Neurocomputing 316 (2018) 49–58

Contents lists available at ScienceDirect

Neurocomputing

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

Brief papers

Cross-covariance regularized autoencoders for nonredundant sparse feature representation

Jie Chen [a,b,*], ZhongCheng Wu [a,c], Jun Zhang [a], Fang Li [a], WenJing Li [a], ZiHeng Wu [a]

[a] Chinese Academy of Sciences, Hefei Institute of Physical Science, Hefei, China
[b] Graduate School of Computer Applied Technology, University of Science and Technology of China, Hefei, China
[c] University of Science and Technology of China, Hefei, China

ARTICLE INFO

Article history:
Received 13 April 2018
Revised 7 June 2018
Accepted 28 July 2018
Available online 7 August 2018

Communicated by. Dr Xin Luo

Keywords:
Autoencoder
Cross-covariance
Deep learning
Feature representation
Receptive fields

ABSTRACT

We propose a new feature representation algorithm using cross-covariance in the context of deep learning. Existing feature representation algorithms based on the sparse autoencoder and nonnegativity-constrained autoencoder tend to produce duplicative encoding and decoding receptive fields, which leads to feature redundancy and overfitting. We propose using the cross-covariance to regularize the feature weight vector to construct a new objective function to eliminate feature redundancy and reduce overfitting. The results from the MNIST handwritten digits dataset, the NORB normalized-uniform dataset and the Yale face dataset indicate that relative to other algorithms based on the conventional sparse autoencoder and nonnegativity-constrained autoencoder, our method can effectively eliminate feature redundancy, extract more distinctive features, and improve sparsity and reconstruction quality. Furthermore, this method improves the image classification performance and reduces the overfitting of conventional networks without adding more computational time.

1. Introduction

Feature representation is an integral component of any object recognition task. A meaningful and representative feature can help obtain higher recognition and classification accuracies [1–6]. Autoencoder (AE) networks [7], an important branch of deep learning, have been widely used for feature extraction and are unsupervised in nature since they do not require labels for learning the features. The AE process takes the original object as its goal for network training, and the original object can be represented by extracting the representative features from the encoding layer. The unsupervised nature of the AE imparts unique advantages in terms of feature representation and learning, and researchers proposed some improved models of AE for specific problems in different fields [8–19], such as how to recover human pose from videos or images is a key issue in the field of human action recognition, researchers construct multi-modal or multi-task deep AE to achieve human pose recovery [13,14]. To improve the imaging quality of low-dose computed tomography (CT) images, researchers built stacked sparse denoizing AE for low-dose CT restoration [15]. In the field of image retrieval and ranking, scholars use AE to initially obtain distance metric in different visual spaces and construct a new ranking model [16]. In the field of driving behavior analysis, scholars use deep sparse AE to learn the multi-dimensional driving behavior hidden features for the mass driving behavior data acquired through the CAN bus to distinguish different styles of driving behavior [11]. In addition, scholars have realized multi-class classification by constructing a new AE expansion model such as denoizing AE, sparse AE and group sparse AE [18–20].

The AE architecture contains two types of parameters: nodes and weighted connections (filters or receptive fields (RFs)) [20–22]. As the number of nodes in the hidden layers increases, the number of weight parameters to be learned exponentially increases, which causes an important challenge in which the network parameters tend to overfit to the given training data [20].

Furthermore, our observations and those of other researchers [12,23,24] indicate that when models present high levels of overfitting, this condition is typically accompanied by the redundancy of feature weights. As a result, slight differences in feature weights capture similar patterns (classes) and affect the feature representation of models. This condition is particularly evident in [12], which presents a large number of similar weighted connections learned by the conventional sparse autoencoder (SAE) and nonnegativity-constrained autoencoder (NCAE) models, resulting in many extracted features that are duplicative and redundant.

* Corresponding author at: Chinese Academy of Sciences, Hefei Institute of Physical Science, Hefei, China.

E-mail addresses: cj2016@mail.ustc.edu.cn (J. Chen), zcwu@iim.ac.cn (Z. Wu), zhang_jun@hmfl.ac.cn (J. Zhang), lif@hmfl.ac.cn (F. Li), wenjingliwuxi@gmail.com (W. Li), ziheng88@mail.ustc.edu.cn (Z. Wu).

https://doi.org/10.1016/j.neucom.2018.07.050

In the present study, we focus on solving the problems of feature weights' redundancy and overfitting in conventional SAE and NCAE models. To this end, we propose using the cross-covariance (XCov) to regularize the feature weights. The proposed regularization method aims to reduce feature weights' correlation to minimize the XCov between all pairs of weight vectors in the encoding layer. Thus, redundant feature weights can be eliminated, extracting more distinct features and reducing overfitting.

Works related to ours include the following:

(i) XCov regularization [25–27]. Here, the researchers regularized the hidden layer activations by XCov and used the XCov function as a cost function to train the model. In contrast, we focus on the weight correlations rather than activation independence. Therefore, our algorithm directly regularizes the feature weight connections and not the hidden layer activations.

(ii) Orthogonal / correlation weight regularization. These works describe cosine similarity [28], deep canonical correlation analysis (deep CCA) [29] and correlational neural networks (CorrNets) [30]. Researchers have used the correlation coefficient/cosine similarity between feature weights to express the strength of the associated relationships. Our metric is different from that method; we use the XCov as a metric to measure the relationship between feature weight vectors.

(iii) The classic regularization methods include dropout [31], drop connect [32], and batch normalization [33]. Dropout aims to reduce co-adaptation of activations by randomly dropping units and their connections. Dropping weights with drop connect prevents feature co-adaptation. Batch normalization focuses on faster optimization by reducing internal co-variate shift, which is the constant variation of a layer's input as the process learns.

(iv) With feature weight clusters [12,34], researchers have used offline and online agglomerative clustering. This approach aims to reduce the number of feature weights according to the differentiation of weights based on their distances and the reconstruction error value. Different from previous approaches, we do not change the number of hidden layer nodes and feature weights. We directly introduce the concept of XCov into the cost function to minimize the XCov between feature weight vectors. Furthermore, related work on the K-sparse AE [35–37] aimed to reduce the number of feature weights by sorting the hidden units' activations and retaining the k largest units while setting the rest to zero. This method is similar to the feature weight cluster; the key idea is to reduce the number of feature weights.

Based on the previous research on the SAE [38–40], NCAE [8,10] and other derived works [12,34], the primary contributions of this study are as follows. (1) We propose a new feature weight regularization algorithm that uses the XCov to regularize the feature weights. We construct a new cost function for network training to eliminate feature redundancy, extract more representative and distinct hidden features and achieve better data reconstruction capabilities. In this way, the feature representation capabilities of the conventional NCAE model can be enforced. (2) Furthermore, we improve the image classification performance of the NCAE model to reduce overfitting.

This paper is organized as follows.

Section 2 introduces the basic theory of AE, SAE and NCAE. Section 3 introduces the proposed XCov regularization algorithm, and the solution for the optimization cost function is derived. Section 4 discusses the performance of the proposed algorithm and compares this algorithm with the SAE [40], NCAE [8,10], agglomerative clustering-SAE (Agglo-SAE) [12,34] and agglomerative clustering-NCAE (Agglo-NCAE) [12,34] in terms of eliminating the feature weight redundancy on several datasets (namely, MNIST, NORB and Yale). Then, we verify that our proposed method provides significantly better image classification performance on the MNIST and NORB datasets than the related deep networks (DNs) reported in the literature based on the SAE [40], NCAE [8], Agglo-SAE [12,34], Agglo-NCAE [12,34], Decov [25] and dropout AE (DpAE) [41]. Section 5 summarizes the results obtained and discusses the future work.

2. Nonnegativity-constrained autoencoder

An AE neural network is an unsupervised feature learning framework that tries to reconstruct its input vector at the output through unsupervised learning [21,42]. The general schematic of the AE is shown in Fig. 1. The network tries to learn a function

$$\mathrm{r} = f_{\mathrm{w,b}}(\mathrm{x}) \approx \mathrm{x} \tag{1}$$

where x is the input vector and r is the reconstructed vector. In addition, $\mathrm{W} = \{\mathrm{W}^{(1)}, \mathrm{W}^{(2)}\}$ and $\mathrm{b} = \{\mathrm{b}^{(1)}, \mathrm{b}^{(2)}\}$ represent the weights and biases of the encoding and decoding layers, respectively. To optimize the parameters of the model in (1), the average reconstruction error is used as the cost function

$$J_{AE}(\mathrm{W}, \mathrm{b}) = \frac{1}{2\mathrm{M}} \sum_{m=1}^{\mathrm{M}} \left\| r^{(m)} - x^{(m)} \right\|^2 \tag{2}$$

where M is the number of training samples.

To prevent overfitting in the AE, when the elements of W become large, we limit the elements of W with the L2 norm as a penalty term. In addition, we require that the future hidden layers are sparse since we wish to obtain more prominent features; therefore, we use the Kullback–Leibler (KL) divergence [43,44] to calculate the sparse item. Thus, the objective function of a conventional SAE has three components [8,45,46]

$$J_{SAE}(\mathrm{W}, \mathrm{b}) = J_{AE}(\mathrm{W}, \mathrm{b}) + \frac{\alpha}{2} \sum_{l=1}^{2} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l-1}} \left(\omega_{ij}{}^{(l)}\right)^2 + \beta J_{KL}(p \parallel \widehat{p}) \tag{3}$$

The first term on the right side of (3) is the reconstruction error. The second term represents the weight penalty term a that can control the strength of the penalty term ω_{ij}^{l} that expresses the connection between the jth unit in layer L-1 and the ith unit in layer l. The third term is the sparse item β used to adjust the size of the term. The sparse term is

$$J_{KL}(p \parallel \widehat{p}) = \sum_{j=1}^{n'} p \log \frac{p}{p_j} + (1-p) \log \frac{1-p}{p_j} \tag{4}$$

where n' is the number of hidden layer nodes and p is the sparsity target chosen to be a small positive number near 0. The average activation of this hidden unit is

$$p_j = \frac{1}{\mathrm{M}} \sum_{m=1}^{\mathrm{M}} h_j(\mathrm{x}^{(m)}) \tag{5}$$

To constrain the conventional SAE to extract non-negative latent features, the second term (i.e., the weight decay term) in (3) is replaced with J_N in (6) to penalize the negative weights [47]

$$J_N(\omega_{ij}{}^{(l)}) = \frac{\alpha}{2} \sum_{l=1}^{2} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l-1}} \begin{cases} (\omega_{ij}{}^{(l)})^2 & \omega_{ij}{}^{(l)} < 0 \\ 0 & \omega_{ij}{}^{(l)} \geq 0 \end{cases} \tag{6}$$

We ultimately obtain the following cost function for the NCAE [8]:

$$J_{NCAE}(\mathrm{W}, \mathrm{b}) = J_{AE}(\mathrm{W}, \mathrm{b}) + J_N(\omega_{ij}{}^{(l)}) + \beta J_{KL}(p \parallel \widehat{p}) \tag{7}$$