Engineering Applications of Artificial Intelligence 42 (2015) 57–66

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A novel algorithm of extended neural networks for image recognition ☆

Kankan Dai, Jianwei Zhao, Feilong Cao *

Department of Applied Mathematics, College of Sciences, China Jiliang University, Hangzhou 310018, Zhejiang Province, PR China

ARTICLE INFO

Article history:
Received 31 October 2014
Received in revised form
17 March 2015
Accepted 17 March 2015
Available online 10 April 2015

Keywords:
Feedforward neural networks
Matrix data
BP algorithm
Image recognition

ABSTRACT

As a class of important classifiers, feedforward neural networks (FNNs) have been used considerably in the study of pattern recognition. Since the inputs to FNNs are usually vectors, and many data are usually presented in the form of matrices, the matrices have to be decomposed into vectors before FNNs are employed. A drawback to this approach is that important information regarding correlations of elements within the original matrices are lost. Unlike traditional vector input based FNNs, a new algorithm of extended FNN with matrix inputs, called two-dimensional back-propagation (2D-BP), is proposed in this paper to classify matrix data directly, which utilizes the technique of incremental gradient descent to fully train the extended FNNs. These kinds of FNNs help to maintain the matrix structure of the 2D input features, which helps with image recognition. Promising experimental results of handwritten digits and face-image classification are provided to demonstrate the effectiveness of the proposed method.

1. Introduction

It is well-known that image recognition is a hot topic in the fields of machine learning and computer vision. In such recognition systems, many data images, such as handwritten digital images, face images, and palm images, are usually presented in the form of matrices. Clearly, determining how to classify these kinds of data is an important topic in pattern recognition.

Traditional image recognition system contains three steps: image pre-processing, feature extraction, and classification. At present, there are various methods for feature extraction, such as principal component analysis (PCA) (Jolliffe, 2002), independent component analysis (ICA) (Hyvärinen and Oja, 2000), and several popular applied classifiers, including K-nearest neighbourhoods (KNNs) (Shakhnarovich et al., 2008), support vector machines (SVMs) (Vapnik, 2000), feedforward neural networks (FNNs) (Hornik et al., 1989), and so on.

Nevertheless, these methods are usually based on vector inputs. Thus, when they are used in image processing, we first have to expand the matrix inputs into vector form. These types of transformation often lead to the loss of important information regarding the original matrix data, and thus affect the recognition process. On the other hand, expanding matrix inputs into vectors usually causes high dimensionality and increases the complexity of the used models.

To solve this problem, two-dimensional (2D) methods that directly operate on matrix data are proposed, for example, the two-dimensional principal component analysis (2DPCA) (Yang et al., 2004; Zhang and Zhou, 2005) and two-dimensional linear discriminant analysis (2DLDA) (Li and Yuan, 2005; Sanguansat et al., 2006; Yang et al., 2010), which were verified useful for extracting effective information about the inner structure of matrix data, as well as reducing the computational complexity of the extraction. It is natural to raise the question: For the existing vector-based classifiers, such as the SVM and FNN methods, can they be extended for matrix input?

Neural networks have played an important role in pattern recognition (Bishop, 1995; Lin et al., 1997; LeCun et al., 1998; Shang et al., 2006). In order to classify matrix data directly, and to preserve the matrix or 2D feature structure effectively, Lu et al. (2014) proposed a novel classifier, the two-dimensional neural network with random weights (2D-NNRW) method, and achieved good performance on face recognition. In fact, it is an extended 2D single hidden layer feed forward neural network (2D-SLFN) model that employs left and right projecting vectors to regress matrix inputs. Additionally, it uses the random idea to train the network, i.e. it randomly sets the left and right projecting vectors and the hidden biases, and then determines the output weights by solving a linear equation system. The results obtained in Lu et al. (2014) show that the use of 2D-SLFN improves face recognition accuracy.

The randomness in the NNRW algorithm can be understood in deeper detail when we consider the function approximation with Monte Carlo (MC) methods. It was shown in Igelnik and Pao (1995) that any continuous function defined on a compact set can be

☆ Supported by the National Natural Science Foundation of China (Nos. 61272023 and 91330118) and Zhejiang Provincial Natural Science Foundation of China (No. LY14A010027).

* Corresponding author.
E-mail address: feilongcao@gmail.com (F. Cao).

http://dx.doi.org/10.1016/j.engappai.2015.03.010
0952-1976/

represented by a limit-integral of a multivariate continuous function that is integrated in parameter space. Although using the NNRW algorithm can simplify the learning steps taken by SLFNs, the following issues still remain in both the NNRW and 2D-NNRW methods:

- The number of hidden nodes should be sufficiently large and supervised initialization is needed in order to model and compensate for the system's uncertainties.
- There exists an over-fitting phenomenon caused by many additional hidden nodes in the NNRW method due to the MC approximating approach.
- There is a predictive instability caused by randomly assignment of nonlinear weights, and the way of learning SLFNs with NNRW methods using singular value decomposition (SVD) usually produces large magnitude linear weights, which makes the networks highly sensitive to new data.

To overcome these issues, we have attempted to fully train the network with the aim of implementing incremental gradient based learning for 2D-SLFN. A learning algorithm called two-dimensional back-propagation (2D-BP) is proposed, where a momentum modification is added to improve convergence. A series of comparative studies of handwritten digits and face-image classification were carried out. The results of the testing datasets are promising, and support a positive statement regarding their performances among the 1D-BP and 2D-NNRW methods.

The rest of this paper is organized as follows. FNNs models and their corresponding training algorithms are reviewed in Section 2. A detailed description of our method is given in Section 3. An evaluation of the performance of our algorithm, the handwritten digits and face datasets employed in the experiments, and our results, which include comparisons and discussions, are presented in Section 4. Conclusions are presented in the final section.

2. FNN models and learning

2.1. Single hidden-layer feedforward neural networks

Generally, a SLFN is described as follows:

$$f(x) = \sum_{k=1}^{L} \beta_k g(w_k^\top x + b_k) + \alpha, \tag{1}$$

where x is the input pattern vector, $w_k = [w_{k1}, w_{k2}, \dots, w_{kd}]^\top$, b_k are input layer weights and biases, respectively, and $\beta_k = [\beta_{k1}, \beta_{k2}, \dots, \beta_{ko}]^\top$ and α are the output layer weights and biases, respectively.

It is true that SLFNs are universal approximators (Hornik et al., 1989; Cybenko, 1989; Barron, 1993), even when the hidden-layer weights and bias are randomly assigned. In the case that weights and bias are randomly assigned, i.e., the input layer weights w_k and biases b_k as defined on a probabilistic space $S_p(\Omega, P)$, where (Ω, P) should be determined in the learning stage, then we just need to tune the linear weights (β, α), and call SLFN as a neural network with random weights (NNRW) (Schmidt et al., 1992). It has been proved that the NNRW's approximation error converges to zero with the order $(C/\sqrt{L})$ (Igelnik and Pao, 1995; Pao et al., 1994), where L is the number of hidden nodes and C is a constant.

Notations

o	the dimensionality of the output
d	the dimensionality of the vector input
m	the first dimensionality of the matrix input
n	the second dimensionality of the matrix input
$\mathbf{u}_k$	the left projection vector on the kth hidden node
$\mathbf{v}_k$	the right projection vector on the kth hidden node
$g(\cdot)$	the active function
L	the number of hidden nodes
$\mathbf{t}^p$	the expected output of the pth pattern
$\mathbf{y}^p$	the actual output on the pth pattern

We have attempted to approximate the function $y = e^x - x\sin(x)\cos(x)$ using SLFNs (i.e. randomly generate 50 points on the interval $(-7, 2)$). Both the NNRW algorithm and the general gradient based full network training (BFGS) can achieve sufficient precision (see Fig. 1). However, they are different approaches that result in different problems.

When using the NNRW method, which is motivated by Monte Carlo integration, it only offers a statistical measure of the approximation quality. Its approximation rate is achievable only when the number of hidden nodes in the network is sufficient large (Tyukin and Prokhorov, 2009). After the input layer weights and biases are randomly assigned, we always use the least square method to tune the SLFN, which results in large-magnitude linear

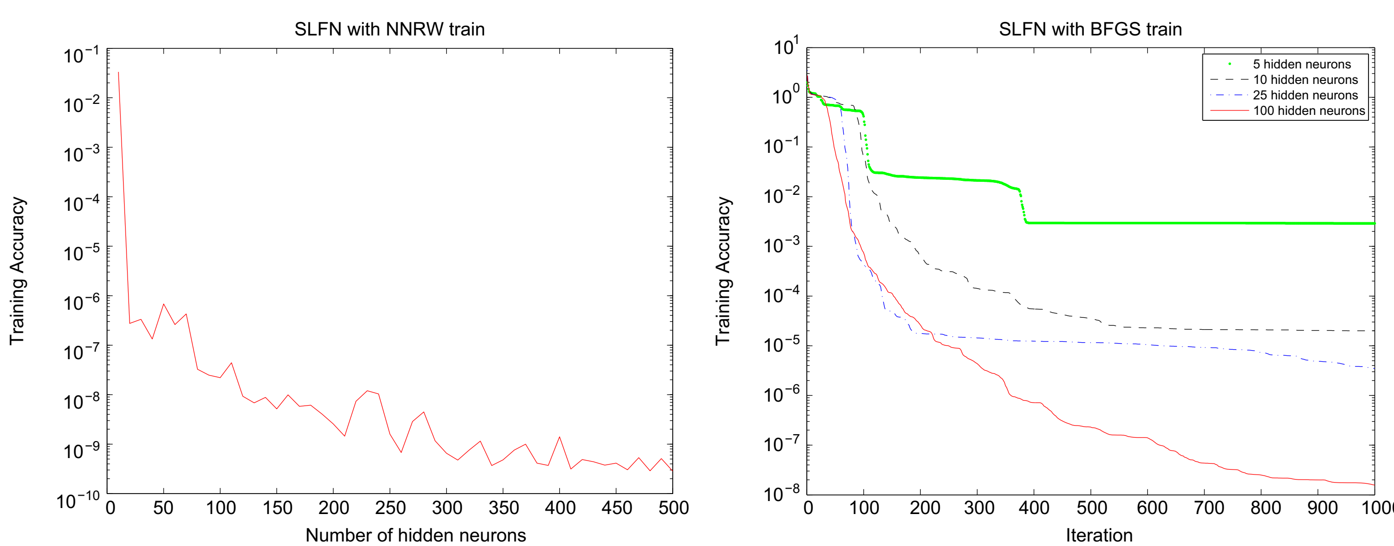

Fig. 1. Neural network training.