Group Equivariant Convolutional Networks

Daniel Ralston

1/22/2024

Agenda

- Introduction and Motivation
- Related Works
- Background: Groups, Group Actions, and Equivariance
- Methods: G Equivariant Convolution
- Results
- Demonstration of the Code
- Conclusion

Introduction

Related Works

- Invariant representations improve image analysis, achieved by pose normalization and group averaging (Lowe, 2004; Reisert, 2008; Kondor, 2007) and extended by scattering convolution networks (Bruna and Mallat, 2013).
- Scattering networks, which use wavelet convolutions and nonlinearities, have been adapted for stable invariants in object and texture recognition (Sifre and Mallat, 2013; Oyallon and Mallat, 2015).
- Equivariant representations have been advanced through architectures like transforming autoencoders (Hinton et al., 2011) and convolutional architectures (Gens and Domingos, 2014) with unconventional learning applications (Agrawal et al., 2015).
- High-dimensional transformations and symmetry exploitation in convolutional networks have been utilized for tasks like galaxy morphology prediction, enhancing vision-related models (Dieleman et al., 2015; 2016; Cohen and Welling, 2014; 2015).

Groups

- A *group* is a set G with a binary operation $\cdot$ such that:
  1. G is closed under $\cdot$
  2. $\cdot$ is associative
  3. There exists an identity element $e \in G$ such that $e \cdot g = g \cdot e = g$ for all $g \in G$
  4. For each $g \in G$, there exists an inverse $g^{-1} \in G$ such that $g \cdot g^{-1} = g^{-1} \cdot g = e$
- In this paper, the authors focus on groups of rigid transformations of the plane (e.g. subgroups of $SE(2)$, the 2-dimensional special Euclidean group)

$p4$ and $p4m$

- $p4$ – all 2-dimensional integer translations and rotations by multiples of $\frac{\pi}{2}$
  - The underlying set can be described as a set of matrices where $r \in \{0, 1, 2, 3\}$ and $u, v \in \mathbb{Z}$

$$g(r, u, v) = \begin{bmatrix} \cos(\frac{\pi}{2}r) & -\sin(\frac{\pi}{2}r) & u \\ \sin(\frac{\pi}{2}r) & \cos(\frac{\pi}{2}r) & v \\ 0 & 0 & 1 \end{bmatrix}$$

- $p4m$ – all 2-dimensional integer translations, rotations by multiples of $\frac{\pi}{2}$, and mirror reflections
  - For $r \in \{0, 1, 2, 3\}$, $u, v \in \mathbb{Z}$, and $m \in \{0, 1\}$

$$g(m, r, u, v) = \begin{bmatrix} (-1)^m \cos(\frac{\pi}{2}r) & (-1)^{m+1} \sin(\frac{\pi}{2}r) & u \\ \sin(\frac{\pi}{2}r) & \cos(\frac{\pi}{2}r) & v \\ 0 & 0 & 1 \end{bmatrix}$$

- In both cases, the binary operation is matrix multiplication

Group Actions

- Critical to this paper, the authors use the fact that these groups act on the set of images.
- A group G is said to act on a set X if there exists a function $\gamma : G \times X \to X$ such that

$$\gamma(e, x) = x \ (e \text{ is the identity element of } G)$$

$$\gamma(g_1, \gamma(g_2, x)) = \gamma(g_1 g_2, x)$$

- $p4$ and $p4m$ act on $\mathbb{Z}^2$ (specifically $\mathbb{Z}^2 \times \{1\} \subset \mathbb{R}^3$) by matrix-vector multiplication:
  - Ex: For $A \in p4$ and $[u, v, 1]^T \in \mathbb{Z}^2 \times \{1\}$,

$$A[u, v, 1]^T = [u', v', 1]^T \in \mathbb{Z}^2 \times \{1\}$$

Acting on the set of images

- The authors describe the set of images as the collection of functions $f : \mathbb{Z}^2 \to \mathbb{R}^K$ (where f has compact (rectangular) support)
- $p4$ and $p4m$ act on the the set of images $\{f\}$ with the function L:

$$L_g(f)(x) = f(g^{-1}x)$$

where $g^{-1}x$ denotes matrix-vector multiplication

Results – Comparison

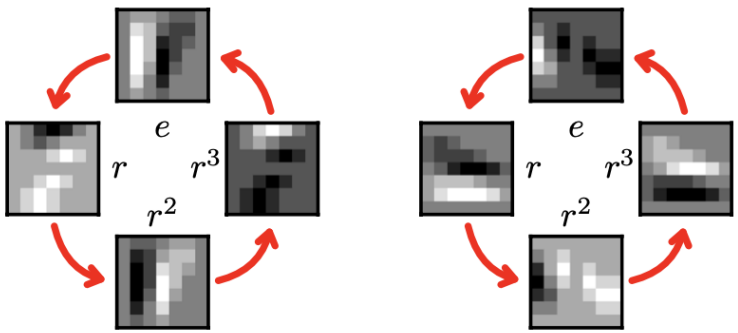


Figure 1. A p4 feature map and its rotation by r.

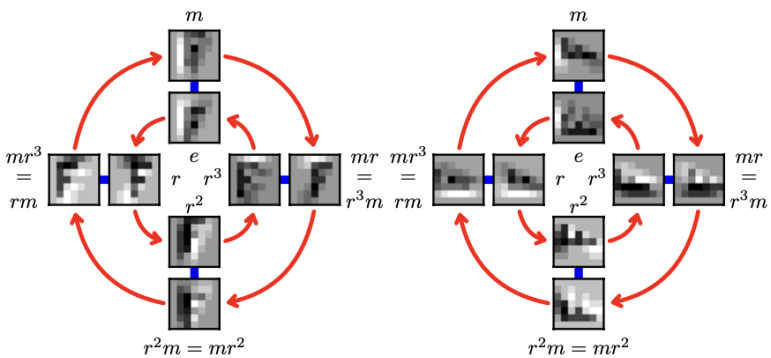


Figure 2. A p4m feature map and its rotation by r.

Equivariance

- Let G be a group acting on X and Y. A function $f : X \to Y$ is called *equivariant* if

$$f(gx) = gf(x) \text{ for all } g \text{ in } G \text{ and } x \text{ in } X$$

- Fix a kernel Ψ and denote $*$ as the convolution operator. Suppose G is the group of integer translations acting by T on the set of images.

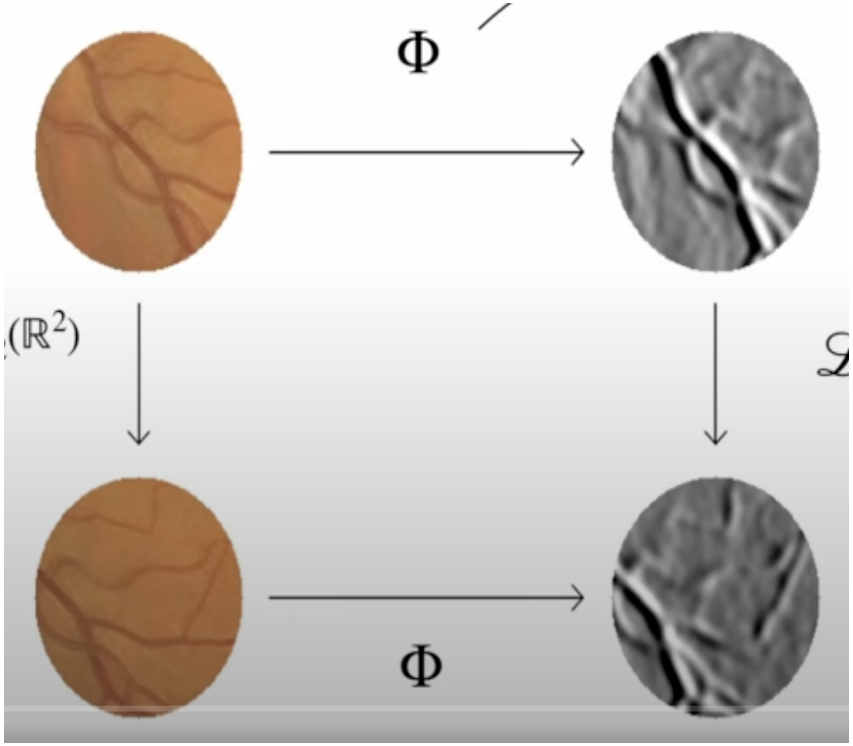
$$T(g, f)(x) = f(-g + x)$$

- For any image $f \in \{f\}$, the convolution operator $(\cdot) * \Psi : \{f\} \to \{f\}$ is equivariant with respect to translation:

$$(T(g, f) * \Psi)(x) = T(g, (f * \Psi))(x)$$

(i.e. you can translate an image and then convolve it or you can first convolve the image and then translate the output)

Translation Equivariance



Key Idea

- The authors want a convolution operator that is equivariant with respect to the group action of $p4$ or $p4m$
- We want a convolution where inputting a rotated image is the same as inputting a non-rotated image, convolving, and then rotating the output

Group Equivariant Convolution

- Let $f : \mathbb{Z}^2 \to \mathbb{R}^K$ be an image.
- Let G be either $p4$ or $p4m$ and let Ψ be a kernel. Then the *group equivariant convolution* $*_G$ is defined as

$$[f *_G \Psi](g) = \sum_{y \in \mathbb{Z}^2} \sum_{k \in \{1,\dots,K\}} f_k(y) \Psi_k(g^{-1}y)$$

$$\left(= \sum_{y \in \mathbb{Z}^2} \sum_{k \in \{1,\dots,K\}} f_k(gy) \Psi_k(y) \right)$$

Group Equivariant Convolution

- Equivariance property – denoting L_g be the action of $g \in G$ on the set of images,

$$L_h(f *_G \Psi)(g) = L_h \left(\sum_{y \in \mathbb{Z}^2} \sum_{k \in \{1,\dots,K\}} f_k(y) \Psi_k(g^{-1}y) \right)$$

$$= \sum_{y \in \mathbb{Z}^2} \sum_{k \in \{1,\dots,K\}} f_k(hy) \Psi_k(g^{-1}y)$$

$$= [L_h(f) *_G \Psi](g) \qquad \text{(note equivariance)}$$

$$= \sum_{y \in \mathbb{Z}^2} \sum_{k \in \{1,\dots,K\}} f_k(y) \Psi_k(h^{-1}g^{-1}y)$$

$$= [f *_G L_{h^{-1}}(\Psi)](g)$$

A Subtlety

- After the first layer of a $*_G$ convolution on an image f:

$$[f *_G \Psi](g)$$

  letting g vary, our output is a set of images $\{[f *_G \Psi](g)\}_{g \in G}$ that now also depends on G
- Thus, our filter needs to take as inputs not elements in $\mathbb{Z}^2$ (translations), but also elements in G (translations, rotations, reflections)

Inner G-CNN Convolutions

- For inner layers the convolutional operator $*_G$ is defined as

$$[f *_G \Psi](g) = \sum_{h \in G} \sum_{k \in \{1,\dots,K\}} f_k(h) \Psi_k(g^{-1}h)$$

- Equivariance property follows similarly as before:

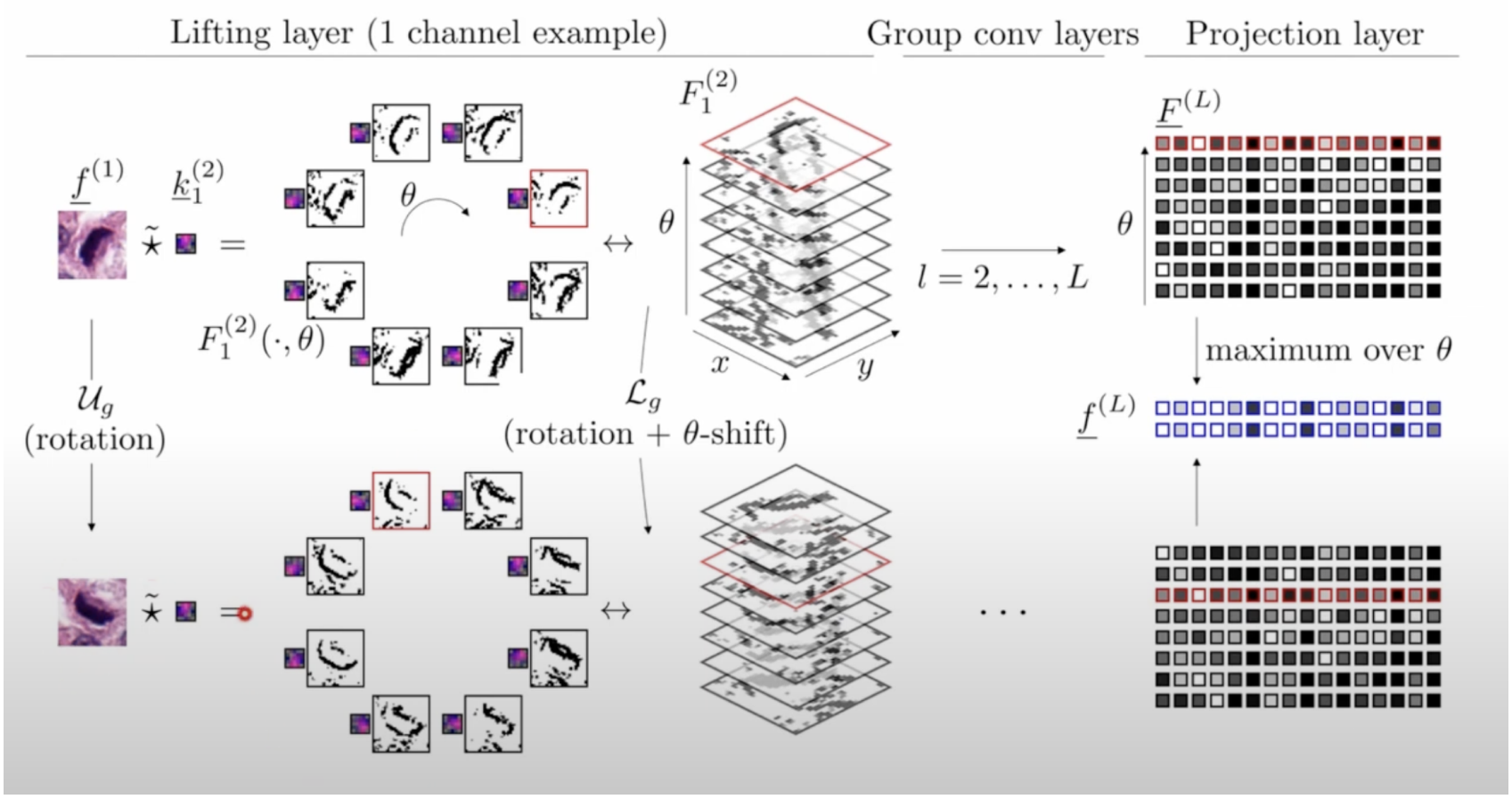
$$L_u([f *_G \Psi])(g) = L_u \left(\sum_{h \in G} \sum_{k \in \{1,\dots,K\}} f_k(h) \Psi_k(g^{-1}h) \right)$$

$$= \sum_{h \in G} \sum_{k \in \{1,\dots,K\}} f_k(uh) \Psi_k(g^{-1}h)$$

$$= [L_u(f) *_G \Psi](g)$$

$$= [f *_G L_{u^{-1}}(\Psi)](g)$$

G-CNN Visualization



Activations and Pooling

- Activations commute with the group action L on $\{f\}$:

$$L_g(\sigma(f(x))) = \sigma(f(g^{-1}x)) = \sigma(L_g(f)(x))$$

- Suppose P is a pooling operator over a region, ex max pooling over a sub region of G say H,

$$P(f)(g) = \max_{h \in H} f(hg)$$

- The authors show pooling commutes with the group action on an image, i.e. $L_g(P(f)) = P(L_g(f))$
- The authors propose H be a subgroup of G so that pooling partitions the set of images $\{[f *_G \Psi](g)\}_{g \in G}$ into cosets (equivalence classes)
- Ex if $G = p4$ and H is the subgroup of rotations, then pooling on $\{[f *_G \Psi](g)\}_{g \in G}$ reduces the output to be a function on $p4/H \cong \mathbb{Z}^2$

Results – Rotated MNIST

- Rotated MNIST dataset – 62,000 randomly rotated handwritten digits
- Model selection on the validation set led to a CNN architecture (Z2CNN) with 7 layers, relu activation, batch normalization, and dropout, optimized with Adam, outperforming early models but not state of the art.
- However the introduction of p4-convolutions and pooling over rotations in the final layer, paired with a parameter-adjusted architecture (P4CNN), nearly halved the state-of-the-art error rate (2.28% vs 3.98% error).
- A modified Z2CNN with p4-convolutions and coset max-pooling (P4CNNRotationPooling) showed improved performance over the baseline, though it was less effective than P4CNN without intermediate rotation pooling.

Results – CIFAR-10 and CIFAR-10+

- CIFAR-10 dataset – 60k 32x32 images across 10 classes, with 40k for training, 10k for validation, and 10k for testing.
- CIFAR-10+ dataset – CIFAR-10 augmented with flips and translations
- Authors tested architectures All-CNN-C (Springenberg et al, 2015) and ResNet44 (He et al, 2016), with adaptations using p4 and p4m convolutions to maintain parameter counts while expanding internal representations.
- The p4m-CNN outperformed all published results on unmodified CIFAR10, although direct comparisons are challenging due to architectural differences

Results – Comparison

Network	Test Error (%)
Larochelle et al. (2007)	10.38 ± 0.27
Sohn & Lee (2012)	4.2
Schmidt & Roth (2012)	3.98
Z2CNN	5.03 ± 0.0020
P4CNNRotationPooling	3.21 ± 0.0012
P4CNN	**2.28** ± **0.0004**

Table 1. Error rates on rotated MNIST (with standard deviation under variation of the random seed).

Network	G	CIFAR10	CIFAR10+	Param.
All-CNN	$\mathbb{Z}^2$	9.44	8.86	1.37M
	$p4$	8.84	7.67	1.37M
	$p4m$	7.59	7.04	1.22M
ResNet44	$\mathbb{Z}^2$	9.45	5.61	2.64M
	$p4m$	**6.46**	**4.94**	2.62M

Table 2. Comparison of conventional (i.e. $\mathbb{Z}^2$), $p4$ and $p4m$ CNNs on CIFAR10 and augmented CIFAR10+. Test set error rates and number of parameters are reported.

Implementation

Conclusion

- $p4$ and $p4m$ convolution layers act as effective replacements for traditional convolutions, consistently improving network performance.
- Future work includes extending G-CNNs to hexagonal lattices and 3D space groups, expanding symmetries and potentially improving pattern recognition capabilities.
- Challenges remain in extending the method to continuous groups and managing the computational load for large transformation groups.
- This work exemplifies the "structured representations" philosophy, suggesting that adding mathematical structure to neural network representations could reveal deeper similarities between concepts.

References