Group Equivariant Convolutional Networks

October 3, 2022

Giovanni Pica 1816394

Abstract

In this paper will be introduced the main concepts of the Group Equivariant Convolutional Neural Networks (Cohen & Welling, 2016a) (G-CNNs) that are a natural generalization of the original Convolutional Neural Networks and those networks can be generalized to exploit larger groups of symmetries, including rotations and reflections.

1. Introduction

We know that Convolutional Neural Network are translation equivariant, which means in few words that applying the convolution on a translated input is the same thing to do the translation of the convolution of the input. In other words, the symmetry (translation) is preserved by each layer, which makes it possible to exploit it not just in the first, but also in higher layers of the network. But we also know that those conventional neural networks are not for example rotation equivariant as explained in the figure 1.

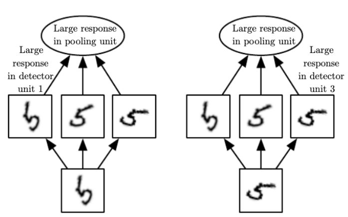

Figure 1. The digit "5" presented to the CNN along with a set of filters the CNN has learned (middle). Since there is a filter inside the CNN that has "learned" what a "5" looks like, rotated by 10 degrees, it fires and emits a strong activation (response). This large activation is captured during the pooling stage and ultimately reported as the final classification.

Group Equivariant Convolutional Neural Networks (Cohen & Welling, 2016a) (G-CNNs) are a natural generalization of the original Convolutional Neural Networks and those networks can be generalized to exploit larger groups of symmetries, including rotations and reflections. This is really important because the added geometry structure allows to deal with context, with the recognition of components (important for example in medical analysis) and also the data augmentation is obsolete in those networks. These networks are based on groups, a group G is a set that with a binary operation given two elements a,b ∈ G it produces another element ab and also it satisfies identivity, associativity, closure and inverse. In those architectures there is some layer or network ϕ that maps one representation which in this particular case has a structure of a G-space for some chosen group G instead of $\mathbb{R}^n$. This map should be structure preserving which means that for G-Spaces should be equivariant:

$$\phi(T_g x) = T'_g \phi(x)$$

That is, transforming an input x by a transformation g (forming $T_g x$) and then passing it through the learned map ϕ should give the same result as first mapping x through ϕ and then transforming the representation. The G-groups used are: the symmetry group that is the set of transformations that leaves the object invariant (an example is the set of 2D integer translations $\mathbb{Z}^2$), the group p4 that consists of all compositions of translations and rotations by 90 degrees about any center of rotation in a square grid and the group operation is given by matrix multiplication; the group pm4 that consists of all compositions of translations, mirror reflections and rotations by 90 degrees about any center of rotation in the grid.

2. Implementation

First of all is important to describe the functions on the groups because is useful to understand better how it works. Let's take a transformation g acting on a set of feature maps as function $f : \mathbb{Z}^2 \rightarrow \mathbb{R}^K$ where K denotes the number of channels. We have this notation:

$$[L_g f](x) = [f \circ g^{-1}](x) = f(g^{-1}x)$$

This says that the operator L_g is an instantiation of the transformation operator T_g and we have:

$$L_g L_h = L_{gh}$$

Email: Giovanni Pica <pica.1816394@studenti.uniroma1.it>.

Deep Learning and Applied AI 2022, Sapienza University of Rome, 2nd semester a.y. 2021/2022.

So g represents a pure translation $t = (u, v) \in \mathbb{Z}^2$ and $g^{-1}x$ simply means $x - t$. Let's move on to how are done the G-CNNs. We have three layers which are G-convolution, G-pooling and nonlinearity and each one commutes with G-transformation of the domain of the image. In the first layer there is an operation called G-correlation that is like the normal correlation function and it's replaced the shift function with a more general transformation from some group G. On the other two layers there are a pooling layer and a nonlinearity map applied to a feature map amounts to function composition and it's used for inherits the transformation properties of the previous layers. The pooling is divided in two steps: the pooling itself and a subsampling step, one is without stride and the other with that. In a G-CNN the notion of "stride" is generalized by subsampling on a subgroup $H \subset G$, so H is a subset of G that is itself a group and the subsampled feature map is then equivariant to H but not G. An example could be a p4 feature map, pooling can be performed over all four rotations at each spatial position (the cosets of the sub-group R of rotations around the origin). The resulting feature map is a function on $\mathbb{Z}^2 \simeq p4/R$ so it will transform in the same way as the input image.

3. Related Work

These new kind of convolutional neural networks have reached a big popularity in the years after the publication. They inspires some nice works, for example two good generalizations proposed by the same author were the Steerable CNNs (Cohen & Welling, 2016b) and the Spherical CNNs for 3D data (Cohen et al., 2018). There are also some interesting work such as the Group Equivariant GAN (Dey et al., 2020), PDE-based Group Equivariant Convolutional Neural Networks (Smets et al., 2020) Equivariant Graph Networks (Satorras et al., 2021), a self-attention module that is equivariant under 3D roto-translations (Fuchs et al., 2020). These networks also were tested in many areas such as Trajectory Prediction with Equivariant Continous Convolution (Walters et al., 2020), 3D G-CNNs for polmunary nodule detection (Winkels & Cohen, 2018) and also on other medical image analysis tasks.

4. Experiments

The following results are shown for confirm the results presented by (Cohen & Welling, 2016a).

4.1. Rotated MNIST

The model used for test this architecture on this dataset is like the pytorch example on the repository `https://github.com/pytorch/examples` and the Convolutional and Dropout layers are replaced with four G-Convolutional layers that uses the G-group p4. After fifty epochs the test error is **1.33**, so the great accuracy can be confirmed.

4.2. CIFAR10

The model used for this dataset is like the example on this repository `https://github.com/kuangliu/pytorch-cifar`, all planar convolutions are replaced with p4m group convolution and the number of filters in each convolutional layer was reduced by sqrt(8) to keep similar number of parameters. The architecture used is the ResNet18 (He et al., 2016) and it gives a good result. After 280 epochs the test error is **5.8** and compared with the same architecture but with planar convolutional layer that has a test error of **6.98**, it's a good result. So also for the CIFAR10 the performances of these particular kind of convolution neural networks are confirmed.

Code. The code is in `https://github.com/bonjon/DeepLearningProject` that contains two notebooks for RotatedMNIST and CIFAR10.

5. Future Work

These networks were widely studied and generalized in these years, we have an implementation for 3D data, medical analysis and some good generalization. Most of all the tasks that are in the literature are about image classification, so a good idea to confirm more the performances of the G-CNNs is to perform some other tasks, for example object detection and semantic segmentation can be two interesting task to do. Another task that I was thinking about could be to use the GAN proposed in (Dey et al., 2020) to generate some new images for example a sketch image to a realistic one. Another example that could follow the same logic as the previous one is to use the Equivariant Graph Neural Network (Satorras et al., 2021) to replace the encoder and the decoder of the two-step molecule generator (Bresson & Laurent, 2019).

References

Bresson, X. and Laurent, T. A two-step graph convolutional decoder for molecule generation. *CoRR*, abs/1906.03412, 2019. URL `http://arxiv.org/abs/1906.03412`.

Cohen, T. and Welling, M. Group equivariant convolutional networks. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2990–2999, New York, New York, USA, 20–22 Jun

2016a. PMLR. URL https://proceedings.mlr.press/v48/cohenc16.html.

Cohen, T. S. and Welling, M. Steerable cnns. *CoRR*, abs/1612.08498, 2016b. URL http://arxiv.org/abs/1612.08498.

Cohen, T. S., Geiger, M., Köhler, J., and Welling, M. Spherical cnns. *CoRR*, abs/1801.10130, 2018. URL http://arxiv.org/abs/1801.10130.

Dey, N., Chen, A., and Ghafurian, S. Group equivariant generative adversarial networks. *CoRR*, abs/2005.01683, 2020. URL https://arxiv.org/abs/2005.01683.

Fuchs, F. B., Worrall, D. E., Fischer, V., and Welling, M. Se(3)-transformers: 3d roto-translation equivariant attention networks. *CoRR*, abs/2006.10503, 2020. URL https://arxiv.org/abs/2006.10503.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Satorras, V. G., Hoogeboom, E., and Welling, M. E(n) equivariant graph neural networks. *CoRR*, abs/2102.09844, 2021. URL https://arxiv.org/abs/2102.09844.

Smets, B. M. N., Portegies, J., Bekkers, E. J., and Duits, R. Pde-based group equivariant convolutional neural networks. *CoRR*, abs/2001.09046, 2020. URL https://arxiv.org/abs/2001.09046.

Walters, R., Li, J., and Yu, R. Trajectory prediction using equivariant continuous convolution. *CoRR*, abs/2010.11344, 2020. URL https://arxiv.org/abs/2010.11344.

Winkels, M. and Cohen, T. S. 3d g-cnns for pulmonary nodule detection. *CoRR*, abs/1804.04656, 2018. URL http://arxiv.org/abs/1804.04656.