Spherical CNNs

Credit: Taco S. Cohen[1], Mario Geiger[2], Jonas Köhler[1], Max Welling[1,3]

[1]University of Amsterdam

[2]EPFL

[3]CIFAR

Presenter: Fuwen Tan
https://qdata.github.io/deep2Read

Outline

1. Background reading
2. Goals
   - CNNs on planar images $\rightarrow$ CNNs on spherical images
3. Equivariance properties
   - Planar CNN is translation-equivariant
   - Spherical CNN is rotation-equivariant
4. Implementation
5. Experiments
   - Equivariance error
   - Rotated MNIST on the sphere
   - Recognition of 3D shapes
   - Prediction of atomization energies from molecular geometry
6. Take-home messages

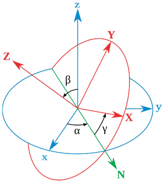
If you get confused

Group Equivariant Convolutional Networks [1]
T.S. Cohen, M. Welling
ICML, 2016.

Outline

Parameterization

$$\begin{aligned} Plane &: x(u,v) \in \mathbb{R}^2 \\ Sphere &: x(\alpha,\beta) = Z(\alpha)Y(\beta)n \in S^2 \\ n &: north\ pole \\ 3D\ Rotation &: R(\alpha,\beta,\gamma) = Z(\alpha)Y(\beta)Z(\gamma) \in SO(3) \end{aligned}$$



Group	Description	Dim.	Matrix Representation
SO(3)	3D Rotations	3	3D rotation matrix
SE(3)	3D Rigid transformations	6	Linear transformation on homogeneous 4-vectors
SO(2)	2D Rotations	1	2D rotation matrix
SE(2)	2D Rigid transformations	3	Linear transformation on homogeneous 3-vectors
Sim(3)	3D Similarity transformations (rigid motion + scale)	7	Linear transformation on homogeneous 4-vectors

Figure: Proper Euler angles geometrical definition. The xyz (fixed) system is shown in blue, the XYZ (rotated) system is shown in red. The line of nodes (N) is shown in green. Credit: `https://en.wikipedia.org/wiki/Euler_angles`

CNNs on planar images

$$
\begin{aligned}
(f * \psi)(x) &= \int_{\mathbb{R}^2} f(y)\psi(x-y)dy \\
f &: \mathbb{R}^2 \rightarrow \mathbb{R}\ (e.g.\ feature\ Maps) \\
\psi &: \mathbb{R}^2 \rightarrow \mathbb{R}\ (e.g.\ locally-supported\ filter)
\end{aligned}
$$

CNNs on planar images

$$
\begin{aligned}
(f * \psi)(x) &= \int_{\mathbb{R}^2} f(y)\psi(T_x^{-1}(y))dy \\
T_x(t) &= t + x \; (translation) \\
T_x^{-1}(t) &= x - t
\end{aligned}
$$

CNNs on spherical images (first layer)

$$
\begin{aligned}
(f * \psi)(x) &= \int_{S^2} f(y)\psi(R_x^{-1}(y))dy \\
x, y &: 3D\ unit\ vector \in S^2 \\
R_x(t) &= R_x \cdot t\ (3D rotation) \\
R_x &: (\alpha, \beta, \gamma) \in SO(3)
\end{aligned}
$$

A subtitle point

$$(f * \psi)(x) \;=\; \int_{S^2} f(y)\psi(R_x^{-1}(y))dy$$

- First layer:
  - Input: $S^2 \to 2D$.
  - Output: $SO(3) \to 3D$.
  - The output is indexed by an entry in $SO(3)$
- An extra dimension modeling the rotation
  - Movement over S^2: 2 dof
  - Rotation around the position x: 1 dof
  - Different from [2], which "restricts the filter to be circularly symmetric about the Z axis."

CNNs on spherical images (higher layers)

$$
\begin{aligned}
(f * g)(R) &= \int_{SO(3)} f(Q) g(R^{-1}(Q)) dQ \\
R, Q &: (\alpha, \beta, \gamma) \in SO(3)
\end{aligned}
$$

Outline

Transformation of the filter and the feature map

$$
\begin{aligned}
[L_g \psi](t) &= \psi(g^{-1}t) \\
[L_g f](t) &= f(g^{-1}t)
\end{aligned}
$$

Equivariance properties of CNNs

- $\phi(T_g x) = T'_g \phi(x)$.
  - transforming an input x by a transformation (e.g. translation) 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.
- Planar CNN is equivariant to translations.
  - $([L_T f] * \psi) = L_T(f * \psi)$
  - f: e.g. earlier CNN layers

Proof

- Planar CNN is equivariant to translations.
  - $T(t) = t + u; T^{-1}(x) = x - u$
  - $dT(t) = d(t + u) = dt$

$$
\begin{aligned}
L_T(f * \psi)(x) &= (f * \psi)(T^{-1}x) = (f * \psi)(x - u) \\
&= \int_{\mathbb{R}^2} f(y)\psi((x - u) - y)dy \\
&= \int_{\mathbb{R}^2} f(y)\psi(x - (u + y))dy \\
\{substitute : v = u + y\} &= \int_{\mathbb{R}^2} f(v - u)\psi(x - v)dv \\
&= \int_{\mathbb{R}^2} f(T^{-1}v))\psi(x - v)dv \\
&= \int_{\mathbb{R}^2} [L_T f](v))\psi(x - v)dv \\
&= ([L_T f] * \psi)(x)
\end{aligned}
$$

Outline

Spherical CNN is rotation-equivariant

- Spherical CNN is equivariant to rotations.
  - $([L_Q f] * \psi) = L_Q(f * \psi)$
  - Requirement:
    - dy is the invariant measure on S^2
    - dQ is the invariant measure on $SO(3)$
    - $dRy = dy; dRQ = dQ$
    - $\int_{S^2} \theta(Ry)dy = \int_{S^2} \theta(Ry)d(Ry) = \int_{S^2} \theta(y)dy$
    - guarantee by the parameterization (appendix A)

$$
\begin{aligned}
(f * \psi)(x) &= \int_{S^2} f(y)\psi(R_x^{-1}(y))dy \\
(f * \psi)(R) &= \int_{SO(3)} f(Q)\psi(R^{-1}(Q))dQ
\end{aligned}
$$

Proof (appendix B)

$$
\begin{aligned}
([L_Q f] * \psi)(R) &= \int_{S^2} f(Q^{-1}y)\psi(R^{-1}y)dy \\
\{substitute : y = Qt\} &= \int_{S^2} f(t)\psi(R^{-1}Qt)d(Qt) \\
&= \int_{S^2} f(t)\psi((Q^{-1}R)^{-1}t)d(t) \\
&= (f * \psi)(Q^{-1}R) \\
&= \big[L_Q(f * \psi)\big](R)
\end{aligned}
$$

GFFT

- GFFT defined on S^2 and $SO(3)$
- SO(3): Wigner D-function
- S^2: spherical harmonics

Outline

Equivariance error

$$\Delta = \frac{1}{n}\sum_{i=1}^{n}\frac{std(L_{R_i}(\Phi(f_i)) - \Phi(L_{R_i} f_i))}{std(\Phi(f_i))}$$

Φ : spherical CNN layers with randomly initialized filters

f_i, R_i := randomly chosen features (with channel K=10) and rotations

$n = 500$

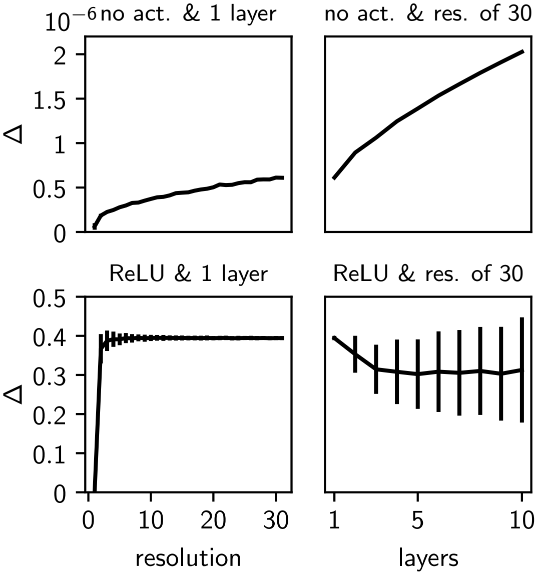


Figure: Δ as a function of the resolution and the number of layers.

Outline

1. Background reading
2. Goals
   - CNNs on planar images → CNNs on spherical images
3. Equivariance properties
   - Planar CNN is translation-equivariant
   - Spherical CNN is rotation-equivariant
4. Implementation
5. Experiments
   - Equivariance error
   - Rotated MNIST on the sphere
   - Recognition of 3D shapes
   - Prediction of atomization energies from molecular geometry
6. Take-home messages

MNIST on the sphere

- Dataset 1 (NR): projected on the northern hemisphere
- Dataset 2 (R): projected on the northern hemisphere and then randomly rotated
- Planar images for baseline methods:
  - stereographic projection

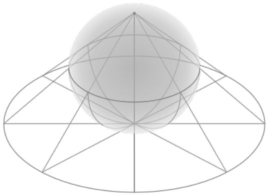


Figure: Stereographic projection.

Results

- Baseline: conv-ReLU-conv-ReLU-FC
  - kernel: 5 x 5
  - channels: 32, 64, 10
- Spherical CNN: S^2conv-ReLU-SO(3)conv-ReLU-FC
  - bandwidth: 30, 10, 6
  - channels: 20, 40, 10

	NR / NR	R / R	NR / R
planar	0.98	0.23	0.11
spherical	0.96	0.95	0.94

Table: Test accuracy for the networks evaluated on the spherical MNIST dataset. Here R = rotated, NR = non-rotated and X / Y denotes, that the network was trained on X and evaluated on Y.

Outline

3D recognition

- SHREC17 task [3]
  - Training data: 51300 non-aligned 3D models
  - Classification: 55 categories
- Representation
  - Ray casting on the surface and its convex hull
  - channels: 6 ((length, cos, sin) x 2)

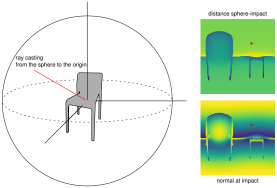


Figure: The ray is cast from the surface of the sphere towards the origin. The first intersection with the model gives the values of the signal. The two images of the right represent two spherical signals in (α, β) coordinates. They contain respectively the distance from the sphere and the cosine of the ray with the normal of the model. The red dot corresponds to the pixel set by the red line.

Results

- Model
  - S^2conv-BN-ReLU (50 features)
  - 2 x (SO(3)conv-BN-ReLU) (70/350 features)
  - max-pooling-BN-FC
  - bandwidths: 128, 32, 22, 7

Method	P@N	R@N	F1@N	mAP	NDCG
Tatsuma_ReVGG	0.705	0.769	0.719	0.696	0.783
Furuya_DLAN	0.814	0.683	0.706	0.656	0.754
SHREC16-Bai_GIFT	0.678	0.667	0.661	0.607	0.735
Deng_CM-VGG5-6DB	0.412	0.706	0.472	0.524	0.624
Ours	0.701 (3rd)	0.711 (2nd)	0.699 (3rd)	0.676 (2nd)	0.756 (2nd)

Table: Results and best competing methods for the SHREC17 competition.

Outline

Molecular energy regression

- QM7 task
  - Input: for each molecule, positions p_i and charges z_i of the atoms
  - $N = 23$ atoms of $T = 5$ types (H, C, N, O, S) for each molecule
  - Output: atomization energy of the molecule (scalar)

Representation as a spherical signal

- A sphere S_i around p_i
- Uniform radius such that no intersections among spheres
- For each possible z and for each point $x \in S^2$:
  - $U_z(x) = \sum_{j \neq i, z_j = z} \frac{z_i \cdot z}{|x - p_i|}$
  - For each atom: a T channel spherical function

Model

- ResNet block
  - $S^2SO(3)con - BN - ReLU - SO(3)conv - BN$
- Shared weights for all atoms: N x F feature maps
- To achieve permutation invariance:
  - Embedding: MLP ϕ
  - Sum pooling
  - Regression: MLP ψ

Results

Method	Author	RMSE
MLP / random CM	(a)	5.96
LGIKA(RF)	(b)	10.82
RBF kernels / random CM	(a)	11.40
RBF kernels / sorted CM	(a)	12.59
MLP / sorted CM	(a)	16.06
Ours		**8.47**

S^2CNN	Layer	Bandwidth	Features
	Input		5
	ResBlock	10	20
	ResBlock	8	40
	ResBlock	6	60
	ResBlock	4	80
	ResBlock	2	160
DeepSet	Layer	Input/Hidden	
	ϕ (MLP)	160/150	
	ψ (MLP)	100/50	

Table 3: Left: Experiment results for the QM7 task: (a) Montavon et al. (2012) (b) Raj et al. (2016). Right: ResNet architecture for the molecule task.

Take-home messages

- One of the best papers in ICLR 2018
- Potential applications on omnidirectional vision (e.g. for AR/VR)
- Potential extensions to more transformation groups (e.g. SE(3))

Taco Cohen and Max Welling.
Group equivariant convolutional networks.
In *International Conference on Machine Learning (ICML)*, pages 2990–2999. PMLR, 20–22 Jun 2016.

J. R. Driscoll and D. M. Healy.
Computing fourier transforms and convolutions on the 2-sphere.
Adv. Appl. Math., 15(2):202–250, June 1994.

M. Savva, F. Yu, Hao Su, M. Aono, B. Chen, D. Cohen-Or, W. Deng, Hang Su, S. Bai, X. Bai, N. Fish, J. Han, E. Kalogerakis, E. G. Learned-Miller, Y. Li, M. Liao, S. Maji, A. Tatsuma, Y. Wang, N. Zhang, and Z. Zhou.
Large-scale 3d shape retrieval from shapenet core55.
In *Proceedings of the Eurographics 2016 Workshop on 3D Object Retrieval*, 3DOR '16, pages 89–98, Goslar Germany, Germany, 2016. Eurographics Association.