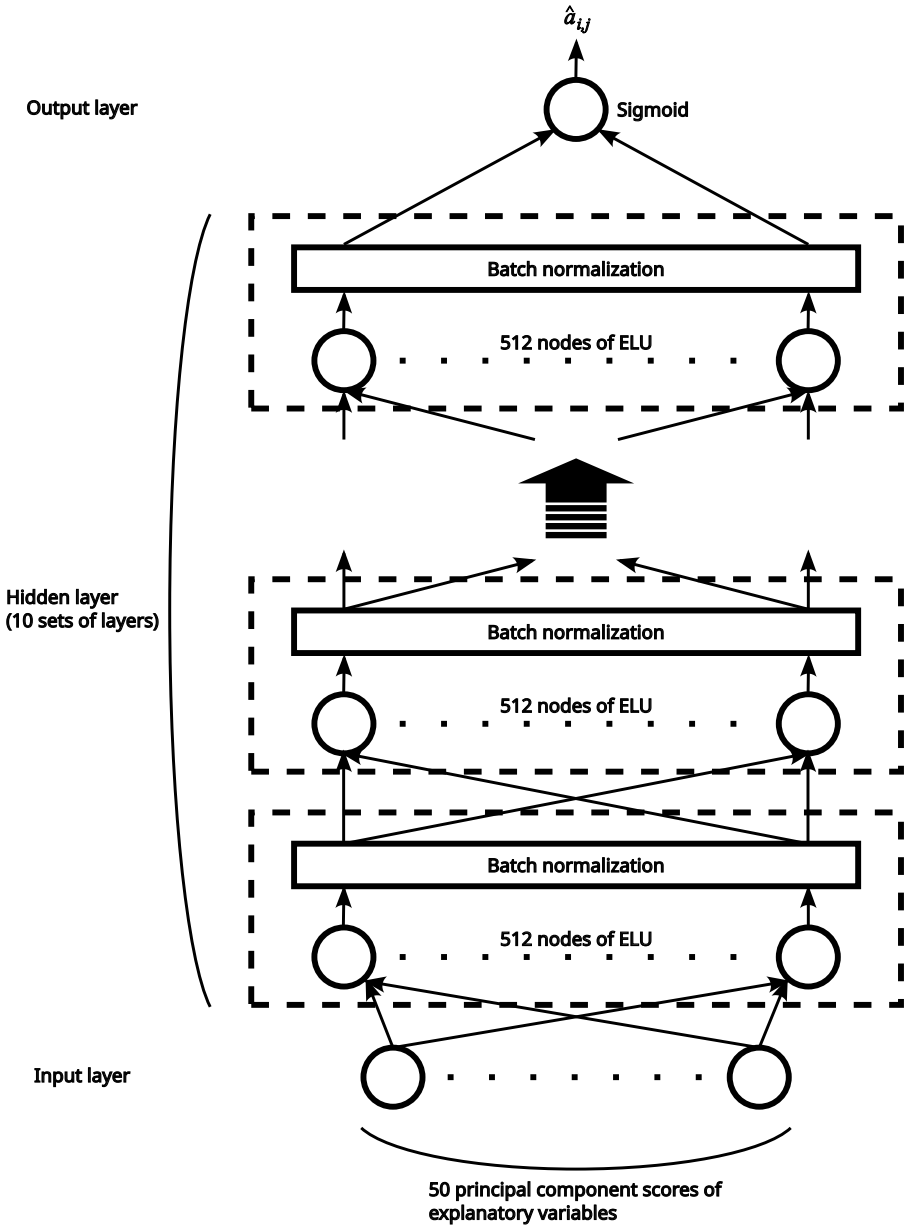


Fig. 2 Multilayer ANN in this analysis. This figure was created using Inkscape

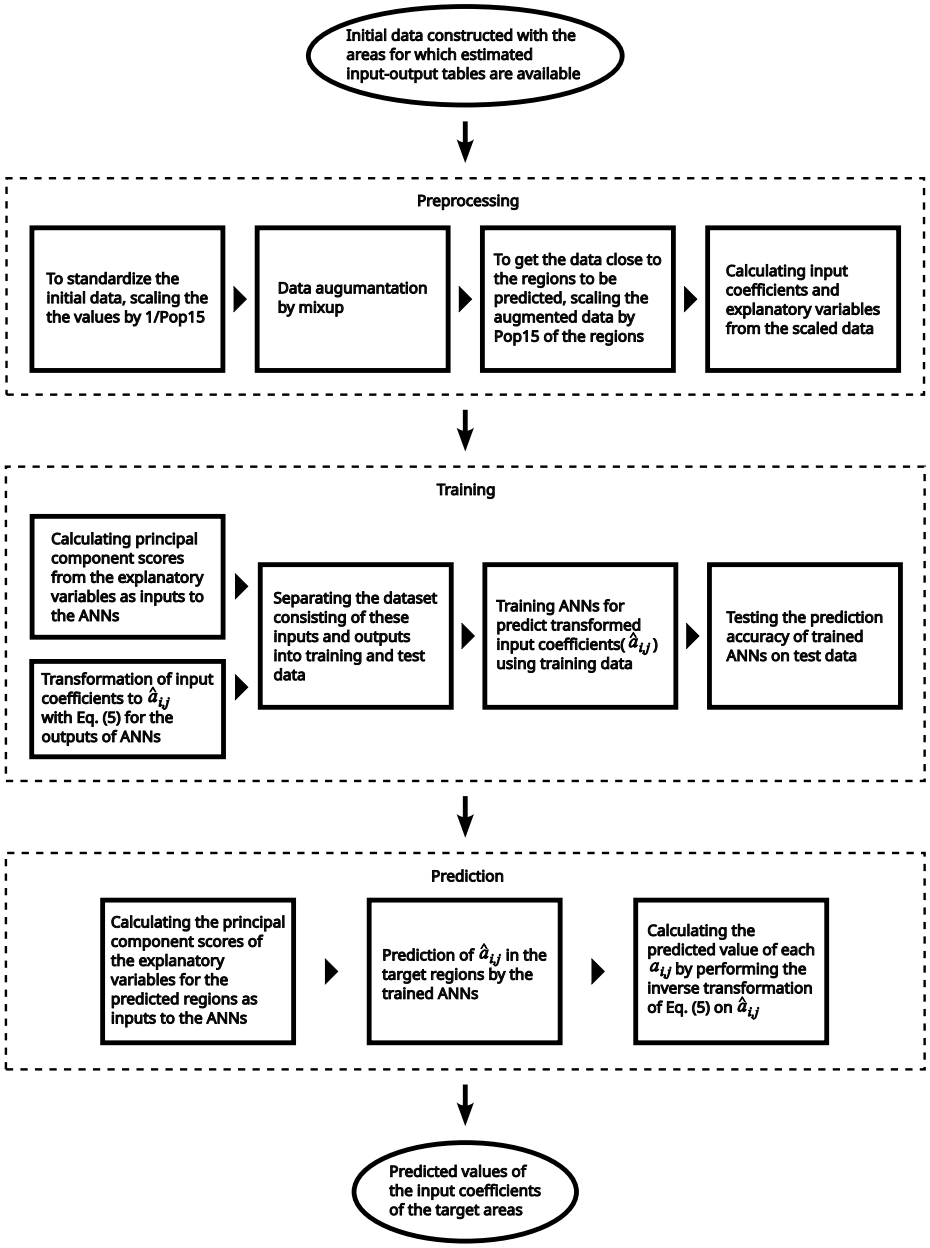


Fig. 3 Model training and inference flow. This figure was created using Inkscape

Table 5 Prediction errors of input coefficients for the entire Japan

	ANN	FLQ			RAS (based on prefectures)			RAS (based on 2011)
		Min	Mean	Max	Min	Mean	Max	
STPE	0.0434	0.3527	1.0329	3.1693	0.0679	0.1697	0.2829	0.1105
MAD	0.0017	0.0136	0.0398	0.1221	0.0026	0.0065	0.0109	0.0043
U2	0.0468	0.4615	1.7153	5.4916	0.0750	0.2257	0.3913	0.1148
RMSE	0.0035	0.0344	0.1279	0.4094	0.0056	0.0168	0.0292	0.0086
MAPE	0.0764	0.3519	0.9990	3.7559	0.1162	0.3270	0.8963	0.1940

was calculated by performing an inverse transformation of Eq. (5). Figure 3 shows the flow of the above procedures. [9]

The aforementioned model training and inferences were performed for each input coefficient. In this study, industries were classified into 12 categories; therefore, the number of input coefficients covered was $12 \times 12 = 144$. However, input coefficients that were consistently zero across all prefectures and cities in the original data were excluded from the training, resulting in a final predicted value of zero for those coefficients. Thus, 131 input coefficients were predicted.

First, we verified the accuracy of the ANN-based predictions of the input coefficients for the entire Japan. The published input–output table for Japan was estimated with relatively high accuracy using the survey method. We verified the accuracy of the proposed method by comparing the estimates of the input coefficients. Table 5 lists the prediction accuracy indices that were derived from the differences between the predicted and published values of the input coefficients, with smaller values indicating higher accuracy. Refer to Hosoe (2014) for the indices in Table 5, which were calculated using the following equations:

$$
\begin{aligned}
\text{STPE} &= \sum_{i,j} |\tilde{a}_{i,j} - a_{i,j}| \,/ \sum_{i,j} a_{i,j} \\
\text{MAD} &= \sum_{i,j} |\tilde{a}_{i,j} - a_{i,j}| \,/ N_a \\
\text{U}_2 &= \sqrt{\sum_{i,j} (\tilde{a}_{i,j} - a_{i,j})^2} \Big/ \sqrt{\sum_{i,j} a_{i,j}^2} \\
\text{RMSE} &= \sqrt{\left[\sum_{i,j} (\tilde{a}_{i,j} - a_{i,j})^2 \right] / N_a} \\
\text{MAPE} &= (1/N_a) \sum_{i,j} |(\tilde{a}_{i,j} - a_{i,j})/a_{i,j}| \,,
\end{aligned}
$$

where $a_{i,j}$ is the published value of the input coefficient and $\tilde{a}_{i,j}$ is its estimated value. The ANN column shows the prediction accuracy of the deep learning method.

[9] F# was used for the mixup and other data processing, and TensorFlow was used for deep learning and prediction in Python.

The FLQ columns show the prediction accuracy of $\tilde{a}_{i,j}$ obtained using the following estimation equation from Flegg et al. (2021):

$$a^r_{i,j} = \begin{cases} \tilde{a}_{i,j}\mathrm{FLQ}_{i,j} & \mathrm{FLQ} < 1 \\ \tilde{a}_{i,j} & \mathrm{FLQ} \geq 1 \end{cases}$$

$$\mathrm{FLQ}_{i,j} = \begin{cases} \lambda(x^r_i/x^n_i)/(x^r_j/x^n_j) & i \neq j \\ \lambda(x^r_i/x^n_i)/(x^r/x^n) & i = j \end{cases}$$

$$\lambda = \left[\log_2(1 + (x^r/x^n))\right]^2 .$$

In the inference using FLQ, x^r_i is the gross output of industry i in region r, x^n_i is the gross output of industry i nationwide, x^r is the gross output in region r, and x^n is the gross output nationwide. We set $\delta = 0.1$. [10] From this equation, the predicted value of the input coefficients in the national input–output table was calculated as $\tilde{a}_{i,j}$ when the actual input coefficients for each of the 42 prefectures were provided for $a^r_{i,j}$. The RAS columns indicate the accuracy of the input coefficients estimated using RAS. The RAS results are shown for the case based on the input coefficients for each prefecture in 2015 and for the case based on the input coefficients for the entire Japan in 2011. The minimum, average, and maximum accuracies for FLQ and RAS are listed side by side because the prediction results differed depending on the data that were used as a reference. In addition, because RAS used actual values from the 2015 national input–output table for intermediate inputs, intermediate demands, and gross outputs, the forecast errors, which may occur in practice, were zero and did not affect the prediction accuracies of the input coefficients.

The values in Table 5 indicate that the deep learning method achieved higher and more stable accuracy in predicting input coefficients for the entire Japan compared to FLQ and RAS. We can observe that the prediction errors of the deep learning were smaller than those of FLQ and RAS; that is, the proposed method could accurately predict the national input coefficients. In addition, FLQ and RAS had different prediction errors depending on the reference input coefficients. However, such fluctuations in the errors did not occur at all with the deep learning. Moreover, the deep learning errors were still lower than those of RAS, which uses actual values for intermediate inputs, intermediate demands, and gross outputs.

Our next step was to examine the prediction accuracy of the city-level input coefficients. Unlike the national input–output tables, the published city-level input–output tables are typically inferred by hybrid or non-survey methods, which have larger inference errors for true input–output tables than survey methods. Therefore, rigorously measuring the accuracy of the input coefficient forecasting methods for cities is difficult. The following part of this section presents the prediction results of the input coefficients using deep learning for three cities (Gujo, Sapporo, and Okayama) and discusses their characteristics based on the errors against published input coefficients.

[10] For δ, we adopted the value from the candidates $(0.1, 0.2, 0.3, \ldots, 0.9)$ with the smallest overall errors in Table 5.

Table 6 Prediction errors of input coefficients in the three cities

	Gujo		Sapporo		Okayama	
	ANN	RAS	ANN	RAS	ANN	RAS
STPE	0.2508	0.2753	0.2971	0.2728	0.2422	0.1811
MAD	0.0104	0.0104	0.0116	0.0097	0.0097	0.0066
U_2	0.3156	0.5190	0.4190	0.3820	0.3335	0.1991
RMSE	0.0234	0.0368	0.0298	0.0259	0.0239	0.0136
MAPE	0.3959	0.0818	0.8203	0.6214	0.5141	0.2884

Gujo City was selected as the forecast target to represent cities with relatively small economies. Sapporo and Okayama were randomly selected to represent large cities. [11]

Table 6 lists the prediction errors relative to the published values of the input coefficients for each city, as shown in Table 5. The forecast error of RAS is shown for comparison with that of the ANN forecast. In the inference of RAS, the input coefficients of the prefecture in which each city is located were used as initial values and the actual estimates published by each city were used for the total intermediate demands, total intermediate inputs, and total gross outputs.

Table 6 shows that the prediction accuracy varied by city. While the error values of the ANN were smaller in Gujo, except for MAPE, the error values of RAS were generally smaller in Sapporo and Okayama. However, published estimates of the total intermediate demands, intermediate inputs, and gross outputs were used to calculate RAS. Thus, the actual RAS forecasts would include those estimation errors.

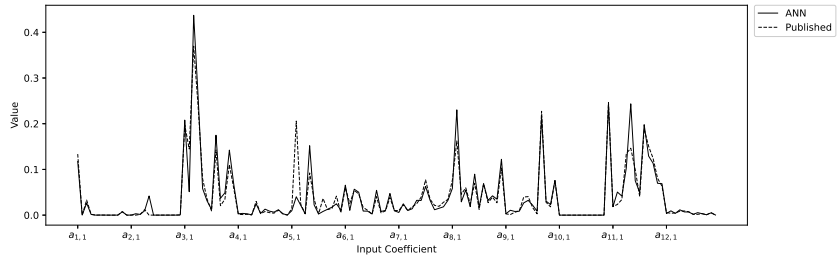
The prediction errors for each input coefficient were examined for these cities. Figures 4a, 4b, and 4c show the deep learning predictions (ANN) and published city estimates (Published) for Gujo, Sapporo, and Okayama, respectively. In all figures, the input coefficients are indicated on the horizontal axis in the order $a_{1,1}, a_{1,2}, \ldots, a_{12,11}, a_{12,12}$. These figures were prepared according to the method described in Papadas and Hutchinson (2002).

For the input coefficients of the three cities, the ANN estimates were generally close to the published values. However, the estimates and published values differed substantially for certain input coefficients, such as $a_{1,1}$ and $a_{8,2}$.

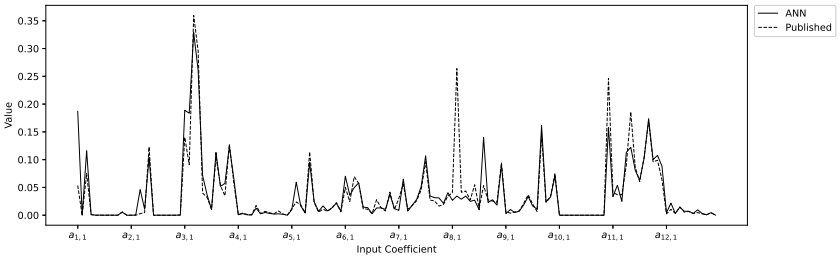
4 Discussion

To predict the input coefficients in the regional input–output table with higher accuracy, we developed a method using an ANN and forecasted the coefficients of Japan and three of its cities. Our method can effectively predict the input coefficients for relatively large regions. For the entire Japan, the prediction accuracy of our method was higher and more stable than that of the conventional non-survey methods. In the forecast for the three cities in Japan, the method produced results that were generally close to the published values. Compared to the results of Papadas and Hutchinson (2002), this study achieved an improvement in the prediction accuracy of the ANN,

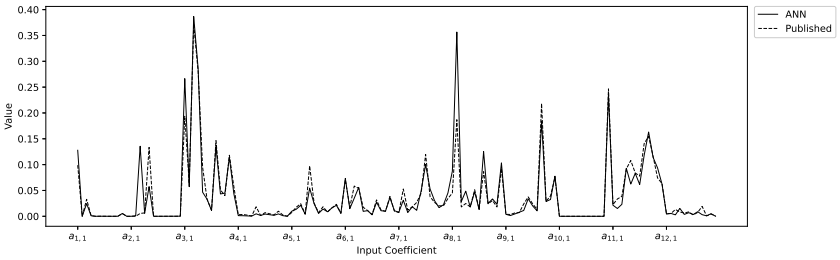
[11] In Japan, the economic scale of ordinance-designated cities, such as Sapporo and Okayama, is larger than that of other cities.



(a) Input coefficients of Gujo



(b) Input coefficients of Sapporo



(c) Input coefficients of Okayama

Fig. 4 Predicted input coefficients for the three cities using deep learning. These figures were created using Matplotlib in Python

which reached the same level as that of RAS. In addition, our method requires fewer assumptions than LQ and RAS and can therefore address the predictive instability of these methods.

Compared to the application of ANNs, such as image recognition, the proposed method achieved highly accurate prediction of the input coefficients using a relatively simple model. The high accuracy achieved in this study may be attributed to the less complex process of generating an input–output table. The process of image recognition is extremely sophisticated and requires complex models for its approximation. However, the rules for compiling an input–output table have been defined and are

simpler than those for image recognition. Therefore, a relatively simple model can approximate the process of generating input coefficients.

In recent years, various models have been developed to improve the prediction accuracy in machine learning. In this study, a simple ANN was used as the prediction model; however, the prediction accuracy of the input coefficients can be further improved by applying these more advanced models.

The model trained in this study integrates the various methods used to estimate the input–output table for each prefecture and city. Most input–output tables published by prefectures are estimated using available primary data and independent surveys according to the general guidelines set by each prefecture. For cities, input–output tables are estimated using different methods for each city, such as hybrid and non-survey methods, because the primary data for constructing input–output tables are extremely limited. [12] As this study assumes a single model $f_{i,j}(\boldsymbol{x})$ for the input coefficient estimation method, the trained model is a synthesis of the estimation methods for each region. Thus, the methodology used in this study has a meta-analytical aspect.

The training of deep learning models is time consuming. In this study, model learning was performed for each of the 131 input coefficients, and the computation time required for model learning and prediction was extremely long compared to that of conventional estimation methods. However, this problem can be solved by improving the computing resources. For example, if the models to be trained are distributed over n computers, the computation time can, in principle, be reduced to $1/n$ compared with the case in which only one computer is used. Alternatively, the speed can be increased using fast processors. Therefore, the computational time required for deep learning becomes minimal if sufficient computing resources are available.

5 Conclusion

We have presented an estimation method using an ANN with mixup to predict input coefficients more accurately and stably than the conventional non-survey method. The approach in this study can be applied to forecast input coefficients of other areas and countries. This would contribute to further refinement of the estimation of regional input–output tables and the quantitative analyses of the regions, such as spillover effects. The method can be further extended to other economic data, such as price indices, where the generation process can be expressed as a single function.

However, this method currently has the following limitations:

First, our method may be less accurate for regions with specific industries. In large regions such as prefectures, resources are distributed among numerous industries, whereas in small regions such as towns and villages, resources may be concentrated in specific industries. As the mixup employed in this study generates a virtual region from prefectures and cities, the distribution of resources in this virtual region is similar to that in the original regions. Hence, for cities, towns, and villages that are heavily reliant on a specific industry, the dissimilarity between these areas and the

[12] However, explanations of detailed estimation methods for input–output tables are seldom provided in these cities.

region created by the mixup process reduces the accuracy of the predictive model. The predictions for the cities in this study exhibit deviations between the predicted and published values for certain input coefficients. These differences may be because the ANN does not fully capture information regarding the characteristics of each industry included in the published values. However, the published values for each city also contain estimation errors with respect to the true values.

Second, from a prediction accuracy perspective, applying the proposed method to future predictions is difficult. Using our method, we trained an ANN on 2011 data, predicted the input coefficients for 2015, and found that the accuracy was extremely poor. As in most econometric models, the current situation at the time of measurement is captured in the ANN through the dataset. Because economic conditions changed between 2011 and 2015, predicting the input coefficients for 2015 accurately using a model trained with a 2011 dataset was difficult.

Third, the original dataset must be of a certain size to achieve a high degree of prediction accuracy. Model learning with the dataset generated by mixup is equivalent to preventing overfitting by regularizing model learning with the original dataset (Wu et al., 2020). Therefore, if the original dataset is excessively small, the prediction accuracy of the trained model remains low, even if the dataset is augmented by a mixup.

Further, the model trained by our method is significantly influenced by outliers. Similar to ordinary linear regression, deep learning is affected by outliers. In addition, owing to the nature of the mixup, one outlier is spread across hypothetical observations, making the effect of outliers even more impactful.

To perform a mixup, augmented variables are limited to those that can be composited and scaled. For example, indicator variables such as interest rates are not subject to the mixup because they are difficult to compose and scale directly.

In applying the mixup proposed in this study, the prior knowledge that "for a virtual region obtained by linear interpolation, the feature vectors should lead to an associated target" needs to be established. If this prior knowledge is not satisfied, the mixup cannot be performed. For example, learning a production function that does not have a constant return to scale through the mixup is difficult. Let y_i be the output of region i and $\boldsymbol{x}_i$ be the production factor vector. If the production function g is not a constant return to scale, the prior knowledge required for mixup is not satisfied, which can be expressed as follows:

$$\lambda_1 y_1 + \lambda_2 y_2 \neq g(\lambda_1 \boldsymbol{x}_1 + \lambda_2 \boldsymbol{x}_2).$$

As mentioned previously, the mixup between a prefecture and its city is also incorrect. For these regions, obtaining a sum of values that includes transfers is infeasible; therefore, the composition cannot be meaningful. The same holds for regions between different points in time, and if the composition is performed, the treatment of transfers between regions must be considered.

By addressing these limitations, the methods in this study could be applied to forecasting a wider range of economic data.

References

Abbasimehr, H., Shabani, M., & Mohsen, Y. (2020). An optimized model using LSTM network for demand forecasting. *Computers & Industrial Engineering*, *143*, 106435, https://doi.org/10.1016/j.cie.2020.106435.

Bacharach, M. (1970). *Biproportional matrices & input–output change*. Cambridge University Press.

Bonfiglio, A., & Chelli, F. (2008). Assessing the behaviour of non-survey methods for constructing regional input–output tables through a Monte Carlo simulation. *Economic Systems Research*, *20*(3), 243–258, https://doi.org/10.1080/09535310802344315.

Botev, A., Lever, G., & Barber, D. (2017). Nesterov's accelerated gradient and momentum as approximations to regularised update descent. *2017 International Joint Conference on Neural Networks* (pp. 1899–1903), https://doi.org/10.1109/IJCNN.2017.7966082.

Chapelle, O., Weston, J., Bottou, L., & Vapnik, V. (2000). Vicinal risk minimization. T. Leen, T. Dietterich, & V. Tresp (Eds.), *Advances in Neural Information Processing Systems* (Vol. 13, pp. 416–422), MIT Press, https://doi.org/10.5555/3008751.3008809.

Dao, T., Gu, A., Ratner, A.J., Smith V., De Sa, C., & Ré, C. (2019). A kernel theory of modern data augmentation. *Proceedings of Machine Learning Research* (pp. 1528–1537).

Flegg, A.T., Lamonica, G.R., Chelli, F.M., Recchioni, M.C., & Tohmo, T. (2021). A new approach to modelling the input–output structure of regional economies using non-survey methods. *Journal of Economic Structures*, *10*(12), https://doi.org/10.1186/s40008-021-00242-8.

Flegg, A.T., & Tohmo, T. (2013). A comment on Tobias Kronenberg's "Construction of regional input–output tables using nonsurvey methods: the role of cross-hauling". *International Regional Science Review*, *36*(2), 235–257, https://doi.org/10.1177/0160017612446371.

Flegg, A.T., & Tohmo, T. (2016). Estimating regional input coefficients and multipliers: the use of FLQ is not a gamble. *Regional Studies*, *50*(2), 310–325, https://doi.org/10.1080/00343404.2014.901499.

Fujimoto, T. (2019). Appropriate assumption on cross-hauling national input–output table regionalization. *Spatial Economic Analysis*, *14*(1), 106–128, https://doi.org/10.1080/17421772.2018.1506151.

Gerking, S.D. (1976). *Estimation of stochastic input–output models: some statistical*

problems. Springer.

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: surpassing human–level performance on ImageNet classification. *2015 IEEE International Conference on Computer Vision* (pp. 1026–1034), https://doi.org/10.1109/ICCV.2015.123.

Hewings, G.J.D. (1977). Evaluating the possibilities for exchanging regional input–output coefficients. *Environment and Planning A: Economy and Space*, *9*(8), 927–944, https://doi.org/10.1068/a090927.

Hiramatsu, T., Inoue, H., & Kato, Y. (2016). Estimation of interregional input–output table using hybrid algorithm of the RAS method and real-coded genetic algorithm. *Transportation Research Part E: Logistics and Transportation Review*, *95*, 385–402, https://doi.org/10.1016/j.tre.2016.07.007.

Holỳ, V., & Šafr, K. (2023). Disaggregating input–output tables by the multidimensional RAS method: a case study of the Czech Republic. *Economic Systems Research*, *35*(1), 95–117, https://doi.org/10.1080/09535314.2022.2091978.

Hosoe, N. (2014). Estimation errors in input–output tables and prediction errors in computable general equilibrium analysis. *Economic Modelling*, *42*, 277–286, https://doi.org/10.1016/j.econmod.2014.07.012.

Isserman, A.M. (1977). The location quotient approach to estimating regional economic impacts. *Journal of the American Institute of Planners*, *43*(1), 33–41, https://doi.org/10.1080/01944367708977758.

Junius, T., & Oosterhaven, J. (2003). The solution of updating or regionalizing a matrix with both positive and negative entries. *Economic Systems Research*, *15*(1), 87–96, https://doi.org/10.1080/0953531032000056954.

Lahr, M., & De Mesnard, L. (2004). Biproportional techniques in input–output analysis: table updating and structural analysis. *Economic Systems Research*, *16*(2), 115–134, https://doi.org/10.1080/0953531042000219259.

Lamonica, G.R., & Chelli, F.M. (2018). The performance of non-survey techniques for constructing sub-territorial input–output tables. *Papers in Regional Science*, *97*(4), 1169–1202, https://doi.org/10.1111/pirs.12297.

Law, R., Li, G., Fong, D.K.C., & Han, X. (2019). Tourism demand forecasting: a deep learning approach. *Annals of Tourism Research*, *75*, 410–423, https://doi.org/10.1016/j.annals.2019.01.014.

Lemelin, A. (2009). A GRAS variant solving for minimum information loss. *Economic Systems Research*, *21*(4), 399–408, https://doi.org/10.1080/09535311003589310.

Lenzen, M., Moran, D.D., Geschke, A., & Kanemoto, K. (2014). A non-sign-preserving RAS variant. *Economic Systems Research*, *26*(2), 197–208, https://doi.org/10.1080/09535314.2014.897933.

Lenzen, M., Wood, R., & Gallego, B. (2007). Some comments on the GRAS method. *Economic Systems Research*, *19*(4), 461–465, https://doi.org/10.1080/09535310701698613.

Morrissey, K. (2016). A location quotient approach to producing regional production multipliers for the Irish economy. *Papers in Regional Science*, *95*(3), 491–506, https://doi.org/10.1111/pirs.12143.

Papadas, C.T., & Hutchinson, W.G. (2002). Neural network forecasts of input–output technology. *Applied Economics*, *34*(13), 1607–1615, https://doi.org/10.1080/00036840110118133.

Ramyar, S., & Kianfar, F. (2019). Forecasting crude oil prices: a comparison between artificial neural networks and vector autoregressive models. *Computational Economics*, *53*(2), 743–761, https://doi.org/10.1007/s10614-017-9764-7.

Richardson, H.W. (1985). Input–output and economic base multipliers: looking backward and forward. *Journal of Regional Science*, *25*(4), 607–661, https://doi.org/10.1111/j.1467-9787.1985.tb00325.x.

Riddington, G., Gibson, H., & Anderson, J. (2006). Comparison of gravity model, survey and location quotient-based local area tables and multipliers. *Regional Studies*, *40*(9), 1069–1081, https://doi.org/10.1080/00343400601047374.

Round, J.I. (1983). Nonsurvey techniques: a critical review of the theory and the evidence. *International Regional Science Review*, *8*(3), 189–212, https://doi.org/10.1177/016001768300800302.

Schaffer, W.A., & Chu, K. (1969). Nonsurvey techniques for constructing regional interindustry models. *Papers of the Regional Science Association*, *23*, 83–101, https://doi.org/10.1007/BF01941876.

Smith, L.N. (2017). Cyclical learning rates for training neural networks. *2017 IEEE Winter Conference on Applications of Computer Vision* (pp. 464–472), https://doi.org/10.1109/WACV.2017.58.

Szabó, N. (2015). Methods for regionalizing input–output tables. *Regional Statistics*, *5*(1), 44–65.

Temursho, U., Oosterhaven, J., & Cardenete, M.A. (2021). A multi-regional generalized RAS updating technique. *Spatial Economic Analysis*, *16*(3), 271–286, https://doi.org/10.1080/17421772.2020.1825782.

Wu, S., Zhang, H., Valiant, G., & Ré, C. (2020). On the generalization effects of linear transformations in data augmentation. *International Conference on Machine Learning* (pp. 10410–10420), https://doi.org/10.5555/3524938.3525902.

Zhang, H., Cisse, M., Dauphin, Y.N., & Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*.

Zhang, L., Deng, Z., Kawaguchi, K., Ghorbani, A., & Zou, J. (2021). How does mixup help with robustness and generalization? *International Conference on Learning Representations*.

Acknowledgments

The author is grateful to Dr. Shinya Kato (Yamaguchi University) for suggestions for confirming the prediction accuracy of the method used in this study.

We would like to thank Editage (www.editage.com) for English language editing.

This version of the article has been accepted for publication, after peer review but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: https://doi.org/10.1007/s10614-024-10641-1. Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms.

Funding

The author declares that no funds, grants, or other support were received during the preparation of this manuscript.

Ethics declarations

Conflict of interest

The author has no relevant financial or non-financial interests to disclose.

Appendix A On the sum of intermediate inputs and gross outputs between regions

In this appendix, we show for intermediate inputs and gross outputs that the sum of the values for two regions is equal to the value when these regions are considered as one. [13]

[13] This explanation was originally presented in a Japanese article authored in 2021. I translated it into English and restated it.

Let us consider the creation of a single region $R12$, which is the composition of any two regions (denoted as $R1$ and $R2$). The intermediate input of $R12$ is equal to the sum of the intermediate inputs of $R1$ and $R2$ for a regional input–output table of the competitive import type.

The intermediate input in a regional input–output table of the competitive import type comprises the aggregate of inputs that are exchanged between industries within the region, the inputs received from other regions within the country, and the inputs imported from abroad by industries within the region (Fujimoto, 2019). That is, for the intermediate input $m_{i,j}$ from industry i to industry j in a region, if $\hat{m}_{i,j}$ is the input within the region, $\dot{m}_{i,j}$ is the input from other regions of the country to industry j in the region, and $\tilde{m}_{i,j}$ is the import from abroad to industry j in the region,

$$m_{i,j} = \hat{m}_{i,j} + \dot{m}_{i,j} + \tilde{m}_{i,j}.$$

The intermediate inputs in $R1$ are denoted as $(m_{i,j}^{R1}, \hat{m}_{i,j}^{R1}, \dot{m}_{i,j}^{R1}, \tilde{m}_{i,j}^{R1})$. Similarly, those in $R2$ are denoted as $(m_{i,j}^{R2}, \hat{m}_{i,j}^{R2}, \dot{m}_{i,j}^{R2}, \tilde{m}_{i,j}^{R2})$. Adding the intermediate inputs of these two regions yields

$$m_{i,j}^{R1} + m_{i,j}^{R2} = \hat{m}_{i,j}^{R1} + \dot{m}_{i,j}^{R1} + \tilde{m}_{i,j}^{R1} + \hat{m}_{i,j}^{R2} + \dot{m}_{i,j}^{R2} + \tilde{m}_{i,j}^{R2}.$$

The inputs within $R12(\hat{m}_{i,j}^{R12})$, inputs from other regions of the country to $R12(\dot{m}_{i,j}^{R12})$, and foreign inputs to $R12(\tilde{m}_{i,j}^{R12})$ are

$$\begin{aligned}
\hat{m}_{i,j}^{R12} &= \hat{m}_{i,j}^{R1} + \hat{m}_{i,j}^{R2} + (\text{input from } R2 \text{ out of } \dot{m}_{i,j}^{R1}) \\
&\quad + (\text{input from } R1 \text{ out of } \dot{m}_{i,j}^{R2}) \\
\dot{m}_{i,j}^{R12} &= (\text{input from all other regions of the country except } R2 \text{ out of } \dot{m}_{i,j}^{R1}) \\
&\quad + (\text{input from all other regions of the country except } R1 \text{ out of } \dot{m}_{i,j}^{R2}) \\
\tilde{m}_{i,j}^{R12} &= \tilde{m}_{i,j}^{R1} + \tilde{m}_{i,j}^{R2}.
\end{aligned}$$

The intermediate input $m_{i,j}^{R12}$ of $R12$ is the sum of $\hat{m}_{i,j}^{R12}$, $\dot{m}_{i,j}^{R12}$, and $\tilde{m}_{i,j}^{R12}$. Thus,

$$\begin{aligned}
m_{i,j}^{R12} &= \hat{m}_{i,j}^{R12} + \dot{m}_{i,j}^{R12} + \tilde{m}_{i,j}^{R12} \\
&= \hat{m}_{i,j}^{R1} + \hat{m}_{i,j}^{R2} \\
&\quad + (\text{input from } R2 \text{ out of } \dot{m}_{i,j}^{R1}) \\
&\quad + (\text{input from } R1 \text{ out of } \dot{m}_{i,j}^{R2}) \\
&\quad + (\text{input from all other regions of the country except } R2 \text{ out of } \dot{m}_{i,j}^{R1}) \\
&\quad + (\text{input from all other regions of the country except } R1 \text{ out of } \dot{m}_{i,j}^{R2}) \\
&\quad + \tilde{m}_{i,j}^{R1} + \tilde{m}_{i,j}^{R2} \\
&= \hat{m}_{i,j}^{R1} + \hat{m}_{i,j}^{R2} + \dot{m}_{i,j}^{R1} + \dot{m}_{i,j}^{R2} + \tilde{m}_{i,j}^{R1} + \tilde{m}_{i,j}^{R2} \\
&= m_{i,j}^{R1} + m_{i,j}^{R2}.
\end{aligned}$$

Therefore, the sum of the intermediate inputs for any two regions is equal to the intermediate input when these two regions are aggregated and considered as a new region.

We confirm that the gross output Y_i^{R12} of industry i in $R12$ is equal to the sum of the gross output of $R1$ and $R2$. In the input–output table of the competitive import type, in addition to intermediate inputs, the final demand for $R12$ can also be calculated as the sum of the final demand for $R1$ and $R2$. As the exports in $R12$ are equal to the sum of exports in $R1$ and $R2$ and the imports in $R12$ are similar, the net exports (the differences between the exports and imports) in $R12$ are equal to the sum of the net exports in $R1$ and $R2$. Shipping to the other regions of industry i in $R12$, L_i^{R12}, is obtained as follows:

$$\begin{aligned} L_i^{R12} = L_i^{R1} + L_i^{R2} &- \sum_j (\text{input from } R1 \text{ out of } \dot{m}_{i,j}^{R2}) \\ &- (\text{input from } R1 \text{ out of } F_i^{R2}) \\ &- \sum_j (\text{input from } R2 \text{ out of } \dot{m}_{i,j}^{R1}) \\ &- (\text{input from } R2 \text{ out of } F_i^{R1}), \end{aligned}$$

where F_i^{R1} is the final demand for industry i in $R1$ and F_i^{R2} is the final demand for industry i in $R2$. Receiving from the other regions of industry i in $R12$, N_i^{R12}, is calculated in the same manner as L_i^{R12}. Thus,

$$\begin{aligned} N_i^{R12} = N_i^{R1} + N_i^{R2} &- \sum_j (\text{input from } R2 \text{ out of } \dot{m}_{i,j}^{R1}) \\ &- (\text{input from } R2 \text{ out of } F_i^{R1}) \\ &- \sum_j (\text{input from } R1 \text{ out of } \dot{m}_{i,j}^{R2}) \\ &- (\text{input from } R1 \text{ out of } F_i^{R2}). \end{aligned}$$

From these equations, the net transfer of industry i in $R12$ is equal to the sum of net transfers in $R1$ and $R2$, as follows:

$$L_i^{R12} - N_i^{R12} = L_i^{R1} + L_i^{R2} - N_i^{R1} - N_i^{R2}.$$

The gross output is the sum of the total intermediate demand (= the row sum of intermediate inputs), final demand, net transfers, and net exports. Thus, the output Y_i^{R12} in $R12$ is calculated as $Y_i^{R1} + Y_i^{R2}$.