Neurocomputing 000 (2018) 1–17

Contents lists available at ScienceDirect

Neurocomputing

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

PCANet: An energy perspective

Jiasong Wu[a,b,d,*], Shijie Qiu[a,d], Youyong Kong[a,d], Longyu Jiang[a,d], Yang Chen[a,d], Wankou Yang[e], Lotfi Senhadji[c,d], Huazhong Shu[a,b,d]

[a] *LIST, Key Laboratory of Computer Network and Information Integration, Southeast University, Ministry of Education, Nanjing 210096, China*
[b] *International Joint Research Laboratory of Information Display and Visualization, Southeast University, Ministry of Education, Nanjing 210096, China*
[c] *University Rennes, INSERM, LTSI - UMR 1099, F-35000 Rennes, France*
[d] *Centre de Recherche en Information Biomédicale Sino-français (CRIBs), SEU, Univ Rennes, INSERM, Nanjing 210096, China*
[e] *School of Automation, Southeast University, Nanjing 210096, China*

ARTICLE INFO

Article history:
Received 12 January 2018
Revised 5 April 2018
Accepted 6 June 2018
Available online xxx

Communicated by Sato-Ilic Mika

Keywords:
Deep learning
Convolutional neural networks
PCANet
Error rate
Energy
Image recognition

ABSTRACT

The principal component analysis network (PCANet), which is one of the recently proposed deep learning architectures, achieves the state-of-the-art classification accuracy in various databases. However, the visualization or explanation of the PCANet is lacked. In this paper, we try to explain why PCANet works well from energy perspective point of view based on a set of experiments. The paper shows that the error rate of PCANet is qualitatively correlated with the inverse of the logarithm of BlockEnergy, which is the energy after the block sliding process of PCANet, and also this relation is quantified by using curve fitting method. The proposed energy explanation approach can also be used as a testing method for checking if every step of the constructed networks is necessary.

1. Introduction

Deep learning [1–10], especially convolutional neural networks (CNNs) [11,12], is a hot research topic that achieves the state-of-the-art results in many image classification tasks, including ImageNet large scale visual recognition [13–17], Labeled Faces in the Wild (LFW) face recognition [18–20], handwritten digit recognition [11,21], and other applications [22–31], etc. The great success of deep learning systems is impressive, but a fundamental question still remains: Why do they work [32]? In the recent years, several attempts have been made for explaining the deep learning systems. These attempts can be roughly categorized into two classes: theoretical explanation and experimental explanation.

Theoretical explanation method tries to elucidate deep learning systems by using various theories, which can be classified into seven subclasses: (1) *Renormalization Theory*. Mehta and Schwab [33] constructed an exact mapping from the variational renormalization group (RG) scheme [34] to deep neural networks (DNNs) based on Restricted Boltzmann Machines (RBMs) [1,2], and thus explained DNNs as a RG-like procedure to extract relevant features from structured data. (2) *Probabilistic Theory*. Patel et al. [32] developed a new probabilistic framework for deep learning based on a Bayesian generative probabilistic model. By relaxing the generative model to a discriminative one, their models recover two of the current leading deep learning systems: deep CNNs and random decision forests (RDFs). (3) *Information Theory.* Tishby and Zaslavsky [35] analyzed DNNs via the theoretical framework of the information bottleneck principle. Steeg and Galstyan [36] further introduced a framework for unsupervised learning of deep representations based on a hierarchical decomposition of information. (4) *Developmental robotic perspective.* Sigaud and Droniou [37] scrutinized deep learning techniques under the light of their capability to construct a hierarchy of multimodal representations from the raw sensors of robots. (5) *Geometric viewpoint*. Lei et al. [38] showed the intrinsic relations between optimal transportation and convex geometry and gave a geometric interpretation to generative models. Dong et al. [39] draw a geometric picture of the deep learning system by finding its analogies with two existing geometric structures, the geometry of quantum computations and the geometry of the diffeomorphic template matching. (6) *Group Theory.* Paul and Venkatasubramanian [40] explained deep learning system from the group-theoretic perspective point of view and showed why higher layers of deep learning framework tend to learn more

[*] Corresponding author at: LIST, Key Laboratory of Computer Network and Information Integration, Southeast University, Ministry of Education, Nanjing 210096, China.

E-mail address: jswu@seu.edu.cn (J. Wu).

https://doi.org/10.1016/j.neucom.2018.06.025
0925-2312/© 2018 Published by Elsevier B.V.

abstract features. Shaham et al. [41] discussed the approximation of wavelet functions using deep neural nets. Anselmi et al. [42] explained deep CNNs by invariant and selective theory, whose ideas come from *i*-Theory [43], which is a recent theory of feedforward processing in sensory cortex. (7) *Energy perspective*. The mathematical analysis of CNNs was performed by Mallat in [44], where wavelet scattering network (ScatNet) was proposed. The convolutional layer, nonlinear layer, pooling layer were constructed by prefixed complex wavelets, modulus operator, and average operator, respectively. Owning to its characteristic of using prefixed filters which are not learned from data, ScatNet was explained in [44] from *energy perspective* both in theory and experiment aspect, that is, ScatNet maintains the energy of image in each layer although using modulus operator. ScatNet achieves the state-of-the-art results in various image classification tasks [45] and was then extended to semi-discrete frame networks [46] as well as complex valued convolutional nets [47].

Experimental explanation methods tend to understand deep learning systems by inverting them to visualize the "filters" learned by the model [48]. For example, Larochelle et al. [49] presented a series of experiments on Deep Belief Networks (DBN) [1] and stacked autoencoder networks [3] by using artificial data and indicate that these models show promise in solving harder learning problems that exhibit many factors of variation. Goodfellow et al. [50] examined the invariances of stacked autoencoder networks [3] and also convolutional deep belief networks (CDBNs) [4] by using natural images and natural video sequences. Erhan et al. [48] studied three filter visualization methods (Maximizing the activation, Sampling from a unit of a network, Linear combination of previous layers' filters) on DBN [1] and Stacked Denoising Autoencoders [5]. Szegedy et al. [51] reported two intriguing properties of deep neural networks. Zeiler and Fergus [52] proposed DeConvNet method in which the network computations were backtracked to identify which image patches are responsible for certain neural activations. DeConvNet uses AlexNet [11] as an example to observe the evolution of features during training and to diagnose potential problems by using such a model. Simonyan et al. [53] demonstrated how saliency maps can be obtained from a Convnet by projecting back from the fully connected layers of the network. Girshick et al. [54] showed visualizations that identify patches within a dataset that are responsible for strong activations at higher layers in the model. Mahendran and Vedaldi [55] gave a general framework to invert CNNs and they tried to answer the following question: given an encoding of an image, to which extent is it possible to reconstruct the image itself?

Recently, Chan et al. [56] proposed a new deep learning algorithm called principal component analysis network (PCANet), whose convolutional layer, nonlinear processing layer, and feature pooling layer consist of principal component analysis (PCA) filter bank, binarization, and block-wise histogram, respectively. Chan et al. [56] also visualize the filters of PCANet like in [48] and [52]. Although PCANet is constructed with most basic units, it surprisingly achieves the state-of-the-art performance for most image classification tasks. PCANet arouses the interest of many researchers in this field. For example, Gan et al. proposed a graph embedding network (GENet) [57] for image classification. Wang and Tan [58] presented a C-SVDDNet for unsupervised feature learning. Feng et al. [59] presented a discriminative locality alignment network (DLANet) for scene classification. Ng and Teoh [60] proposed discrete cosine transform network (DCTNet) for face recognition. Gan et al. [61] presented a PCA-based convolutional network by combining the structure of PCANet and the LeNet-5 [11,12]. Zhao et al. [62] proposed multi-level modified finite radon transform network (MMFRTN) for image upsampling. Lei et al. [63] developed stacked image descriptor for face recognition. Li et al. [64] proposed SAE-PCA network for human gesture recognition in RGBD (Red, Green, Blue, Depth) images. Zeng et al. [65] proposed a quaternion principal component analysis network (QPCANet) for color image classification. Wu et al. [66] proposed a multilinear principal component analysis network (MPCANet) for tensor object classification. Although PCANet has been extensively investigated, the question still remains: Why it works well by using the most basic and simple units? To the best of our knowledge, no attempt to explain every step of the PCANet is available in the literature.

In this paper, (1) We present a new way to visualize, explain and understand every step of PCANet from an *energy perspective* on experiment aspect by using five image databases: Yale database [67], AR database [68], CMU PIE face database [69], ORL database [70], and CIFAR-10 database [71]. The proposed energy explanation approach can provide more information than the filter visualization method reported in [56]; (2) We shows qualitatively that the error rate of PCANet is correlated with the inverse of the logarithm of BlockEnergy, which is the energy after the block sliding process of PCANet, and then we try to find quantitatively their relations by using curve fitting method; (3) we show that the proposed energy explanation approach can be used as a testing method for checking if every step of the constructed networks is necessary; and (4) The energy explanation approach proposed in this paper can be extended to other PCANet-based networks [57–66].

The paper is organized as follows. PCANet is reviewed in Section 2. Section 3 presents an energy method to visualize, explain and understand every step of PCANet. Discussion is given in Sections 4 and 5 conclude the work.

2. Review of principal component analysis network

In this section, we first review the PCANet [56], whose architecture is shown in Fig. 1 and can be divided into three stages, including 10 steps. Suppose that we have N input training images $\{\mathbf{I}_i, i = 1, 2, \ldots, N\}, \mathbf{I}_i \in \mathbb{R}^{m\times n}$, and that the patch size (or two-dimensional filter size) of all stages is $k_1 \times k_2$, where k_1 and k_2 are odd integers satisfying $1 \le k_1 \le m$, $1 \le k_2 \le n$. We further assume that the number of filters in layer i is L_i, that is, L_1 for the first stage and L_2 for the second stage. In the following, we describe the structure of PCANet in detail.

Let the N input images $\{\mathbf{I}_i, i = 1, 2, \ldots, N\}$ be concatenated as follows:

$$\mathbf{I} = \begin{bmatrix}\mathbf{I}_1 & \mathbf{I}_2 & \cdots & \mathbf{I}_N\end{bmatrix} \in \mathbb{R}^{m\times Nn}. \tag{1}$$

2.1. The first stage of PCANet

As shown in Fig. 1, the first stage of PCANet includes the following 3 steps:

Step 1: the first patch sliding process.

The images are padded to $\mathbf{I}'_i \in \mathbb{R}^{(m+k_1-1)\times(n+k_2-1)}$ before sliding operation. Out-of-range input pixels are taken to be zero. This can ensure all weights in the filters reach the entire images. We use a patch of size $k_1 \times k_2$ to slide each pixel of the ith image $\mathbf{I}'_i \in \mathbb{R}^{(m+k_1-1)\times(n+k_2-1)}$, then reshape each $k_1 \times k_2$ matrix into a column vector, which is then concatenated to obtain a matrix

$$\mathbf{X}_i = \begin{bmatrix}\mathbf{x}_{i,1} & \mathbf{x}_{i,2} & \cdots & \mathbf{x}_{i,mn}\end{bmatrix} \in \mathbb{R}^{k_1k_2\times mn}, \quad i = 1, 2, \ldots, N, \tag{2}$$

where $\mathbf{x}_{i,j}$ denotes the jth vectorized patch in $\mathbf{I}_i$.

Therefore, for all the input training images $\{\mathbf{I}_i, i = 1, 2, \ldots, N\}$, we can obtain the following matrix

$$\mathbf{X} = \begin{bmatrix}\mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_N\end{bmatrix} \in \mathbb{R}^{k_1k_2\times Nmn}, \tag{3}$$

Step 2: the first mean remove process.

In this step, we subtract patch mean from each patch and obtain

$$\bar{\mathbf{X}}_i = \begin{bmatrix}\bar{\mathbf{x}}_{i,1} & \bar{\mathbf{x}}_{i,2} & \cdots & \bar{\mathbf{x}}_{i,mn}\end{bmatrix} \in \mathbb{R}^{k_1k_2\times mn}, \quad i = 1, 2, \ldots, N, \tag{4}$$