Spherical CNNS

Liangjie Cao

July 8, 2018

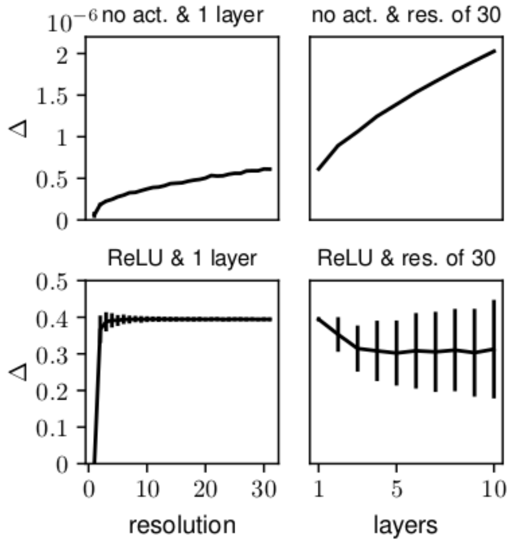

Figure 1. Δ as a function of the resolution and the number of layer

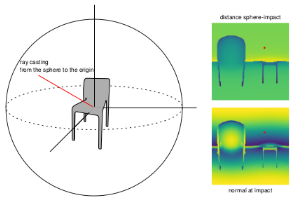

Figure 2. The ray line is cast from the surface of the sphere in direction of its center.The first intersection with the model gives the values of the signal on the sphere. The two images of the right represent two spherical signals in (α, β) representation. They contain respectively the distance from 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

1. Experiments

In this paper we have presented the first instance of a group equivariant CNN for a continuous, non-commutative group. In the discrete case, one can prove that the network is exactly equivariant, but although they can prove $[L_R f] \times \psi = L_R[f \times \psi]$ for continuous functions f and ψ on the sphere or rotation group, this is not exactly true for the discretized version that they actually compute. Hence, it is reasonable to ask if there are any significant discretization artifacts and whether they affect the equivariance properties of the network. If equivariance can not be maintained for many layers, one may expect the weight sharing scheme to become much less effective.

They first tested the equivariance of the $SO(3)$ correlation at various resolutions b. The authors do this by first sampling $n = 500$ random rotations R_i as well as n feature maps f_i with $K = 10$ channels. Then they compute $\Delta = \frac{1}{n}\sum_{i=1}^{n} std(L_R\Phi(f_i) - \Phi(L_R, f_i))/std(\Phi(f_i))$, where Φ is a composition of $SO(3)$ correlation layers with randomly initialized filters. In case of perfect equivariance, they expect this quantity to be zero. The results (Figure 1(top)), show that although the approximation error Δ grows with the resolution and the number of layers, it stays manageable for the range of resolutions of interest.

They repeat the experiment with ReLU activation function after each correlation operation. As shown in figure 1(bottom), the error is higher but stays flat. This indicates that the error is not due to the network layers, but due to the feature map rotation, which is exact only for bandlimited functions.

1.1. Rotated MNIST on the sphere

In this experiment they evaluate the generalization performance with respect to rotations of the input. For testing they propose a version MNIST dataset projected on the sphere (see Fig. 3). They created two instances of this dataset: one in which each digit is projected on the northern hemisphere and one in which each projected digit is additionally randomly rotated.

Architecture and Hyperparameters: As abaseline model, the authors use a simple CNN with layers conv-ReLU-conv-ReLU-FC-softmax, with filters of size 5×5, k= 57, 114, 10 channels, and stride 3 in both layers. They compare to a spherical CNN with layers S^2 conv-ReLU-$SO(3)$conv-ReLU-FC-softmax, bandwidth b = 30, 10, 5

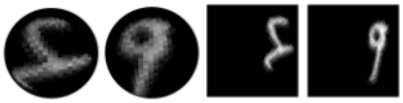

Figure 3. Two MNIST digits projected onto the sphere using stereographic projection. Mapping back to the plane results in nonlinear distortions

and k = 100, 200, 10 channels. Both models have about $165K$ parameters.

Results: We trained each model on the non-rotated (NR) and the rotated (R) training set and evaluated it on the non-rotated and rotated test set. See table 1. While the planar CNN achieves high accuracy in the NR / NR regime, its performance in the R / R regime is much worse, while the spherical CNN is unaffected. When trained on the non-rotated dataset and evaluated on the rotated dataset (NR / R), the planar CNN does no better than random chance. The spherical CNN shows a slight decrease in performance compared to R/R, but still performs quite well.

	NR-NR	R-R	NR-R
planar	0.99	0.45	0.09
spherical	0.91	0.91	0.85

Table 1. 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

2. Recognition of 3D shapes

Next, the authors applied S^2CNN to 3D shape classification. The SHREC17 task [3] contains 51300 3D models taken from the ShapeNet dataset [1] which have to be classified into 55 common categories (tables, airplanes, persons, *etc.*). There is a consistently aligned regular dataset and a version in which all models are randomly perturbed by rotations. They concentrate on the latter to test the quality of our rotation equivariant representations learned by S^2CNN.

The project the 3D meshes onto an enclosing sphere using a straightforward ray casting scheme (see Fig. 2). For each point on the sphere we send a ray towards the origin and collect 3 types of information from the intersection: ray length and cos / sin of the surface angle. They further augment this information with ray casting information for the convex hull of the model, which in total gives us 6 channels for the signal. This signal is discretized using a Driscoll-Healy grid [2] with bandwidth b = 128. Ignoring non-convexity of surfaces we assume this projection captures enough information of the shape to be useful for the recognition task.

References

[1] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, and H. Su. ShapeNet: an Information-Rich 3D model repository. *Computer Science*, 2015. 2

[2] J. R. Driscoll and D. M. Healy. *Computing Fourier Transforms and Convolutions on the 2-Sphere*. Academic Press, 1994. 2

[3] M. Savva, F. Yu, H. Su, M. Aono, B. Chen, D. Cohen-Or, W. Deng, H. Su, S. Bai, and X. Bai. Large-scale 3D shape retrieval from ShapeNet core55. In *Eurographics 2016 Workshop on 3d Object Retrieval*, 2016. 2