Experimental Exploration of the Power of Group Equivariant Convolutional Networks

July 30, 2023

Jessica Frabotta

Abstract

Group Equivariant Convolutional Networks (G-CNNs) exploit data symmetry to enhance CNN performance, overcoming the limitations of traditional CNNs in effectively handling and leveraging symmetries using group theory. The objective of this work is to replicate and expand upon the methodology presented in the paper "Group Equivariant Convolutional Networks" (Taco S. Cohen, 2016), which first introduced G-CNNs to the world. Additionally, the project focuses on evaluating the effectiveness of G-CNNs in improving robustness against adversarial attacks. All code is publicly available at https://github.com/JessicaFrabotta/DLAI_project

1. Introduction

CNNs are famously equivariant with respect to translation. Arguably, this property played a pivotal role in the advent of deep learning, reducing the number of trainable parameters by orders of magnitude. However, simple CNN are not equivariant to rotation and reflection operations. But what exactly is equivariance, and why is it so important? Let's recap some important definitions that will help in understanding how G-CNNs work and why they are so powerful.

Equivariance Equivariance ensures that if a neural network maps elements from set *X* to set *Y*, applying a transformation *T* to the input data first and then passing it through the model *F* gives the same result as passing the original input through *F* and then applying *T* to the output.

Group theory A group *G* is a set with an operation $*$ satisfying closure, associativity, an identity element, and inverse elements.

Email: Jessica Frabotta <frabotta.1758527@studenti.uniroma1.it>.

Deep Learning and Applied AI 2023, Sapienza University of Rome, 2nd semester a.y. 2022/2023.

The groups we will use in this work are *p4* and *p4m*. The group *p4* includes rotations about the origin by 90 degrees, as well as all the elements of *T* the group of all translations of $\mathbb{Z}^2$. If we add a reflection operation to p4 (ensuring closure by including all compositions), we obtain the group p4m.

2. Methods

We are now finally ready to define group convolution and describe how to implement G-CNNs.

Let *X* be the space of grayscale images on a grid. Group convolution is the sum of the product of a filter ψ with a translated input image x, considering various transformations on the group G. The result is an output function over G, denoted as Y, with a natural G action on it. Utilizing a general filter ψ in group convolution extends the input space of grayscale images X to a larger space Y. This operation is sometimes also called *lifting convolution*. However, we cannot directly stack multiple layers to create a deep network because the output space of the layer is different from the input space. To address this issue, we define group convolution, which allows to perform convolution on functions over the group's elements (functions in *Y*, the space of functions over the group *G*). By using the group convolution with a filter ψ in *Y*, we can map *Y* to *Y*, enabling the stacking of multiple layers for building a deep model.

3. Experiments

All experiments were performed using Google Colab, and the architectures were implemented in PyTorch. In this section, I will just describe the benchmark CNN architectures. The corresponding p4- and p4m-networks are obtained by halving the number of filters in each p4-conv layer and dividing it by approximately $\sqrt{8} \approx 3$ in each p4m-conv layer to achieve almost the same number of parameters and ensure a fair comparison. Even when not explicitly stated, the metric used to report the results is test accuracy.

Rotated MNIST The rotated MNIST dataset contains 62k randomly rotated handwritten digits. The benchmark architecture consists of 7 layers of 3 × 3 convolutions (except for the final layer, which has 4 × 4 convolutions) with 20 channels in each layer. It uses ReLU activation functions, batch normalization, dropout, and max-pooling after layer 2.

Network	Test accuracy	Parameters
Z2CNN	85.47%	28070
P4CNN	90.35%	29790

CIFAR-10 The CIFAR-10 dataset consists of 60k RGB images of size 32 × 32, divided into 10 classes. Two kinds of baseline architectures were compared: All-CNN-C (Jost Tobias Springenberg, 2014) and ResNet44 (Kaiming He, 2015)

To evaluate the impact of data augmentation, I compared the networks on CIFAR10 and augmented CIFAR10+. CIFAR10+ differs from vanilla CIFAR10 for moderate data augmentation with horizontal flips and small translations.

Network	CIFAR10	CIFAR10+	Parameters
AllCNNC	75.07%	74.98%	1372254
P4AllCNNC	75.95%	77.5%	1371006
P4MAllCNNC	76.96%	77.2%	1219678
ResNet44	73.45%	75.08%	2631962
P4MResNet44	86.49%	87.92%	2619864

Plant Leaves The Plant Leaves dataset consists of 12 classes and contains approximately 13k RGB images. I chose a leaf classification dataset because leaves exhibit various symmetries. Two baseline architectures, ResNet44 and MyCNN, along with their equivariant counterparts, were trained and tested. MyCNN consists of four sets of convolutional layers with varying channel numbers (8, 32, 64, and 128), utilizing 3x3 filters, and applying batch normalization, ReLU activation, and 0.3 dropout rate after each layer.

Network	Test accuracy (%)	Parameters
ResNet44	89.34%	2631962
P4MResNet44	95.30%	2619864
MyCNN	96.11%	403340
MyP4CNN	98.55%	398780

4. Adversarial attacks

While working on this project, I began to question whether Group Equivariant Convolutional Networks could provide greater robustness against adversarial attacks when compared to traditional CNNs. However, existing literature did not provide a definitive answer to my question. To satisfy this curiosity, I decided to conduct some experiments.

I used the Fast Gradient Sign Method (FGSM) (Ian J. Goodfellow, 2014), a white-box attack, to generate adversarial examples. FGSM involves adding a perturbation $\eta = \epsilon \cdot \text{sign}(\nabla_x J(w, x, y))$ to maximize the loss function, ultimately confusing the model and creating adversarial examples. In the formula ϵ is a hyperparameter, representing the magnitude of the perturbation. Tests were performed on all the datasets we saw so far. The FGSM adversarial attack is more effective on RGB images due to the higher dimensionality and richer color information so the values of epsilon tested on CIFAR-10 and Plant Leaves are smaller than the ones used for Rotated MNIST.

Rotated MNIST

Epsilon	Z2CNN	P4CNN
0.05	69.9%	76.7%
0.1	42.6%	48.0%
0.15	22.3%	25.8%
0.2	14.9%	15.6%

CIFAR-10 and Plant Leaves In these images even very small perturbations generated with epsilon equal to 0.1 result in the generation of images that are severely distorted and almost unrecognizable, even to humans.

Eps	AllCNNC	P4MAllCNNC	MyCNN	MyP4CNN
0.001	72.8%	74.9%	69.1%	92.3%
0.01	27.7%	24.5%	38.4%	55.3%
0.1	0.68%	0.88%	0.44%	0.35%

5. Discussion and conclusion

During the experiments, the Group Equivariant Convolutional Networks (G-CNNs) consistently outperformed their regular CNN counterparts. Data augmentation benefits G-CNNs just like regular convolutional networks, and perhaps even more. The project's findings showed that using group convolutions is a promising way to boost the performance of convolutional neural networks while keeping their size manageable. In addition, the experiments demonstrated that Group Equivariant Convolutional Networks are generally more robust against adversarial attacks compared to simple CNNs. This is likely because they take into account the symmetries and invariances present in the data, making them less vulnerable to perturbations. However, it's important to note that their effectiveness depends on the dataset and the type of attack used. My results are based on experiments using the Fast Gradient Sign Method (FGSM) as the only type of attack. In future works, I would like to test their robustness against other kinds of attacks. In conclusion, while not a one-size-fits-all solution, group equivariant networks show promise in improving the robustness of deep learning systems against adversarial attacks.

References

Ian J. Goodfellow, Jonathon Shlens, C. S. Explaining and harnessing adversarial examples. 2014. URL https://arxiv.org/abs/1412.6572.

Jost Tobias Springenberg, Alexey Dosovitskiy, T. B. M. R. Striving for simplicity: The all convolutional nets. 2014. URL https://arxiv.org/abs/1412.6806.

Kaiming He, Xiangyu Zhang, S. R. J. S. Deep residual learning for image recognition. 2015. URL https://arxiv.org/abs/1512.03385.

Taco S. Cohen, M. W. Group equivariant convolutional networks. 2016. URL https://arxiv.org/abs/1602.07576.