Group-Equivariant Neural Networks in Computer Vision

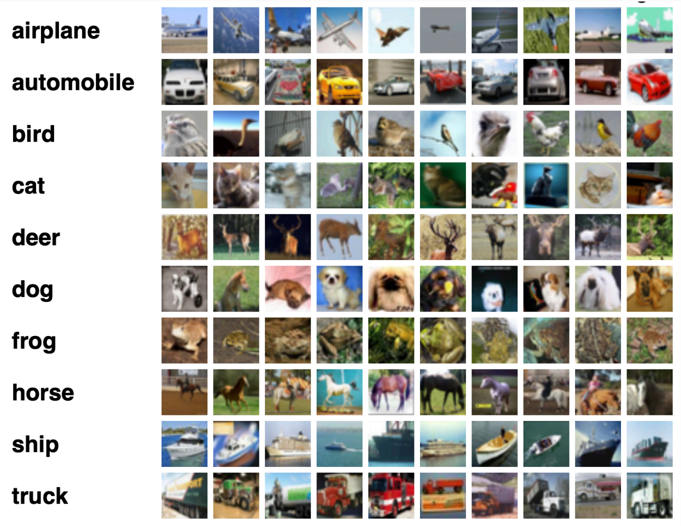
Wern Juin Gabriel Ong

Max Planck Institute for Mathematics in the Sciences & Bowdoin College

Table of contents

Convolutions and Computer Vision

A Task in Computer Vision



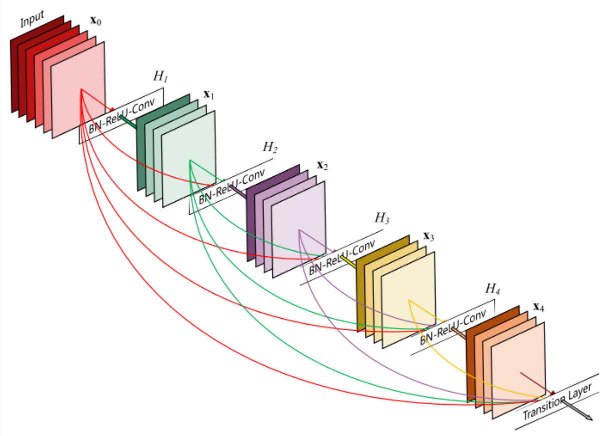
A Task in Computer Vision – Restated

Our eyes and brain together form a function $\mathbb{R}^{32\times32\times3} \to [10]$ taking an image in CIFAR-10 to its class.

Question

Can we find an approximator (perhaps a deep neural network $\mathcal{N}$) $\mathbb{R}^{32\times32\times3} \to [10]$ such that it correctly takes an image in CIFAR-10 to its class?

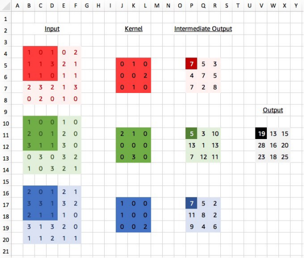
Convolutions in Computer Vision

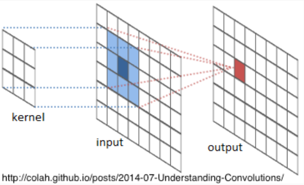


Residual networks, that use convolutional layers, achieve industry leading performance on the CIFAR-10 data set [3].

Figure from [2].

Convolutions in Computer Vision





Let us recall the mathematics behind convolutional layers [9].

- We have a function on our input space of images.
- We feed translated copies of an image to the function.
- We store the function output for each translation.

Equivariance of Functions

Convolutional Layers as Equivariant

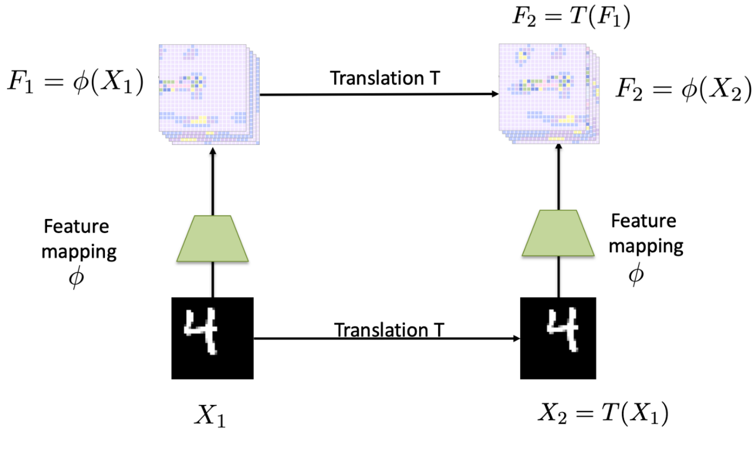


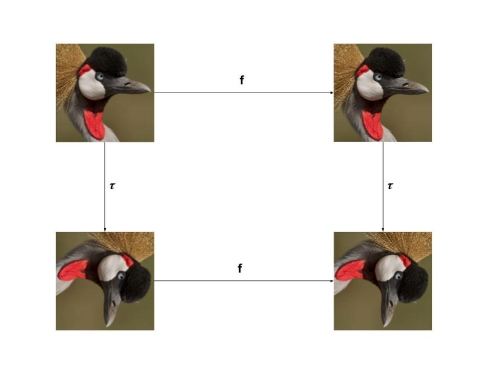
Figure from [10].

Equivariance of Functions

Definition (Equivariance) [7]

Let G be a finite group and $f : \mathbb{R}^n \to \mathbb{R}^m$ where $G \circlearrowright \mathbb{R}^n$ and $G \circlearrowright \mathbb{R}^m$. Suppose that $\varphi : G \hookrightarrow S_n$ and $\psi : G \hookrightarrow S_m$ are given. f is G-equivariant if and only if $f(\varphi(g) \cdot \vec{x}) = \psi(g) \cdot f(\vec{x})$ for all $g \in G$ and any $\vec{x} \in \mathbb{R}^n$.

Equivariance of Functions



Example (Image Sharpening)

Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be a function that sharpens images and $\tau_\alpha : \mathbb{R}^n \to \mathbb{R}^n$ for $\{\tau_\alpha\}_{\alpha \in A}$ be the group of 90° rotations under composition. f is in fact invariant under the group $(\{\tau_\alpha\}_{\alpha \in A}, \circ)$.

Equivariance and CNNs

The translation equivariance of CNNs has been instrumental in enabling this performance [over traditional deep neural networks]. However, the vulnerability of CNNs to transformations such as rotation and scaling remains a challenge, since standard CNNs do not learn features that are equivariant with respect to these transformations.

Quote from [5].

Equivariance in Images

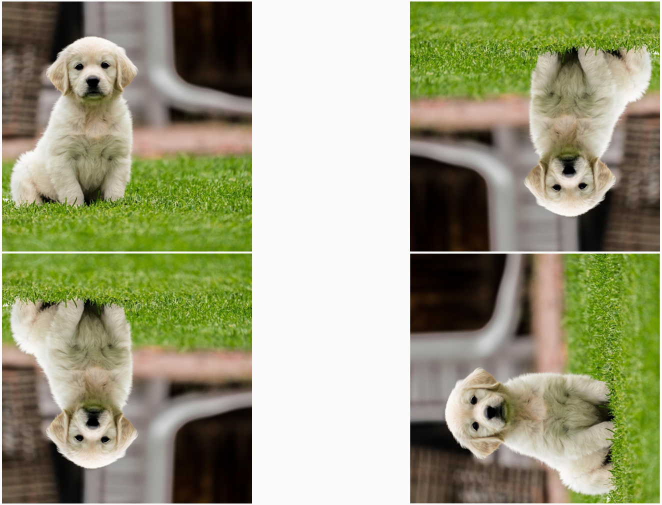


Figure from [8].

The Group $p4m$

The set of symmetries we are looking at is the wallpaper group $p4m$.

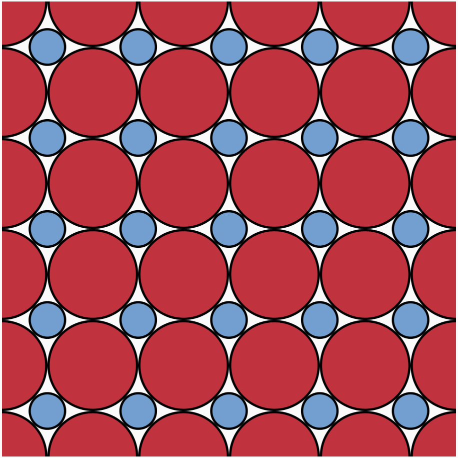


Figure from [4].

Equivariance in Images

We want to take advantage of the *p*4*m* structure of images in our neural network classifier.

Desideratum

If we have $\mathcal{N} : \mathbb{R}^{32\times 32\times 3} \to [10]$ a neural network classifier of CIFAR-10 images, we want our convolutional layers to be *p*4*m* equivariant.

Equivariance of Neural Networks

Equivariance and Convolution

Definition (G-Convolution)

Let $f, g : G \to \mathbb{C}$ and G a compact group. The convolution of f with g

$$(f * g)(u) = \int_G f(uv^{-1})g(v)d\mu(v)$$

where the integration is with respect to the Haar measure μ.

Theorem (Kondor and Trivedi) [6]

A neural network is equivariant to the action of a compact group G if and only if each layer of the neural network is a G-convolution.

The proof is quite technical, requiring knowledge of Fourier analysis on compact groups as well as representation theory.

Universal Approximation in Neural Networks

Denote $\sum^n(\sigma)$ the space of neural networks from $\mathbb{R}^n \to \mathbb{R}$ for σ an activation function.

Theorem (Cybenko)

Let $K \subset \mathbb{R}^n$ be compact and $f : K \to \mathbb{R}$ continuous. There exists $\mathcal{N} \in \sum^n(\sigma)$ such that

$$\|\mathcal{N}(\vec{x}) - f(\vec{x})\|_\infty < \epsilon$$

for all $\epsilon \in \mathbb{R}_{>0}$ and $\vec{x} \in K$.

This generalizes easily to find neural network approximations $\mathcal{N} \in \sum^{n,m}(\sigma)$ approximating $f : K \to \mathbb{R}^m$.

Appproximation of Equivariant Functions?

Question

Let f be an equivariant function. Does there exist $\mathcal{N} \in$ "Some Space of Neural Networks" such that $\|\mathcal{N} - f\|_\infty < \epsilon$?

A Theorem of Sannai, et. al.

Theorem (Sannai, et. al.) [7]

Let G be a finite group which is a subgroup of S_n. Let $K \subset \mathbb{R}^n$ be compact and stable for the corresponding G-action in $\mathbb{R}^n$. Then for any $f : K \to \mathbb{R}^m$ by $\vec{x} \mapsto (f_1(\vec{x}), \ldots, f_m(\vec{x}))$ which is continuous and G-equivariant and any $\epsilon \in \mathbb{R}_{>0}$ there exists some equivariant neural network $\mathcal{N}_G^{\mathrm{Equiv}} : K \to \mathbb{R}^m$ such that $\|f - \mathcal{N}_G^{\mathrm{Equiv}}\|_\infty < \epsilon$.

Proof

- Show that an S_n equivariant function $F : \mathbb{R}^n \to \mathbb{R}^m$ is actually $(f, f \circ (1\ 2), \ldots, f \circ (1\ n))$ for f a $\mathrm{Stab}(1)$ invariant function.
- So F is really the pointwise composition

$$F = (f, \ldots, f) \circ (\varepsilon, (1\ 2), \ldots, (1\ n)).$$

- We know that $(f, \ldots, f)$ can be approximated by a neural network – since invariant functions can be approximated by neural networks.

Proof

- The crux of the proof is showing $(\varepsilon, (1\,2), \dots, (1\,n)) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ can be approximated by neural networks as well.
- Construct a group action $\circlearrowleft: S_n \times \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n} \implies \sigma = \tilde{\sigma}_i(1\ \sigma_i^{-1}(i))$ where $\sigma \in S_n, \tilde{\sigma}_i \in \mathrm{Stab}(1)$ where $\tilde{\sigma}_i$ is unique.
- $(\varepsilon, (1\,2), \dots, (1\,n)) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$, with some algebra, can be approximated by a neural network with $\mathrm{Stab}(1)$ invariant layers and is equivariant under S_n. $\square$

Equivariant Neural Networks "In the Wild"

Previous results in the literature show the superior performance of group-equivariant CNNs in image classification [1].

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.

Which is behavior I have observed in my testing.

Metric	$p4m$-CNN	CNN
Cross-Entropy Loss	0.679	0.832
Classification Accuracy	77%	70%

Conclusion

Summary

Here are some takeaways:

- CNNs are equivariant under $\mathbb{Z}^2$, G-equivariant CNNs are equivariant under a finite group G.
- G-equivariant neural networks, like (convolutional) neural networks, are universal approximators.

Summary

Here are some takeaways:

- There are packages to implement G-convolutions in ***PyTorch*** and ***TensorFlow***.
- G-convolutions provide significant performance improvements over CNNs.
- G-equivariant neural networks are computationally expensive to implement ($\sim$ 8 times the memory).

Questions?

References i

- T. S. Cohen and M. Welling.
  Group equivariant convolutional networks.
  2016.
- V. Feng.
  Implicit equivariance in convolutional networks, 2017.
- K. He, X. Zhang, S. Ren, and J. Sun.
  Deep residual learning for image recognition, 2015.
- T. Kennedy.
  Compact packings of the plane with two sizes of discs.
  2004.
- N. Khetan, T. Arora, S. U. Rehman, and D. K. Gupta.
  Implicit equivariance in convolutional networks, 2021.

References ii

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

A. Sannai, Y. Takai, and M. Cordonnier.
Universal approximations of permutation invariant/equivariant functions by deep neural networks, 2019.

L. Veljanovski.
Golden retriever has ’puppy tantrum’ in adorable viral video.

E. Wagstaff and F. Fuchs.
Cnns and equivariance.

C. Wolf.
What is translation equivariance, and why do we use convolutions to get it?, 2020.