3D Group-Equivariant Neural Networks for Octahedral and Square Prism Symmetry Groups

Marysia Winkels
University of Amsterdam / Aidence
marysia@aidence.com

Taco S. Cohen
University of Amsterdam
taco.cohen@gmail.com

Abstract

Convolutional Neural Networks (CNNs) require a large amount of annotated data to learn from, which is often difficult to obtain. In this paper we show that the sample complexity of 3D CNNs can be significantly improved by using 3D roto-translation group convolutions (G-Convs) instead of the more conventional translational convolutions. These 3D G-CNNs were applied to the problem of false positive reduction for pulmonary nodule detection, and proved to be substantially more effective in terms of performance and speed of convergence compared to a strong and comparable baseline architecture with regular convolutions, data augmentation and a similar number of parameters. For every dataset size tested, the G-CNN achieved a score close to the CNN trained on ten times more data.

1 Introduction

Many recent efforts in machine learning have focused on learning from large (annotated) datasets. However, in data-limited domains such as healthcare, there is a need for *data efficient* solutions. Relative to fully connected networks, CNNs are already quite data efficient due to the translational weight sharing in the convolutional layers. One important property of convolution layers that enables translational weight sharing, but is rarely discussed explicitly, is *translation equivariance*: a shift in the input of a layer leads to a shift in the output, $f(T\mathbf{x}) = Tf(\mathbf{x})$. Because each layer in a CNN is translation equivariant, all internal representations will shift when the network input is shifted, so that translational weight sharing is effective in each layer of a deep network.

Many kinds of patterns maintain their identity not just under translation, but also under other transformations such as rotation and reflection. So it is natural to ask if CNNs can be generalized to other kinds of transformations, and indeed it was shown that by using *group convolutions*, weight sharing and equivariance can be generalized to essentially arbitrary groups of transformations [1]. Although the general theory of G-CNNs is now well established [1, 2, 3, 4], a lot of work remains in developing easy to use group convolution layers for various kinds of input data with various kinds of symmetries. This is a burgeoning field of research, with G-CNNs being developed for discrete 2D rotation and reflection symmetries [1, 5, 2], continuous planar rotations [6, 7, 8], 3D rotations of spherical signals [9], and permutations of nodes in a graph [10].

In this paper, we develop G-CNNs for three-dimensional signals (such as volumetric CT images) acted on by discrete translations, rotations, and reflections. This is highly non-trivial, because the discrete roto-reflection groups in three dimensions are non-commutative and have a highly intricate structure. We show that when applied to the task of false-positive reduction for pulmonary nodule detection in chest CT scans, 3D G-CNNs show remarkable data efficiency, yielding similar performance to CNNs trained on $10\times$ more data. Our implementation of 3D group convolutions is publicly available[1], so that using them is as easy as replacing `Conv3D()` by `GConv3D()`.

Parts of this paper appeared previously in the first author's thesis.

[1] https://github.com/tscohen/GrouPy

2 Three-dimensional G-CNNs

This section will explain the 3D group convolution in an elementary fashion. The goal is to convey the high level idea, focusing on the algorithm rather than the underlying mathematical theory, and using visual aids where this is helpful. For the general theory, we refer the reader to [1, 2, 3, 4].

To compute the conventional (translational) convolution of a filter with a feature map, the filter is translated across the feature map, and a dot product is computed at each position. Each cell of the output feature map is thus associated with a translation that was applied to the filter. In a group convolution, additional transformations like rotations and reflections are applied to the filters, thereby increasing the degree of weight sharing. More specifically, starting with a canonical filter with learnable parameters, one produces a number of transformed copies, which are then convolved (translationally) with the input feature maps to produce a set of output feature maps. Thus, each learnable filter produces a number of *orientation channels*, each of which detects the same feature in a different orientation.

As shown in [1], if the transformations that are applied to the filters are chosen to form a *symmetry group* H, the resulting feature maps will be equivariant to transformations from this group (as well as being equivariant to translations). More specifically, if we transform the input by $h \in H$ (e.g. rotate it by 90 degrees), each orientation channel will be transformed by h in the same way, *and* the orientation channels will get shuffled by a permutation matrix $\rho(h)$. The channel-shuffling phenomenon occurs because the transformation h changes the orientation of the input pattern, so that it gets picked up by a different orientation channel / transformed filter. The particular way in which the channels get shuffled by each element $h \in H$ depends on the structure of H (i.e. the way transformations $g, k \in H$ compose to form a third transformation $h = gk \in H$).

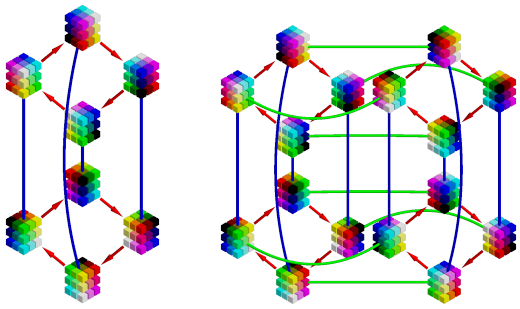


Figure 1: Cayley diagrams of the groups D_4 (left) and D_{4h} (right).

In the simplest instantiation of G-CNNs, the group H is the set of 2D planar rotations by $0, 90, 180$ and 270 degrees, and the whole group G is just H plus 2D translations. In this case, there are four orientation channels per feature, which undergo a cyclic permutation under rotation. In 3D, however, things get substantially more complicated, as the filters in a three-dimensional CNN are not squares, but cubes or rectangular cuboids.We would restrict our study of symmetry groups to cubes, but in many 3D imaging modalities such as CT and MRI, the pixel spacing in the x and y directions can be different from the spacing in the z direction, so that a $k \times k \times k$ filter corresponds to a spatial region that is not a cube but a cuboid with a square base. In addition to the cube / rectangular cuboid choice, there is the choice of whether to consider only rotations (orientation-preserving symmetries) or reflections as well. Thus, we end up with four symmetry groups of interest: the orientation-preserving and non-orientation preserving symmetries of a rectangular cuboid (called D_4 and D_{4h}, resp.), and the orientation-preserving and non-orientation-preserving symmetries of a cube (O and O_h, resp.)

Despite the apparent simplicity of cubes and rectangular cuboids, their symmetry groups are surprisingly intricate. In figure 1, we show the *Cayley diagram* for the groups D_4 and D_{4h}.[2] In a Cayley diagram, each node corresponds to a symmetry transformation $h \in H$, here visualized by its effect on a canonical $3 \times 3 \times 3$ filter. The diagrams also have lines and arrows of various colors that connect the nodes. Each color corresponds to applying a particular *generator* transformation. By applying generators in sequence (i.e. following the edges of the diagram) we can make any transformation in the group. Because different sequences of generator transformations can be the equal, there can be several paths between two nodes, leading to an intricate graph structure. Since every group element can be written as a composition of generators, we can obtain its permutation

[2] A similar Cayley diagram can be constructed for O and O_h.

matrix from the permutations associated with the generators. The derivation of these permutations only has to be done once when implementing 3D G-CNNs; when using them, this complexity is hidden by an easy to use function `GConv3D()`.

When calling `GConv3D()` on the input (e.g. CT scans), the input feature maps and filters do not have orientation channels. When a filter with n_0 channels (number of input channels) is transformed by $h \in H$, each of the channels is transformed by h simultaneously, resulting in a single transformed filter with n_0 channels. This happens for every filter and each $h \in H$, leading to a bigger filter bank with $n_1 \cdot |H|$ filters (each of which still has n_0 input channels). Because the output feature maps of a G-Conv layer have orientation channels, the filters in the second and higher layers will also need orientation channels to match those of the input. These orientation channels of the filter need to be shuffled by $\rho(h)$, where the permutation matrix $\rho(h)$ is dependent on $h \in H$. If the input to layer l has n_l feature channels, each of which has $|H|$ orientation channels (for a total of $n_l \cdot |H|$ 3D channels), each of the n_{l+1} filters will also have n_l feature channels with $|H|$ orientation channels each. The filters again get transformed by each element $h \in H$, so that we end up with $n_{l+1} \cdot |H|$ transformed filters, and equally many 3D output channels. When applying a transformation $h \in H$ to a filter that has orientation channels, we must also shuffle the orientation channels of the filter. Doing so, the feature maps of the next layer will again have orientation channels that jointly transform equivariantly with the input of the network, so we can stack as many of these layers as we like while maintaining equivariance of the network.

3 Experiments

The experiments in this work will focus on the task of false positive reduction for pulmonary nodule detection, which is a relatively straightforward classification problem on 3D patches extracted from volumetric CT chest images where a candidate patch is deemed either *nodule* or *non-nodule*. The train, validation and test sets consist of $12 \times 72 \times 72$ sized patches around a nodule candidate center (normalised such that each voxel represents $1.25 \times .5 \times .5$mm of lung tissue) which are extracted from scans from the public NLST dataset (for training and validation) and LIDC/IDRI dataset (for testing) respectively. A total of $30,000$ data samples are available for training, $8,889$ for validation, and $8,582$ for testing.

The performance of networks with G-Convs for various 3D groups G are compared to a baseline network with regular 3D convolutions. We conduct this experiment for various training dataset sizes varying from 30 to 30,000 data samples. Each training set is balanced, and each smaller training set is a subset of all larger training sets.

For evaluation, we use various probability thresholds to determine the sensitivity at seven predefined average false positives per scan rates ($\frac{1}{8}; \frac{1}{4}; \frac{1}{2}; 1; 2; 4;$ and 8). These sensitivities are averaged to compute an overall system score. In addition, we evaluated the convergence speed of G-CNNs and regular CNNs, and found that the former converge substantially faster.

3.1 Network architectures & training procedure

A baseline network was established with 6 convolutional layers consisting of $3 \times 3 \times 3$ convolutions, batch normalization and ReLU nonlinearities. In addition, the network uses 3D max pooling with same padding after the first, third and fifth layer, dropout after the second and fourth layer, and has a fully-connected layer as a last layer. The baseline, when trained on the whole dataset, was found to achieve competitive performance based on the LUNA16 grand challenge leader board, and therefore deemed sufficiently representative of a modern pulmonary nodule detection system.

The G-Conv variants of the baseline were created by simply replacing the 3D convolution in the baseline with a G-Conv for the group D_4, D_{4h}, O or O_h. As this leads to an increase in number of 3D channels and number of parameters per filter, the number of desired output channels (n_{l+1}) is divided by $\sqrt{|H|}$ to keep the number of parameters roughly the same and the network comparable to the baseline.

We minimize the cross-entropy loss using the Adam optimizer [11]. The weights were initialized using the uniform Xavier method [12]. For training, we use a mini-batch size of 30 (the size of the smallest training set) for all training set sizes. We use validation-based early stopping. A single data augmentation scheme (continuous rotation by $0 - 360^o$, reflection over all axes, small translations

over all axes, scaling between $.8 - 1.2$, added noise, value remapping) was used for all training runs and all architectures.

3.2 Results

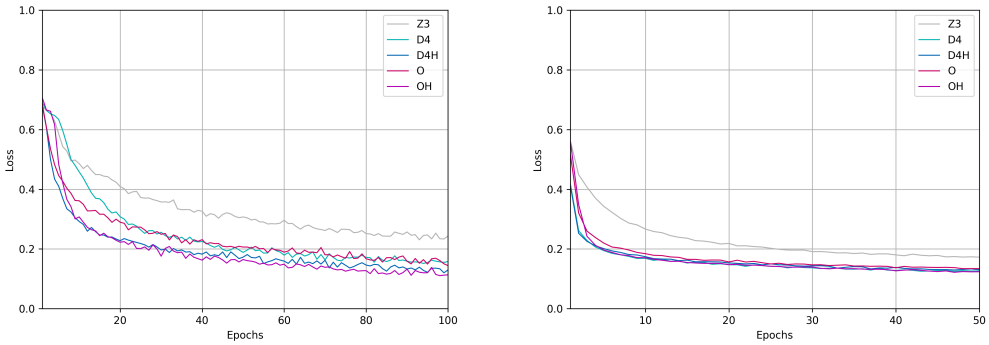
Table 1 contains the overall system score for each group and training set size combination, and has the highest (**bold**) and lowest (*italic*) scoring model per training set size highlighted. Figure 2 plots the training loss per epoch, for training runs with dataset size 3,000 and 30,000. Table 2 lists the number of training epochs required by each network to achieve a validation loss that was at least as good as the best validation loss achieved by the baseline.

N	$\mathbb{Z}^3$	D_4	D_{4h}	O	O_h
30	*0.252*	0.398	0.382	**0.562**	0.514
300	*0.550*	0.765	0.759	**0.767**	0.733
3,000	*0.791*	0.849	0.844	0.830	**0.850**
30,000	*0.843*	0.867	**0.880**	0.873	0.869

Table 1: Overall score for all training set sizes N and transformation groups G. The group $G = \mathbb{Z}^3$ corresponds to the standard translational CNN baseline.

N	$\mathbb{Z}^3$	D_4	D_{4h}	O	O_h	total epochs
3,000	*82*	33	22	21	**11**	*100*
30,000	*41*	4	9	7	**3**	*50*

Table 2: Number of epochs after which the loss is equal to or lower than the lowest validation loss achieved on the baseline for each group.



(a) Train loss on 3,000 samples

(b) Train loss on 30,000 samples

Figure 2: Learning curves for all networks trained on 3,000 and 30,000 samples.

4 Conclusion

In this work we have presented 3D Group-equivariant Convolutional Neural Networks (G-CNNs), and applied them to the problem of false positive reduction for pulmonary nodule detection. 3D G-CNN architectures – obtained by simply replacing convolutions by group convolutions – unambiguously outperformed the baseline CNN on this task, especially on small datasets, without any further tuning. The aggregated system scores show that not only do *all* G-CNNs outperform the baseline when trained on the same training set, they also regularly outperform the baseline trained on $10\times$ the data. Moreover, group-convolutional models show a faster decline in training loss within the early stages of training compared to the baseline and typically take only a few epochs to reach a validation loss that is better than the best validation loss achieved by the baseline. This improvement in statistical efficiency corresponds to a major reduction in cost of data collection, and brings pulmonary nodule detection and other computer-aided diagnosis systems closer to reality.

References

[1] T.S. Cohen; M. Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pages 2990–2999, 2016.

[2] T.S. Cohen; M. Welling. Steerable CNNs. In *International Conference on Learning Representations*, 2017.

[3] R. Kondor; S. Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. 2018.

[4] T. S. Cohen; M. Geiger; M. Weiler. Intertwiners between induced representations (with applications to the theory of equivariant neural networks). 2018.

[5] S. Dieleman; J. D. Fauw; K. Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. *International Conference on Machine Learning*, page 1889–1898, June 2016.

[6] D. E. Worrall; S. J. Garbin; D. Turmukhambetov; G. J. Brostow. Harmonic networks: Deep translation and rotation equivariance. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[7] M. Weiler; F. A. Hamprecht; M. Storath. Learning steerable filters for rotation equivariant CNNs . In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] E.J. Bekkers; M.W. Lafarge; M. Veta; K.A.J. Eppenhof; J.P.W. Pluim. Roto-Translation covariant convolutional networks for medical image analysis. April 2018.

[9] T.S. Cohen; M. Geiger; J. Koehler; M. Welling. Spherical CNNs. In *International Conference on Learning Representations (ICLR)*, 2018.

[10] R. Kondor; H.T. Son; H. Pan; B. Anderson; S. Trivedi. Covariant compositional networks for learning graphs. January 2018.

[11] D.P. Kingma; J. Ba. Adam: A method for stochastic optimization. *CoRR*, 2014.

[12] X. Glorot; Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.