The application of feed-forward neural network for the X-ray image fusion

ZhangJian, WangXueWu
Department of Engineering Physics, Tsinghua University, Beijing, P. R. China
zjmagicworld@gmail.com, wangxuewu@tsinghua.edu.cn

Abstract. Under the current urgent circumstances of the aviation security, all countries are intensifying the security inspection. In the view of the specialty of dangerous goods, simulating the X-ray images of the superimposed objects to train inspectors, would become a convenient and effective way to strengthen the inspectors’ priori knowledge of threatening ones. In this paper, depending on the learning algorithm ‘OWO-HWO’, we design a three layers feed-forward neural network. The characteristics and advantages of the neural network on the field of X-ray image fusion are completely analyzed in the paper.

1. Introduction

Under the current urgent circumstances of the aviation security, all countries are gradually intensifying the security inspection and there have been amounts of security equipments used in the airfields. The judgments of inspectors primarily depend on the priori knowledge of the corresponding dangerous goods under the X-ray. How to strengthen the inspectors’ priori knowledge of threatening ones becomes the key factor of intensifying security inspection.

As well as known, training relying on real X-ray images of dangerous objects is the only way of improving the inspectors’ priori knowledge. We couldn’t deny the fact that collecting dangerous objects, such as drugs or guns is a tough work. Therefore we couldn’t conveniently update the X-ray images database. Simulating for X-ray images of the dangerous objects overlapped with other daily objects becomes an effective way of solving the contradiction.

In recent years, the general method of fusing X-ray images applied in the industrial field primarily relies on the attenuation law of x-ray intensity in material[1]. By interpolating algorithm, the color index and gray value on every fusing pixel are calculated. The attenuation law of X-ray in material could be written as

\[ I_1 = I_0 e^{-\rho m l_m} \]  \hspace{1cm} (1.1)

where \( I_0, I_1 \) are the intensities of X-ray before and after X-ray penetrates the material, respectively. \( \rho_m \) is the mass attenuation coefficient, and \( l_m \) is the mass thickness.

In this paper, we primarily adopt the feed-forward neural network to achieve the fusion of X-ray images. The technology of artificial neural network has been widely used in all kinds of fields, such as approximating function[2], pattern recognition[3]. Ken-Ichi F[4] has proved that any continuous

1 Project supported by the National Natural Science Foundation of China (Grant No. 10705018).
mapping can be approximately realized by the three layers neural network. According to this theory, we could use the neural network to simulate any unknown complex system under arbitrary-precision. On the aspects of learning algorithm, there have been amounts of research progresses. Martin F M [5] explores the scaled conjugate gradient algorithm. Lera G and Pinzolas M [6] design the Levenberg-Marquart (LM) algorithm. Maniezzo V [7] brings the genetic algorithm into the field of neural network, which could give a global optimization solution. In addition, there are also back-propagation[8], and steepest descent and Newton's method [9].

In this paper, depending on the learning algorithm OWO-HWO[10], we build a three layers neural network, which has 30 units in the hidden layer. In section 2, we describe the primary configuration of feed-forward network and explore the characters of OWO-HWO learning algorithm. In section 3, we make the comparison between traditional algorithm and neural network model. Finally, the section 4 concludes this paper.

2. Feed-forward neural network

In this section, we will describe the structure of feed-forward neural network and the characters of OWO-HWO learning algorithm.

![Fig. 1 The X-ray image of phone and the components of one pixel](image1)

![Fig. 2 The structure of feed-forward neural network](image2)

2.1. Structure of feed-forward neural network

According to the theory of Ken-Ichi F[4], we adopt the three layers feed-forward neural network. In an X-ray image collected on the security equipment, a pixel consists of color index(CIndex) and gray value(Gray) (Fig.1). In the input layer, the data includes the color index and gray value of the corresponding pixels. In the output layer, the data includes the ones of the corresponding pixel after fusing. The whole structure of feed-forward neural network is depicted in Fig.2.

Every hidden unit adopts the sigmoid activation function. We use the set\(\{(X_p, T_p)\}\)to represents the training patterns, where \(X_p = \{C_{pA}, G_{pA}, C_{pB}, G_{pB}\}\), \(C_{pA}, G_{pA}\) and \(C_{pB}, G_{pB}\) denote the color index and gray value on the A and B pixel for the \(p\)th sample, respectively; \(T_p = \{C_p, G_p\}\), \(C_p, G_p\) represent desired output values in the corresponding pixel for the \(p\)th sample. Through detecting the difference between \(T_p\) and \(Y_p\) (the value calculated by the neural network), the learning algorithm automatically adjusts the parameters of the neural network to reduce the
MSE (Mean Square Error) between the simulating value and real value.

2.2. Brief description of the OWO-HWO [9]

The whole algorithm is divided into two parts, that one process is OWO, which optimizes the output weights, and the other one is HWO, which optimizes the hidden weights.

For the \( k \)th hidden unit, the net input \( \text{net}_{pk} \) and the output activation \( O_{pk} \) for the \( p \)th training pattern are

\[
\text{net}_{pk} = \sum_{n=1}^{N+1} W_{hi}(k,n)x_{pn} \tag{2.1}
\]

\[
O_{pk} = f(\text{net}_{pk}) \tag{2.2}
\]

Where \( x_{pn} \) denotes the \( n \)th element of \( X_p \) and \( W_{hi}(k,n) \) denotes the weight connecting the \( n \)th input unit to the \( k \)th hidden unit. The activation function \( f \) is sigmoidal

\[
f(x) = \frac{1}{1 + e^{-x}} \tag{2.3}
\]

For the learning process of the feed-forward network, the overall performance could be measured by MSE, which can be written as

\[
E = \frac{1}{N_t} \sum_{p=1}^{N_t} \sum_{i=1}^{M} [t_{pi} - y_{pi}]^2 \tag{2.4}
\]

(1) Output weight optimization

In the output optimization process, for finding the optimum weights \( W_o \), we could solve the linear equations, which derive when the gradients of \( E \) with respect to the output weights are set to zero. These equations are

\[
\sum_{n=1}^{L} W_o(i,n)R(n,m) = C(i,m) \tag{2.5}
\]

Where \( C \) is the cross correlation matrix with elements:

\[
C(i,m) = \frac{1}{N_t} \sum_{p=1}^{N_t} t_{pl} \cdot \tilde{x}_{pm} \tag{2.6}
\]

\( R \) is the auto correlation matrix:

\[
R(n,m) = \frac{1}{N_t} \sum_{p=1}^{N_t} \tilde{x}_{pm} \cdot \tilde{x}_{pm} \tag{2.7}
\]

Where \( \tilde{x}_{pm} \) is the \( m \)th element of \( \tilde{X}_p \) and \( \tilde{X}_p \) is an augmented vector of \( X_p \).

(2) Hidden weight optimization

There exists the uniform monotonicity between \( E \) and \( E_\delta(k) \), where \( E_\delta(k) \) is one simplified form of \( E \). Minimizing \( E \) is equivalent to minimizing \( E_\delta(k) \). We could set the gradients of the \( E_\delta(k) \) with respect to the hidden weight change \( e(k,n) \) to zero. Thus, we could calculate the autocorrelation matrix \( R \) and cross correlation matrix \( C_\delta \).

\[
R(n,m) = \frac{1}{N_t} \sum_{p=1}^{N_t} (x_{pm} \cdot \Delta x_{pm}) \cdot (f_{pk})^2 \tag{2.8}
\]

\[
C_\delta(k,m) = \frac{1}{N_t} \sum_{p=1}^{N_t} [\Delta \text{net}_{pk}^* \cdot x_{pm}] \cdot (f_{pk})^2 \tag{2.9}
\]

Where \( \Delta \text{net}_{pk}^* \) is the difference between current value \( \text{net}_{pk} \) and an optimum value \( \text{net}_{pk}^* \); \( f_{pk} \) is shorthand for the activation derivative \( f'(\text{net}_{pk}) \).

So we have

\[
\sum_{n=1}^{N_t+1} e(k,n)R(n,m) = C_\delta(k,m) \tag{2.10}
\]
After calculating the $e(k, n)$ we could update the hidden weights using the following formula:

$$W_{hi}(k, n) \leftarrow W_{hi}(k, n) + Z \cdot e(k, n)$$

(2.11)

where $Z$ is the learning rate.

2.3. Performance analysis of OWO-HWO

In this section, based on the training patterns, we will analyze the performance of the algorithm on the aspect of MSE.

In the following analysis, we could observe the changes of MSE versus the changes of the number of hidden units. The three-layer network has 4 inputs and 2 outputs. In the result (Fig.3), we could clearly notice that there is a quick descent trend in the MSE of the training patterns. But as the number of hidden units increases, the rate of MSE descent becomes more and more slow.

![Fig. 3 Performance analysis of OWO-HWO](image)

![Fig.4 Correlation degree analysis of neural network](image)

To some extents, if the correlation degree among the hidden units is too large, there exist too much unuseful units in the hidden layer, which could critically influence the performance of the whole system. The correlation degree could be defined as

$$CD = \sum_{j=1}^{W_h} \langle \bar{m}_j, \bar{m}_{j+1} \rangle$$

(2.9)

where $\bar{m}_j$ is the orthonormal result of $m_j$ and $m_j = \{O_{1j}, O_{2j}, ..., O_{pj}\}$, $O_{pj} (j = 1, 2, ..., N_h)$ denotes the output of the $i$th hidden unit for the $p$th sample. Due to orthogonalization in the order of hidden units by Gram-Schmidt procedure, the failure of this procedure means that the new adding vector $m_i$ is linearly dependent on hidden units that have already been orthognormalized. That is to say, the hidden unit is unuseful. Based on this principle, the correlation degree defined in (2.9) will suddenly increase in some point. This point will become the optimum number of hidden units. Fig.4 displays the change of correlation degree versus the number of hidden units. From the result, we could find that, when the number of hidden units is over 30, the correlation degree increases suddenly. In this paper, we adopt 30 hidden units.

3. Experiments

Data set: the data set contains two groups of X-ray images. In every group, there are 3 images. The
first two images are the ones which prepare to fuse, and the third image is the one which is collected under the real condition of overlapping the corresponding two objects. The fusion model is displayed in the table 1.

In the table 1, it also displays the comparison of fusion effect depending on the traditional method and neural network method. Obviously, the fusion effect gained by neural network method is much closer to the real situation. Under the traditional method, there are some obvious discrepancies between the fusion effect and the real effect, such as the A and B region in the group 1 and the A region in the group 2. We could easily see the A region of fusion effect gained by traditional method bias against green. However, the primary hue of the A region of real effect is blue. In addition, the color of B region of fusion effect by traditional method is deeper than the same region of real effect. The same problem also exists in the group 2. Under the traditional method, in the group 1, the MSE of color index and gray value is $E_{CT} = 3.4$, $E_{GT} = 4633$, respectively. However, under the neural network method, the MSE is $E_{CNN} = 2.0$, $E_{GNN} = 3337$, respectively. In the group 2, the MSE gained by neural network method is also smaller than the traditional method. From the statistical data, we also could see that the neural network method could generate good fusion effect.

| Table 1 Comparison between traditional method and neural network method |
|---|---|---|---|
| Fusion Model | Real Effect | Fusion Effect (Traditional Method) | Fusion Effect (Neural Network Method) |
| 1 | ![Image of group 1] | ![Image of group 1 fusion effect (traditional)] $E_{CT} = 3.4$ $E_{GT} = 4633$ | ![Image of group 1 fusion effect (neural network)] $E_{CNN} = 2.0$ $E_{GNN} = 3337$ |
| 2 | ![Image of group 2] | ![Image of group 2 fusion effect (traditional)] $E_{CT} = 2.5$ $E_{GT} = 4159$ | ![Image of group 2 fusion effect (neural network)] $E_{CNN} = 1.8$ $E_{GNN} = 2833$ |

4. Conclusions

From the study about the application of artificial neural network on the fusion of X-ray images, we could easily see that the neural network possesses great advantages in simulating complex system and fitting nonlinear function. Considering on the statistical fluctuations of the real physical process $\text{Std}_{\text{color index}} = 0.5 \sim 1.3$ and $\text{Std}_{\text{gray}} = 350 \sim 450$, and the experimental error, the fusion precision of color index could fall into the acceptable range. However, for the gray value, there exist some
deviations between the fusion result and the real value. There is one possible reason that there is a big discrepancy in the complexity between the nonlinear relations conformed by color index and gray value. Therefore, it couldn’t use one single model to perfectly simulate the whole system at the same time, which also become the hot topic we further research in the future.

References:
[1] Wang, Li, Kang, et al. Material discrimination by high-energy X-ray dual-energy image. High Energy Physics and Nuclear Physics. 2007, 31(11) 1076-1081
[2] Ferrari S and Stengel R F. Smooth function approximation using neural networks. Neural Networks. 2005,16(1) 24-38
[3] Carpenter G A and Grossberg S. The ART of adaptive pattern recognition by a self-organizing neural network. Computer. 1988,21(3) 77-88
[4] Ken-Ichi F. On the approximate realization of continuous mappings by neural networks. Neural Networks.1989, 2(3) 183-192
[5] Martin F M. A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks. 1993, 6(4) 525-533
[6] Lera G and Pinzolas M. Neighborhood based Levenberg–Marquardt algorithm for neural network training. Neural Networks. 2002, 13(5) 1200-1203
[7] Maniezzo V. Genetic evolution of the topology and weight distribution of neural networks. Neural Networks. 1994, 5(1) 39-53
[8] Lehtokangas M. Modeling with constructive backpropagation. Neural Networks. 1999,12(4-5) 707-716
[9] Roberto B. First and Second Order Methods for Learning: Between Steepest Descent and Newton's Method. Neurocomputing. 1992, 4(2) 141-166
[10] Chang H Yu, et al. An efficient hidden layer training method for the multilayer perceptron Neurocomputing. 2006,70(1-3) 525-535