Group Equivariant Convolutional Networks

September 13, 2021

Marco De Tommasi

Abstract

In this report are described the principal properties of the Group Equivariant Convolutional Networks (G-CNNs). These networks, has been introduced recently and there are a lot of possible applications that involves different topics like physics, biology and image classification in general. For some discrete groups, the implementation of this networks is straightforward, so their performaces can be easly compared with the standard architectures. In the following are implemented two examples of G-CNNs, for the p4 and p4m groups and tested on the rotated Minist, Cifar10 and Cifar10+ datasets. In the last section, are presented some of the latest possible implementations and some new possible generalizations.

1. Introduction

The main properties of the G-CNNs are related to the concept of equivariance. Recalling the standard convolution operation of functions f and ψ

$$(f \star \psi)(x) = \int f(t)\psi(x-t)\, dt \tag{1}$$

If we define the shift operator $(\lambda_t f)(x) = f(x-t)$, we notice that an important property of convolution is the *shift-equivariance*: $\lambda_t f \star \psi = \lambda_t(f \star \psi)$. If the filter is designed to respond to some pattern in f, this property tells us that the filter will respond in the same way no matter where the pattern is finded on the image. The main objective of the G-CNNs, is to allow a pattern recognition in a more general way. We would like to hold the *equivariance* property, but now we must generalize this concept for others operations too (instead of just translations as for standard convolutions) (Estevens, 2020). It is possible to define the concept of *group-equivariance* and define it with respect to the action of a group[1].

Equivariance Let T_g and T'_g indicate the group action[2] for some $g \in G$. A given linear map (or a layer) $\Phi : X \to Y$ is equivariant to actions G when:

$$\Phi(T_g\, x) = T'_g \Phi(x) \tag{2}$$

This definition assure that the network or layer Φ that maps one representation to another should be *structure preserving* (Cohen, 2016). The equivariance constrain the network to encode some other type of symmetries leading to better generalizations.

Group-Equivariant Convolution Although convolutions are equivariant to *traslations* they are not equivariant to other isometries, like for instance, the rotation. What can be demostrated is that a convolution of a rotated image $L_r f$, where L_r is the rotation operator, convolved with a filter ψ, is the same as the rotation by r of the original image f convolved with the inverse rotated filter $L_{r^{-1}}\psi$. For this reason, if a CNN learns rotated copies of the same filter, the total feature maps given by stacking togheter the single feature maps obtained with different rotated filters is equivariant, although individual feature maps are not (Cohen, 2016).

2. Methods

In order to define the group convolution for the groups p_4 (rotation and translation) and $p4m$ (rotation, translation and reflection) is necessary, for what said before, to build a first layer, that maps our inputs images into an element of space given by the action of the group one the space we are working with. In practice, this means that, given a function defined in the space $X \in \mathbb{Z}^2$, the action[3] of the group convolution will give us a solution that is no more in the space X but in a bigger dimension space that we call Y, and con-

Email: Marco De Tommasi <detommasi.1760258@studenti.uniroma1.it>.

Deep Learning and Applied AI 2021, Sapienza University of Rome, 2nd semester a.y. 2020/2021.

[1] For Group definition see (Estevens, 2020) pg(3-8)

[2] See https://en.wikipedia.org/wiki/Group_action

[3] In this case, the action of group G on the functions f is defined as $[g.f](h) := f(g^{-1} \cdot h)$

tain all the possible solution given by the action of the elements of G on the elements of space X. Let's consider the equation (1) and, by replacing the shift by a more general transformation for some group G, we get

$$[f \star \psi](g) = \sum_{y \in \mathbb{Z}^2} \sum_k f_k(y)\psi_k(g^{-1}y) \tag{3}$$

Notice that both f and ψ are functions of the plane $\mathbb{Z}^2$ but the feature map $f \star \psi$ is a function defined on the group Y . The starting layer of every G-CNN will always be of this type. For all the layers after the first, the mapping will be from an element of space Y to itself.

$$[f \star \psi](g) = \sum_{h \in G} \sum_k f_k(h)\psi_k(g^{-1}h) \tag{4}$$

where h and g are both element of the group space G and the output ψ and f are now defined on the space Y given by the action of the group G on the space X where f and ψ are defined. In practice, for some kind of group G like $p4$, the group elements can be splitted into traslation and rotations separately. This means that the G-correlation $(f \star \psi)$ can be computed as a filter transformation $L_s\psi$ and for all four rotations (or the eight rotation-flips for $p4m$) and then use a fast planar convolution on the f and the augmented filter bank (Cohen, 2016). Notice that this procedure is applied to the functions $\psi(h)$, so the filter is already in Y space, which means that there are in the case of $p4$ group, four rotared filters that will be again rotated, for a total of a 4×4 matrix permutation.

3. Experimental Results [4]

Rotated MNIST. For this evaluation (and the next one on Cifar10 too) it has been used the validation set and a number of epochs of 10, 20 and 30. Here are reported the averaged results. For MNIST it has been used a CNN architecture with a total of 5 convolutional layers. In each layer is used a ReLU activation function and a maxpooling. For the optimization it has been used Adam algorithm. For the p4CNN, the convolutional layers has been replaced with the p4-convolution, with a number of input and output features divided by $\sqrt{4} = 2$. In this way we have that the networks have roughly the same number of parameters, this can be useful for a fair comparison of the models. The pooling for the p4-convolutional layers has been evaluated on the spatial dimension only for the first layer. For last layer the pooling is evaluated through the group dimension of the filter too (4 for $p4$ group).

Cifar10. For this evaluation it has been used a CNN with a total o 4 convolutional layers, with ReLU and Maxpooling at each layer of the network. This time dropout has been used giving better performances. For the optimization it has been used Adam algorithm. As seen before, in order to have similar number of parameters, the input and output features of the convolutional layers has been divided by a factor $\sqrt{n}$ where n is the group dimension (4 or 8). Even in this case, the pooling through the group dimension has been done only in the last layer, for better performances (Cohen, 2016). The network has been tested on different datasets in order to evaluate their performances in case of data augmentation. The last two datasets are the Cifar10 with random horizontal flips and translations Cifar10+ and Cifar10+R is the same datasets with rotation in the range $[0; 2\pi]$.

[4] https://github.com/MarcoDeTommasi/DL-AI-Project

Table 1. Performance comparison between Z2CNN and P4CNN for Rotated MNIST Dataset.

Networks	Accuracy(%)	N. Parameters
Z2CNN	83.1%	393K
P4CNN	**92.7%**	395K

Table 2. Performance comparison between Z2CNN, P4CNN and P4MCNN for Cifar10, Cifar10+ and Cifar10+ with rotations.

Networks	Acc.(%) Cifar10	Acc.(%) Cifar10+	Acc.(%) Cifar10+R	N. weights
Z2CNN	70.0 %	73.0%	53.5%	246K
P4CNN	71.2%	73.9%	62.1%	262K
P4MCNN	70.1%	73.7%	64.1%	227K

4. Discussion and Possible Applications

The results show that $p4$ and $p4m$ convolution layers can be used as a good replacement of standard convolutions, improving the results in specific cases. The results are great for the rotated MNIST while for CIFAR10 the performances seems to be roughly the same, this can be due to the fact that this dataset does not present a specific symmetry of the kind implemented. In general the G-CNNs seems to benefit from the data augmentation in the same way as convolutional networks, for Cifar10+ the general performances increased while, for a significative change of the data like for Cifar10+R, the G-CNNs seems to be more robust in the learning process respect to the classical CNN. Knowing the stat-of-art of the GCNN-s, like steerable CNNs (Weller, 2021) that generalize GCNNs to isometies and continuous spaces E(2), a new possible implementation could involve other kind of mathematical transformations like homothety in order to achieve some new possible symmetry recognition for same deformed features. Another interesting implementation is the Attentive GCNNs that through a self-attention accentuate meaningful symmetry combinations and suppress non-plausible ones (Romero, 2020).

References

Cohen, T. Group equivariant convolutional networks, https://arxiv.org/pdf/1602.07576.pdf. 2016.

Estevens, C. Theoretical aspects of group equivariant neural networks, https://arxiv.org/pdf/2004.05154.pdf. 2020.

Romero, D. Attentive group equivariant convolutional networks https://arxiv.org/pdf/2002.03830.pdf. 2020.

Weller, M. General e(2)-equivariant steerable cnns, https://arxiv.org/pdf/1911.08251.pdf. 2021.