Neurocomputing 290 (2018) 26–33

Contents lists available at ScienceDirect

Neurocomputing

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

Deep Rotation Equivariant Network

Junying Li[a], Zichen Yang[b], Haifeng Liu[b], Deng Cai[a,*]

[a] The State Key Laboratory of CAD&CG, College of Computer Science, Zhejiang University, Hangzhou, China
[b] College of Computer Science, Zhejiang University, Hangzhou, China

ARTICLE INFO

Article history:
Received 16 November 2017
Revised 22 January 2018
Accepted 7 February 2018
Available online 15 February 2018

Communicated by Jun Yu

Keywords:
Neural network
Rotation equivariance
Deep learning

ABSTRACT

Recently, learning equivariant representations has attracted considerable research attention. Dieleman et al. introduce four operations which can be inserted into convolutional neural network to learn deep representations equivariant to rotation. However, feature maps should be copied and rotated four times in each layer in their approach, which causes much running time and memory overhead. In order to address this problem, we propose Deep Rotation Equivariant Network consisting of cycle layers, isotonic layers and decycle layers. Our proposed layers apply rotation transformation on filters rather than feature maps, achieving a speed up of more than 2 times with even less memory overhead. We evaluate DRENs on Rotated MNIST and CIFAR-10 datasets and demonstrate that it can improve the performance of state-of-the-art architectures.

1. Introduction

Convolutional neural networks (CNNs) recently have made great success in computer vision tasks [1–5]. One of the reasons to its success is that weight sharing of convolution layers ensures the learnt representations are translation equivariant [6], i.e., shifting an image and then feeding it through the network is the same as feeding the original image and then shifting the resulting representations.

However, CNNs fail to exploit rotation equivariance to tackle vision problems on datasets with rotation symmetry in nature, especially microscopic images or aerial images, which can be photographed from any angle. Thus, current studies focus on dealing with this issue.

One widely used method to achieve rotation equivariance is to constrain the filters of the first convolutional layer to be rotated copies of each other, and then apply cross-channel pooling immediately after the first layer [7–9]. However, only shallow representations equivariant to rotation can be learnt by applying one convolutional layer. In addition, such representations are nearly trivial, since pooling rotated copies is approximately equivalent to convolving non-rotated inputs with highly symmetric filters.

To solve this problem, Dieleman et al. [10] introduce four operations which can be combined to make these models able to learn deep representations equivariant to rotation. However, every feature map should be copied and rotated by these operations four times, which causes high memory and running time overhead.

In this paper, we give a comprehensive theoretical study on approaches to rotation equivariance with CNNs. We propose a novel CNN framework, Deep Rotation Equivariant Network (DREN) to obtain deep equivariance representations. We prove that DREN can achieve the identical output to that of [10] with much less running time and memory requirements.

We evaluate our framework on two datasets, Rotated MNIST and CIFAR-10. On Rotated MNIST, it can outperform the existing methods with less number of parameters. On CIFAR-10, it can improve the results of state-of-the-art models with the same number of parameters. Moreover, our implementation achieves a speed up of more than 2 times as that of [10], with even less memory overhead.

2. Related works

Learning invariant representations by neural networks has been studied for over a decade. Early works focus on refinement of restricted Boltzmann machines (RBMs) and deep belief nets (DBNs). Kavukcuoglu et al. [11] give an approach to automatically generate topographic maps of similar filters in an unsupervised manner and these filters can produce local invariance when being pooled together. Norouzi et al. [12] develop convolutional RBM (c-RBM), using weight sharing to achieve shift-invariance. Later, a following work by Schmidt and Roth [13] incorporates linear transformation invariance into c-RBMs, yielding features that have a notion of transformation performed. The model proposed by Lee et al.

* Corresponding author.
E-mail addresses: microljy@zju.edu.cn (J. Li), haifengliu@zju.edu.cn (H. Liu), dengcai@cad.zju.edu.cn (D. Cai).

https://doi.org/10.1016/j.neucom.2018.02.029
0925-2312/

[14] uses a probabilistic max-pooling layer to support efficient probabilistic inference, which also shows the property of translation invariance.

Recently, convolutional neural networks have become the most popular models in various computer vision tasks [15–18]. One of the advantages of CNNs is its translation equivariant property provided by weight sharing [6]. However, it cannot deal with rotation transformation of input images. Thus, many variants of CNNs have been proposed to settle these problems. Basically, the idea of most of the related works [9,19,20] is to stack rotated copies of images or features to obtain rotation equivariance.

There are also other methods. Gens and Domingos [21] propose deep symmetry networks that can form feature maps over arbitrary transformation groups approximately. Methods proposed by Wu et al. [7,8] show that rotation convolution layers followed by a cross-channel pooling over rotations could achieve rotation equivariance. In fact, none of the directional features could be extracted by these methods, since pooling is applied right after one rotation convolution. Cohen and Welling [6] propose a group action equivariant framework by stacking group acted convolution and provide a theoretically grounded formalism to exploit symmetries of CNNs. Worrall et al. [22] present harmonic networks, a CNN structure exhibits equivariance to patch-wise translation and 360-rotation. Recently, the vector field network proposed by Marcos et al. [23] apply interpolation to deal with rotation of general degrees.

Dieleman et al.[10] introduce four operations to encode rotation symmetry into feature maps to build a rotation equivariant neural network. However, feature maps should be rotated each time to ensure equivariance, which obviously costs much time and memory. Our approach presents a different way to overcome this issue by rotating filters, which brings about exactly the same results, but in a more efficient way.

3. Equivariance and invariance

In this section, we briefly discuss the notions of equivariance and invariance of image representations. Formally, a representation of a CNN can be regarded as a function f mapping from image spaces to feature spaces.

We say, a representation f is equivariant to a family $\mathfrak{T}$ of transformations on image spaces, if for any transformation $\mathcal{T} \in \mathfrak{T}$, there exists a corresponding transformation $\mathcal{T}'$ on feature spaces, such that

$$f(\mathcal{T}x) = \mathcal{T}'f(x), \tag{1}$$

for any input images x. Intuitively, this means that the learnt representation f of CNNs changes in an expected way, when the input image is transformed.

There is another stronger case when $\mathcal{T}'$ is the identity map, i.e., the map fixing the inputs, for all $\mathcal{T} \in \mathfrak{T}$. This indicates that the representations remain unchanged no matter how the input data is transformed by transformations in $\mathfrak{T}$, i.e., the representation is invariant. Invariance is an ideal property of representation, because a good object classifier must output an invariant class label no matter what location of the object lies in.

The goal of this paper is to present a novel convolutional neural network framework, which learns representations that are equivariant to rotation transformations $\mathcal{R}$, s. t. $\mathcal{R}' = \mathcal{R}$, that is

$$f(\mathcal{R}x) = \mathcal{R}f(x), \tag{2}$$

for any input image x. Fig. 1 gives an example of rotation equivalent representations learnt by DREN, comparing to a traditional CNN. The reason that we do not directly work on rotation invariant representations is that this kind of rotation equivariance can be easily lifted to rotation invariance, for instance, by a global pooling operation [24], i.e., the kernel size of this pooling layer is equal to the size of feature maps.

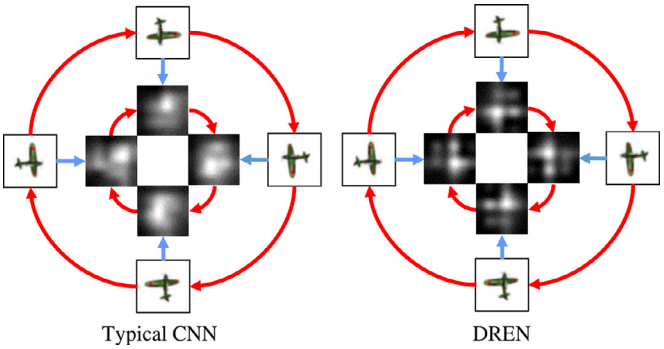

Fig. 1. Latent representations learnt by a CNN and a DREN (Proposed), where R stands for clockwise rotation. The left part is the result of a typical CNN while the right one is that of a DREN. In both parts, the outer cycles consist of the rotated images while the inner cycles consist of the learnt representations. Features produced by a DREN is equivariant to rotation while that produced by a typical CNN is not.

Since these are the only four kinds of possible rotation of an image that can be performed without interpolations, we mainly deal with the rotation transformation family $\mathfrak{R} = \{\mathcal{R}_\theta | \theta = k\pi/2, k \in \mathbb{Z}\}$. However, our experimental results show that our framework can achieve good performance when dealing with rotation for general degrees.

4. Rotation equivariant convolution

In this section, we define three novel types of convolutional layers, which are combined to learn rotation equivariant features.

4.1. Preliminaries

For the sake of simplicity, we omit bias terms, activation functions and other structures, concentrating on convolution. In addition, we set the stride of any convolution layer be 1. The general case will be discussed in Section 4.5.

First, we vectorize convolution operation formally to simplify our derivation. Shortly, we shall use matrix multiplication to describe multi-channel convolution. Let us assume that the input of a convolutional layer contains n feature maps(images) $\{x_j\}_{j=1}^n$. This layer has mn filters, denoted by W_{ij} with $1 \le i \le m$, $1 \le j \le n$. These can be organized as a matrix(vector) x of size n and a matrix W of size m, n in which entries are filters or feature maps. We refer to such a matrix (vector) a hyper-matrix (hyper-vector). Then, the convolution W^*x is defined to be a hyper-vector of size m whose ith entry is

$$\sum_{1 \le j \le n} W_{ij} * x_j, \tag{3}$$

for each $1 \le i \le m$. One can verify that this is actually equivalent to ordinary multi-channel convolution.

Next, we introduce the rotation operator $\mathcal{R}$, rotating a filter or a feature map by a degree of $\pi/2$ counterclockwise. We also define the action of $\mathcal{R}$ on a hyper-matrix W. $\mathcal{R}(W)$ is defined to be entrywise rotation. There are three obvious facts about the rotation operator that are frequently used in the sequel.

1. Convolutionally distributive law: $\mathcal{R}(W * x) = \mathcal{R}(W) * \mathcal{R}(x)$. This indicates that rotating filters and feature maps simultaneously before convolution yield rotated outputs.
2. Additively distributive law: $\mathcal{R}(X_1 + X_2) = \mathcal{R}(X_1) + \mathcal{R}(X_2)$. Note that + is entrywise addition of matrices.
3. Cyclic law: $\mathcal{R}^4$ is equal to the identity transformation.