Neurocomputing 172 (2016) 399–404

Contents lists available at ScienceDirect

Neurocomputing

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

Improved neural network tomography by initial learning with coarse reconstructed image

Masaru Teranishi

Department of Information Systems and Management, Hiroshima Institute of Technology, Hiroshima, Japan

ARTICLE INFO

Article history:
Received 15 September 2014
Received in revised form
5 February 2015
Accepted 18 March 2015
Available online 21 August 2015

Keywords:
Few view tomography
Neural network
Back propagation
Inverse problem
Ill-posed problem
Model fitting

ABSTRACT

This paper presents a stable learning method of the neural network tomography, in case of asymmetrical few view projection. The neural network collocation method (NNCM) is one of effective reconstruction tools for symmetrical few view tomography. But in cases of asymmetrical few view, the learning process of NNCM tends to be unstable and fails to reconstruct appropriate tomographic images. We solve the unstable learning problem of NNCM by introducing a coarse reconstructed image in the initial learning stage of NNCM. The numerical simulation with an assumed tomographic image shows the effectiveness of the proposed method.

1. Introduction

The Computerized Tomography (CT) is one of effective remote sensed imaging method for medical and industrial non-destructive diagnostics [1]. CT is also applied to fusion plasma imaging measurements, thanks to its remote sensing facility.

But most fusion experimental devices have few observation port due to the vacuum vessel structure. Then the viewing angles for tomographic measurement are limited to few and asymmetrical directions. For such asymmetrical few view, standard CT reconstruction methods are difficult to use.

For the few view CT problem, there are mainly two kinds of reconstruction approaches: regularization methods for ill-posed equation and model fitting methods. Ma et al. and Takeda and Ma proposed a new model fitting CT reconstruction method by using a neural network [2,3] which is called "Neural Network Collocation Method" (NNCM). The NNCM reconstructs tomographic image by the error back propagation (BP) learning algorithm based on the projection data.

Actually, the effectiveness of NNCM is reported for few view CT problem [2], but in these literature, views are set in symmetric directions. But in actual cases of plasma CT imaging, the view directions are set in "asymmetrical" angles. NNCM has drawbacks for asymmetrical few view CT with instability of learning and often fails to reconstruct appropriate tomographic image.

The paper proposes a stable learning method for NNCM in asymmetrical few view CT. The proposed method stabilizes the reconstruction process by using a coarse reconstructed image as the teacher data for BP algorithm directly, instead of the projection data, in the initial learning stage.

2. Few view CT problem

This paper deals with two-dimensional tomographic imaging. The goal of CT is to reconstruct a spatial distribution of certain physical quantity $f(x,y)$ s at spatial position (x,y) s of a cross section of the object. A tomographic image of the cross section is represented by arranging $f(x,y)$ s for whole of the section. We can measure certain set of unknown $f(x,y)$ values indirectly by scanning with transparent beams like X-rays. An output of ith beam detector g_i means a line integral of $f(x,y)$ along the ith beam path, as shown in Fig. 1.

If the reconstruction region is divided into n pixels, the calculation of g_i is approximated by discrete expression as follows:

$$g_i = \int_{L_i} f(x,y)\,dx\,dy \approx \sum_{j=1}^{n} a_{ij} f_j \tag{1}$$

where f_j is mean image intensity of jth pixel, a_{ij} is the traversing length of the ith beam through the jth pixel. In the following descriptions, we denote a tomographic image as a vector $\vec{f} = (f_1, f_2, \dots, f_j, \dots, f_n)^T$ for computational convenience.

In discrete form of the projection, reconstruction of an unknown tomographic image vector $\vec{f} = (f_1, f_2, \dots, f_j, \dots, f_n)^T$ from the measured projection data vector $\vec{g}_{meas} = (g_1, g_2, \dots, g_j, \dots, g_m)^T$

http://dx.doi.org/10.1016/j.neucom.2015.03.103
0925-2312/

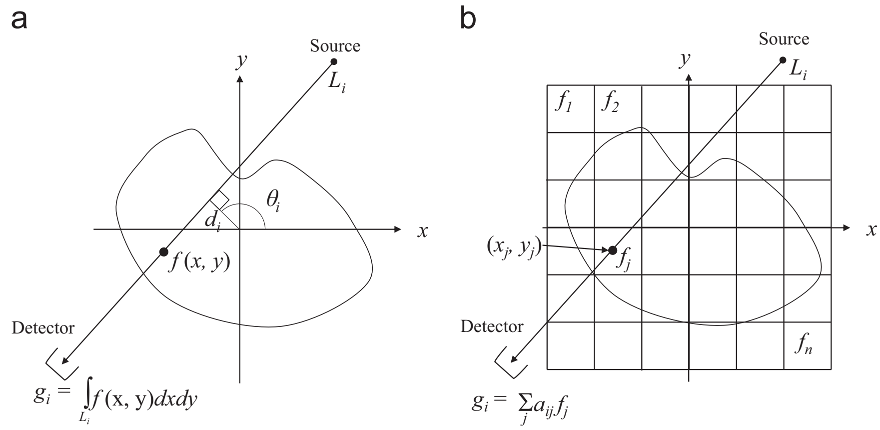

Fig. 1. Projection measurement of g_i along a beam path L_i: (a) continuous form, (b) discrete form.

is equivalent to solve a linear equation as

$$A\vec{f} = \vec{g}^{meas} \quad (2)$$

where matrix A is $m \times n$ dimensional and has a_{ij} of Eq. (1) as its element, and is called the "projection matrix". In general CT application, Eq. (2) is solvable because the number of projection data is more enough than the number of unknowns, i.e., the number of pixels of the reconstruction image ($m \geq n$). But in case of few view CT, the number of projection data is extremely less than the number of unknowns ($m \ll n$), so Eq. (2) becomes ill-posed.

There are two different approaches to solve such few view CT problems. The one is the model fitting methods like ART, SIRT [1], and some kind of series expansions [4], the other is the regularization methods [1]. The NNCM is one of model fitting approaches.

3. Neural network collocation method (NNCM)

Fig. 2 shows an overview of the NNCM.

The NNCM uses a Multi-Layered Perceptron (MLP) type neural network [5]. The MLP consists of three layers: input, hidden, and output layer. The input layer has two neuron units and receives a spatial position (x_j, y_j) of the reconstruction region. The output layer has a neuron unit and outputs the intensity $f^{MLP}(x_j, y_j)$ of the corresponding spatial position. NNCM reconstructs an image $\vec{f}^{MLP}$ by driving the MLP to compute intensities of the whole spatial positions of the reconstruction region.

In the viewpoint of model fitting, The NNCM represents the reconstructed image as a synthesized basis functions of spatial positions. Each unit of the hidden layer represents the model basis function of the reconstructed image. In the hidden layer, the kth unit computes its internal state u_k^h as a sum of weighted spatial position (x_j, y_j) as

$$u_k^h = w_{k,x}^h x_j + w_{k,y}^h y_j + w_{k,b}^h \quad (3)$$

where $w_{k,x}^h$ is the connection weight from an input layer's unit for position x_j, $w_{k,y}^h$ denotes the similar way for y_j, and $w_{k,b}^h$ does for the bias unit. The output of kth unit of spatial position (x_j, y_j) is computed as o_k^h by transforming u_k^h with sigmoid function $\sigma(u) = 1/(1+e^{-u})$ as

$$o_k^h = \sigma(u_k^h) = \sigma(w_{k,x}^h x_j + w_{k,y}^h y_j + w_{k,b}^h). \quad (4)$$

The unit of the output layer computes its internal state as the weighted sum of basis functions of the hidden layer's units and transform the state into the output by sigmoid function. The MLP output $f^{MLP}(\vec{w}, x_j, y_j)$ is determined by current input (x_j, y_j) for the jth pixel and weight vector $\vec{w}$.

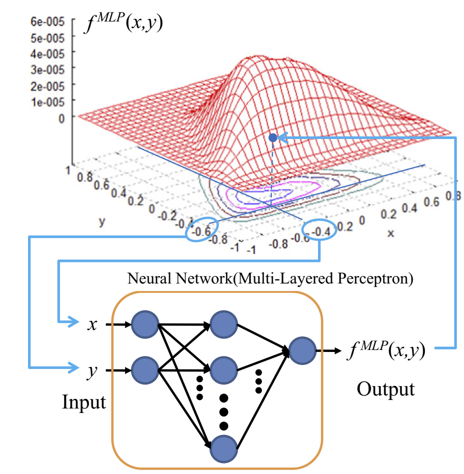

Fig. 2. Usage of the NNCM.

In the NNCM, a MLP reconstructs an appropriate tomographic image for a specific projected data. The reconstruction is achieved by tuning the connection weights using the error back propagation (BP) learning algorithm based on the projection data. The learning result is effective only for one specific projected data. In this meaning, the MLP of the NNCM has no generalized facility for input data. If we want to reconstruct an image for another projection data, then we should reset the MLP connections and let the MLP learn from the beginning. The goal of CT reconstruction is to fit the projection data of the reconstructed image to the measured one, i.e., to reconstruct a tomographic image by minimizing residual of projection $\|A\vec{f}^{MLP} - \vec{g}^{meas}\|$. On the other hand, the error function used in the BP algorithm is computed with the MLP output and the corresponding teacher data. In this way, we have defined the error function of BP for NNCM as

$$E = E^{img} = \frac{1}{2}\sum_{j=1}^{N}\left\{f^{MLP}(\vec{w}, x_j, y_j) - f^{TRUE}(x_j, y_j)\right\}^2 \quad (5)$$

and the BP algorithm modifies the MLP's weights by gradient descent to minimize E as

$$\vec{w}^{(t+1)} = \vec{w}^{(t)} + \Delta\vec{w}^{(t)}$$
$$\Delta\vec{w}^{(t)} = -\eta\frac{\partial E}{\partial \vec{w}^{(t)}} + \beta\Delta\vec{w}^{(t-1)} \quad (6)$$

where t is an index of the current learning iteration, η is the learning step size, β is the momentum coefficient of the past learning.