Group Equivariant CNN and disentangling β-VAE

August 28, 2022

Shokry Ahmedeo

Abstract

Lot of work has been done these years for the development of group equivariant neural networks, both in the direction of discrete symmetry groups and continuous symmetry groups. Once verified the main properties of group equivariant CNNs and confirmed his performace, in this work I'd like to show a potential use of a G-CNN as a disentagling agent for a hyperspherical β-VAE.

1. Introduction and related work

Group equivariant CNNs have shown to be a powerful tool as a generalized version of regular CNNs, they outperform their classical counterparts both on rotated data and on intrinsic symmetrical data. This is given by the fact that a G-CNN share a larger amount of weights and exploit symmetries, leading to a general increase of the expressive capacity of the network (Cohen & Welling, 2016a). An extensive theoretical work on steerable CNNs has been recently done (Weiler & Cesa, 2021b; Cesa et al., 2022), which in general allows the implementation of any compact continuous group transformation in a convolutional neural network.

For now most of the work has been done in trying to better understand the capability and performance of such architectures on classification tasks, an extensive work on MNIST and CIFAR can be also found here (Weiler & Cesa, 2021b). In this work instead I will present some interesting results obtained by implementing a $SO(2)$ G-CNN inside a Variational Autoencoder as its encoder part, in order to disentangle the image content from its rotation. In fact disentangling property can be very useful in context were data comes naturally rotated such as astronomical data or histopathological images. In general there exist in literature approaches in disentangling pose features such as in this work (Bepler et al., 2019), where the generative part of the autoencoder is explicitly a function of the spatial coordinates of the image, in practice enabling the model to make inference on the rotation and translation.

On the other hand, implementing directly an $SO(2)$ equivariant CNN in a VAE is extremely simple and it natually leads to disentanglement property as I will show later.

2. Methods

Regular group convolutions are an extension of the classical CNNs in which the images and the features are modeled as functions $f : \mathbb{Z}^2 \longrightarrow \mathbb{R}^K$. Where $\mathbb{Z}^2$ is the space of image pixels and $\mathbb{R}^K$ is the feature space. In a G-CNN the feature maps are functions in the G group while in a classical CNN thay are functions in $\mathbb{Z}^2$. Therefore in the case of G group, the notion of convolution can be generalized in such a way that the shifting operation present in the regular convolution can be replaced with a more general tranformation in G (Cohen & Welling, 2016a).

This method works very well for discrete groups, but it can't be applied in the case of continuous groups where instead *steerability* property of feature field overcome this problem (Cohen & Welling, 2016b). Steerable CNNs make use of group representation and feature steerability in order to transform each feature field according to its representation in an equivariant manner, enhancing parameter efficiency and effective implementation of continuous groups (Weiler & Cesa, 2021a).

3. Experimental results

MNIST experiment :

The first test was conducted on MNIST dataset. Test images were randomly rotated in order to verify rotation equivariance of the $SO(2)$ G-CNN model. For both architectures I fixed the following hyperparams: 50 epochs, batch size = 100, lr = 0.001 with exponential scheduler and frequency = 5 for the G-CNN model.

CNN architecture : 3 convolutional layers each with 16, 32, 64 channels respectively, kernel size of 3, stride 1 in the first two layers and stride 2 in the last one. Average pooling of size 2 was used in the first layer and batch norm,

Email: Shokry Ahmedeo <shokry.1750037@studenti.uniroma1.it>.

Deep Learning and Applied AI 2022, Sapienza University of Rome, 2nd semester a.y. 2021/2022.

relu activation and dropout everywhere. Lastly we have a fully connected layer with softmax activation.

G-CNN architecture : In order to have roughly the same number of parameters, I choose 3 G-convolutional layers each with 16, 16, 32 channels respectively with batch-norm, dropout and fourierELU activation. In the last layer an additional (1,1) convolution were applied in order to obtain invariant features and a classical fully connected layer with ELU activation, dropout and softmax were placed.

Table 1. Performance comparison between CNN and GCNN on rotated MNIST after 50 epochs.

Model	Train Loss	Valid loss	Test loss	Test acc.	Params.
CNN	**1.471**	**1.473**	2.025	43.01%	43k
G-CNN	1.522	1.496	**1.561**	**90.22%**	38k

As we can see from Table 1, although the CNN for train and validation set has a lower loss value, test loss and accuracy are drastically worst respect to the G-CNN model.

Hyperspheric G-CNN VAE :

For the second part of the experiment I implemented a variation on a S-VAE (Davidson et al., 2018), where I replaced the regular CNN of the encoder with a G-CNN. The main purpose of this variation is to exploit the group simmetry in order to make the S-VAE invariant to rotation.

The advantage of using a S-VAE lies in the regularization benefit given by a spherical distribution used as the approximate posterior. The main problem of gaussian distribution, is that for low dimension it presents a concentrated probability mass around the center, this is in practice a problem when we deal with multiple gaussian distributions since we end up having a messy superposition of gaussians around the center. Using the von Mises-Fisher distribution instead, we don't have this problem anymore since the distributions are now uniformly distributed on the sphere.

Below the von Mises-Fisher distribution (1):

$$q(\mathbf{z}|\mu,\kappa) = \mathcal{C}_d(\kappa)\exp(\kappa\mu^T\mathbf{z}) \tag{1}$$

where $\mathcal{C}_d(\kappa)$ is a normalizing constant containing the modified Bessel function of the first kind at order d. A spherical distribution hower in high dimensions $(d > 20)$ suffers from the vanishing surface, it can be shown that the peak surface can be obtained in the range $d \in (5, 10)$.

Architecture and hyperparameters For the S-GCNN-VAE encoder, I used 3 G-convolutional layers with 64, 64, 128 channels, batch norm, dropout and fourierELU. A last (1,1) G-convolution in order to extract the invariant feature maps were placed as well as two fully connected layer with dropout, batch norm and ELU activation, one for the mean value μ and the other for the concentration value κ .

The decoder is a standard CNN with 3 transposed convolutional layers with 128, 64, 32 channels, batch norm, dropout and relu activation for each layer.

Batch size, learning rate and frequency were fixed respectively at 100, 0.005 and 6, using also an exponential learning rate scheduler. For training I tried different combinations of β and latent dimension.

Table 2. Validation loss on various combination of β and latent dimension after 50 epochs.

β:	0.01	0.1	1	150	500
dim. = 3	43.38k	42.56k	43.13k	43.38k	44.54k
dim. = 6	30.25k	30.94k	30.04k	32.96k	38.04k
dim. = 10	48.27k	**24.45k**	24.68k	28.5k	35.47k

Although the best validation loss is obtained for β = 0.1 and latent dimension = 10, best performace in disentanglement and recontruction was observed for β = 0.1 and latent dimension = 6. This is resonable since a high latent dimension decreases the information loss at cost of generalization. A high β therefore give more weight to the KL term reducing reconstruction ability, so it's important to find the right trade off. Then I decided to retrain the latter model for 100 epochs obtaining the final validation loss of 30.29k. In Figure 1 an image grid showing the disentaglment ability.

Figure 1. Disentanglement ability of a S-GCNN-VAE for β = 0.1 and latent dimension = 6.

4. Conclusions

The first experiment demostrate the equivariance property of the model on rotated images outperforming drastically the classical CNN. It can be also demonstrated that the use of group convolution can benefits even when data comes naturally symmetric under transformation in G (Veeling et al., 2018). Finally a disentangling rotational property can be naturally obtained replacing the classical CNN encoder with a $SO(2)$ G-CNN. The test performed are not conclusive but are a good starting point for other tests and new implementations.

References

Bepler, T., Zhong, E. D., Kelley, K., Brignole, E., and Berger, B. Explicitly disentangling image content from translation and rotation with spatial-vae. *arXiv:1909.11663v1*, 2019.

Cesa, G., Lang, L., and Weiler, M. A program to build e(n)-equivariant steerable cnns. *Conference paper at ICLR 2022*, 2022.

Cohen, T. S. and Welling, M. Group equivariant convolutional networks. *arXiv:1602.07576v3*, 2016a.

Cohen, T. S. and Welling, M. Steerable cnns. *arXiv:1612.08498v1*, 2016b.

Davidson, T. R., Falorsi, L., Cao, N. D., Kipf, T., and Tomczak, J. M. Hyperspherical variational auto-encoders. *arXiv:1804.00891v2*, 2018.

Veeling, B. S., Linmans, J., Winkens, J., Cohen, T., and Welling, M. Rotation equivariant cnns for digital pathology. *arXiv:1806.03962v1*, 2018.

Weiler, M. and Cesa, G. General e(2) - equivariant steerable cnns. *arXiv:1911.08251v2*, 2021a.

Weiler, M. and Cesa, G. General e(2) - equivariant steerable cnns. *arXiv:1911.08251v2*, 2021b.