Spherical CNNs

Credit: Taco S. Cohen\textsuperscript{1}, Mario Geiger\textsuperscript{2}, Jonas Köhler\textsuperscript{1}, Max Welling\textsuperscript{1,3}

\textsuperscript{1}University of Amsterdam

\textsuperscript{2}EPFL

\textsuperscript{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

Credit: Taco S. Cohen, Mario Geiger, Jonas Köhler, Max Welling (shortinst)

Presenter: Fuwen Tan  https://qdata.github.io/deep2Read
If you get confused

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

Credit: Taco S. Cohen, Mario Geiger, Jonas K, Max Welling (shortinst)
**Parameterization**

*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*: \( R(\alpha, \beta, \gamma) = Z(\alpha)Y(\beta)Z(\gamma) \in SO(3) \)

---

### Table: Groups and Transformations

<table>
<thead>
<tr>
<th>Group</th>
<th>Description</th>
<th>Dim.</th>
<th>Matrix Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SO(3)</td>
<td>3D Rotations</td>
<td>3</td>
<td>3D rotation matrix</td>
</tr>
<tr>
<td>SE(3)</td>
<td>3D Rigid transformations</td>
<td>6</td>
<td>Linear transformation on homogeneous 4-vectors</td>
</tr>
<tr>
<td>SO(2)</td>
<td>2D Rotations</td>
<td>1</td>
<td>2D rotation matrix</td>
</tr>
<tr>
<td>SE(2)</td>
<td>2D Rigid transformations</td>
<td>3</td>
<td>Linear transformation on homogeneous 3-vectors</td>
</tr>
<tr>
<td>Sim(3)</td>
<td>3D Similarity transformations</td>
<td>7</td>
<td>Linear transformation on homogeneous 4-vectors</td>
</tr>
</tbody>
</table>

*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](https://en.wikipedia.org/wiki/Euler_angles)
CNNs on planar images

\[(f \ast \psi)(x) = \int_{\mathbb{R}^2} f(y)\psi(x - y)dy\]

\[f : \mathbb{R}^2 \rightarrow \mathbb{R} \ (e.g. \ feature \ Maps)\]

\[\psi : \mathbb{R}^2 \rightarrow \mathbb{R} \ (e.g. \ locally - supported \ filter)\]
CNNs on planar images

\[(f \ast \psi)(x) = \int_{\mathbb{R}^2} f(y) \psi(T_x^{-1}(y)) dy\]

\[T_x(t) = t + x \text{ (translation)}\]

\[T_x^{-1}(t) = x - t\]
(f ∗ ψ)(x) = \int_{S^2} f(y)ψ(R_x^{-1}(y))dy

x, y : 3D unit vector ∈ S^2
R_x(t) = R_x \cdot t (3Drotation)
R_x : (α, β, γ) ∈ SO(3)
\[(f \ast \psi)(x) = \int_{S^2} f(y) \psi(R_x^{-1}(y)) dy\]

- First layer:
  - Input: \(S^2 \rightarrow 2D\).
  - Output: \(SO(3) \rightarrow 3D\).
  - 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."
(f \ast g)(R) = \int_{SO(3)} f(Q) g(R^{-1}(Q)) dQ

R, Q : (\alpha, \beta, \gamma) \in SO(3)
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
Transformation of the filter and the feature map

\[
\begin{align*}
[L_g \psi](t) &= \psi(g^{-1}t) \\
[L_g f](t) &= f(g^{-1}t)
\end{align*}
\]
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 \( \phi \) should give the same result as first mapping \( x \) through \( \phi \) and then transforming the representation.

- Planar CNN is equivariant to translations.
  - \( ([L_T f] \ast \psi) = L_T (f \ast \psi) \)
  - \( f \): e.g. earlier CNN layers
Proof

Planar CNN is equivariant to translations.

- \( T(t) = t + u; \ T^{-1}(x) = x - u \)
- \( dT(t) = d(t + u) = dt \)

\[
L_T(f \ast \psi)(x) = (f \ast \psi)(T^{-1}x) = (f \ast \psi)(x - u)
\]

\[
= \int_{\mathbb{R}^2} f(y) \psi((x - u) - y) dy
\]

\[
= \int_{\mathbb{R}^2} f(y) \psi(x - (u + y)) dy
\]

\[
\{ \text{substitute : } v = u + y \} = \int_{\mathbb{R}^2} f(v - u) \psi(x - v) dv
\]

\[
= \int_{\mathbb{R}^2} f(T^{-1}v)) \psi(x - v) dv
\]

\[
= \int_{\mathbb{R}^2} [L_T f](v)) \psi(x - v) dv
\]

\[
= ([L_T f] \ast \psi)(x)
\]
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
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 \(SO(3)\)
    - \(dRy = dy; dRQ = dQ\)
    - \(\int_{S^2} \theta(Ry)dy = \int_{S^2} \theta(Ry)d(Ry) = \int_{S^2} \theta(y)dy\)
    - guaranteed by the parameterization (appendix A)

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

\[
(f * \psi)(R) = \int_{SO(3)} f(Q)\psi(R^{-1}(Q))dQ
\]
Proof (appendix B)

\[(L_Q f) \ast \psi)(R) = \int_{S^2} f(Q^{-1}y)\psi(R^{-1}y)dy\]

\{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 \ast \psi)(Q^{-1}R)

= [L_Q(f \ast \psi)](R)
• GFFT defined on $S^2$ and $SO(3)$
• $SO(3)$: Wigner D-function
• $S^2$: spherical harmonics
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
Equivariance error

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

\[ \Phi : \text{spherical CNN layers with randomly initialized filters} \]

\[ f_i, R_i := \text{randomly chosen features (with channel K=10) and rotations} \]

\[ n = 500 \]
Figure: $\Delta$ 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: $5 \times 5$
  - channels: 32, 64, 10

- **Spherical CNN**: $S^2$conv-ReLU-SO(3)conv-ReLU-FC
  - bandwidth: 30, 10, 6
  - channels: 20, 40, 10

<table>
<thead>
<tr>
<th></th>
<th>NR / NR</th>
<th>R / R</th>
<th>NR / R</th>
</tr>
</thead>
<tbody>
<tr>
<td>planar</td>
<td>0.98</td>
<td>0.23</td>
<td>0.11</td>
</tr>
<tr>
<td>spherical</td>
<td>0.96</td>
<td>0.95</td>
<td>0.94</td>
</tr>
</tbody>
</table>

**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

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
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 \((\alpha, \beta)\) 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

- $S^2$conv-BN-ReLU (50 features)
- $2 \times (SO(3)\text{conv-BN-ReLU})$ (70/350 features)
- max-pooling-BN-FC
- bandwidths: 128, 32, 22, 7

<table>
<thead>
<tr>
<th>Method</th>
<th>P@N</th>
<th>R@N</th>
<th>F1@N</th>
<th>mAP</th>
<th>NDCG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tatsuma_ReVGG</td>
<td>0.705</td>
<td>0.769</td>
<td>0.719</td>
<td>0.696</td>
<td>0.783</td>
</tr>
<tr>
<td>Furuya_DLAN</td>
<td>0.814</td>
<td>0.683</td>
<td>0.706</td>
<td>0.656</td>
<td>0.754</td>
</tr>
<tr>
<td>SHREC16-Bai_GIFT</td>
<td>0.678</td>
<td>0.667</td>
<td>0.661</td>
<td>0.607</td>
<td>0.735</td>
</tr>
<tr>
<td>Deng_CM-VGG5-6DB</td>
<td>0.412</td>
<td>0.706</td>
<td>0.472</td>
<td>0.524</td>
<td>0.624</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td>0.701</td>
<td>0.711</td>
<td>0.699</td>
<td>0.676</td>
<td>0.756</td>
</tr>
</tbody>
</table>

Table: Results and best competing methods for the SHREC17 competition.
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
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)
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
  - $S^2SO(3)\text{conv} - BN - ReLU - SO(3)\text{conv} - BN$
- Shared weights for all atoms: $N \times F$ feature maps
- To achieve permutation invariance:
  - Embedding: MLP $\phi$
  - Sum pooling
  - Regression: MLP $\psi
### Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Author</th>
<th>RMSE</th>
<th>$S^2$CNN</th>
<th>Layer</th>
<th>Bandwidth</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP / random CM</td>
<td>(a)</td>
<td>5.96</td>
<td>Input</td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>LGIKA(RF)</td>
<td>(b)</td>
<td>10.82</td>
<td>ResBlock</td>
<td>10</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>RBF kernels / random CM</td>
<td>(a)</td>
<td>11.40</td>
<td>ResBlock</td>
<td>8</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>RBF kernels / sorted CM</td>
<td>(a)</td>
<td>12.59</td>
<td>ResBlock</td>
<td>6</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>MLP / sorted CM</td>
<td>(a)</td>
<td>16.06</td>
<td>ResBlock</td>
<td>4</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td></td>
<td><strong>8.47</strong></td>
<td>ResBlock</td>
<td>2</td>
<td>160</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DeepSet</th>
<th>Layer</th>
<th>Input/Hidden</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi$ (MLP)</td>
<td></td>
<td>160/150</td>
</tr>
<tr>
<td>$\psi$ (MLP)</td>
<td></td>
<td>100/50</td>
</tr>
</tbody>
</table>

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.

J. R. Driscoll and D. M. Healy.
Computing fourier transforms and convolutions on the 2-sphere.

Large-scale 3d shape retrieval from shapenet core55.