A Comparison of Equivariant Vision Models with ImageNet Pre-training

Editors: List of editors' names

Abstract

Neural networks pre-trained on large datasets provide useful embeddings for downstream tasks and allow researchers to iterate with less compute. For computer vision tasks, many models that were pre-trained on ImageNet can be easily downloaded for fine-tuning on various tasks. Recently, equivariant models have become prevalent in vision achieving SOTA in many tasks, yet no such models are available that have been pre-trained on large datasets. In this work, we implement several equivariant versions of the residual network architecture and publicly release the weights after training on ImageNet. We also perform a comparison of enforced vs. learned equivariance in the largest data regime to date.

Keywords: Vision, Equivariance, Pre-training

1. Introduction

Symmetry is inherent to most computer vision problems, for example, object identity is invariant to movements of the camera observing it. Equivariant networks utilize such symmetry during processing, enabling them to generalize to unseen transformations of the data. For this reason, equivariant networks have become popular for tasks with limited data.

Another technique for increasing performance on data-scarce applications is to pre-train the network on a different large dataset, such as ImageNet. A pre-trained network can generalize to inputs similar to the pre-training data after fine-tuning on the target task. Since equivariance and pre-training are both aimed at improving performance and generalization, it makes sense to study how they work together. Even though neural networks can learn to be equivariant given sufficient data, it may still be beneficial to build equivariance in the network, particularly when the target task data is scarce.

In this paper, we explore the combination of pre-training and equivariance by training equivariant vision models on ImageNet1k. The contribution of this work is two-fold:

- We trained equivariant architectures on ImageNet1k to generate a public repository[1] of pre-trained equivariant models for computer vision tasks.

- We compared enforced vs. learned equivariance on ImageNet1k, where we find that enforced equivariance only outperforms if representational capacity is maintained.

[1] Github page will be linked in final version.

2. Related Work

Equivariant Vision Model Constraining networks to be equivariant to image transformations has proved beneficial in many applications, such as pose estimation (Esteves et al., 2019; Klee et al., 2023), reinforcement learning (Wang et al., 2022b), object detection (Han et al., 2021) and semantic segmentation (Linmans et al., 2018). These networks are constructed using group convolutions (Cohen and Welling, 2016), where the learned filters are acted upon by elements of the group. Because a group convolution is expensive for large groups, most work has focused on smaller datasets and model architectures (in contrast to current trends in deep learning (Schuhmann et al., 2022)).

An interesting, parallel line of work to ours introduces a custom layer to extract equivariant features from non-equivariant pre-trained models (Basu et al., 2023b,a). This approach avoids the compute overhead of equivariance constraints during training, while reaping the generalization capabilities during fine-tuning.

Equivariance at Scale An under-explored question in the literature is: what is the role of enforced equivariance in the large data regime? In the low data regime, there is consensus that enforced equivariance is beneficial for performance and generalization (Elesedy and Zaidi, 2021). There is some evidence that reducing equivariance error in network operations improves performance of convolution models on large datasets (Zhang, 2019). However, with enough data an unconstrained network may learn to be equivariant while maintaining more representational capacity (e.g. to handle imperfect symmetries (Wang et al., 2022c)). This idea is supported by the work of Gruver et al. (2022) who observe that sufficiently large models trained with effective training recipes can fit equivariances in the data without needing to impose inductive biases in the network.

3. Method

3.1. Model Implementation

We provide equivariant implementations of ResNet (He et al., 2016). The models are implemented using the `escnn` library (Cesa et al., 2022), where all convolutional layers are replaced with group convolution layers that operate on regular representation features. A group pooling layer is applied to the final feature map, followed by spatial pooling and a linear layer to predict class logits. We keep the same number of layers and blocks as the original ResNet implementations, but change the base width (e.g. number of hidden channels) to maintain the same number of trainable parameters. We provide ResNet18, ResNet50, and ResNet101 architectures that are equivariant to several discrete groups: D_1 (horizontal flips), C_4 (rotations by 90 degrees), D_4 (rotations by 90 degrees and horizontal flips), and C_8 (rotations by 45 degrees).

3.2. Training Details

We followed the `IMAGENET1K_V1`[2] procedure from PyTorch (Paszke et al., 2019) to replicate the performance of existing pre-trained models. The models are trained for 90 epochs on the ImageNet1K dataset (Deng et al., 2009) with an initial learning rate of 0.1 that decays

[2] https://github.com/pytorch/vision/tree/main/references/classification

by a factor of 0.1 every 30 epochs. The models are optimized using SGD a momentum of 0.9 and weight decay of 0.0001 on all trainable parameters. We use an effective batch size of 256. During training, the images were augmented with random crops and horizontal flips. The training script was implemented using PyTorch Lightning (Falcon, 2019) to handle training on multi-GPU devices (models are trained on 4x V100 GPUs).

4. Results

4.1. ImageNet Performance

We report the ImageNet1K classification performances of all models in Table 1. As a baseline, we include the performance of the non-equivariant models (group = $\{e\}$), which we re-trained to verify our training scheme was consistent with torchvision. The weights of all pre-trained models here are available for download on our Github (see Appendix A).

Table 1: Comparison of model performance on ImageNet1K

Architecture	Group	Params	Acc@1	Acc@5	Runtime (min/epoch)
ResNet18	$\{e\}$	11.7M	0.697	0.892	9
	D_1	11.5M	0.709	0.901	15
	C_4	11.5M	0.734	0.915	22
	D_4	11.7M	0.737	0.916	34
	C_8	11.7M	0.738	0.914	34
ResNet50	$\{e\}$	25.6M	0.746	0.921	13
	D_1	25.7M	0.769	0.935	34
	C_4	24.7M	0.785	0.943	42
	D_4	24.8M	0.789	0.946	60
	C_8	24.8M	0.787	0.945	60
ResNet101	$\{e\}$	44.5M	0.776	0.937	21
	D_1	44.7M	0.785	0.943	56
	C_4	43.4M	0.801	0.952	71
	D_4	43.9M	0.804	0.953	106

The results in Table 1 show that, given the same representational capacity, equivariant networks slightly outperform non-equivariant networks on ImageNet. This result aligns with existing works (Weiler and Cesa, 2019; Wang et al., 2022a) that show equivariance is beneficial even if the test set does not include symmetric transformations of the data As Weiler and Cesa (2019) mention, symmetry may still exist at the local level, e.g. edge detection is rotation-equivariant. We encourage future work to explore the performance gap between the non-equivariant and equivariant models when fine-tuned on tasks that are equivariant (e.g. tissue segmentation or pose estimation).

It is worth noting that while the models share similar number of trainable parameters, the equivariant models take significantly longer to train (since the effective size of the filters after basis expansion is larger). For instance, we observe resnet101 takes 2.5x longer to train with D_1 equivariance and 5x longer to train with D_4 equivariance.

4.2. Equivariance vs. Data Augmentation

Given that enforcing equivariance can slow down training, it is important to understand whether the performance boost is worth the cost. To this end, we conduct an experiment where the equivariant models are restricted to a similar compute budget as the non-equivariant models. We evaluate two methods for reducing training time of equivariant models: reducing parameter count to use the same filter size after basis expansion, and reducing number of training epochs for fixed parameter models. All models are trained on a rotated version of ImageNet1k (images are randomly rotated by up to 360 degrees), following the training protocol from Section 3.2. The results for ResNet50 models are shown in Table 3 (we observe similar trends for ResNet18, see Appendix B).

Table 2: Comparison on rotated ImageNet1K with restricted compute budget.

	Group	Params	Epochs	Acc@1	Acc@5
	$\{e\}$	25.6M	90	0.725	0.908
Reduced Epochs	C_4	24.7M	27	0.727	0.910
	D_4	24.8M	18	0.715	0.905
Reduced Params	C_4	5.4M	90	0.707	0.898
	D_4	2.7M	90	0.678	0.881

The results show that using equivariant models with a similar number of parameters perform best, even with significantly less training epochs. However, we find that more equivariance constraints can restrict learning, even if the symmetry is present in the problem: classification of rotated ImageNet is D_4 invariant but the C_4-invariant models outperform the D_4-invariant models for both the capacity- and training-limited settings. Lastly, we find that the non-equivariant model performs similarly to the enforced equivariant models (with similar capacity). One notable limitation of our analysis is that the classification task is invariant; we expect that enforced equivariance would provide more of a benefit on fully equivariant tasks. We hope to extend our analysis to such tasks (segmentation or optical flow) in the future.

5. Conclusion

In this work, we provide implementations of popular residual networks that are equivariant to rotations or reflections in the image plane. We include weights of each model after pre-training on ImageNet-1K that can be easily downloaded. Our hope is that the availability of these models encourages more researchers to try out and fine-tune equivariant models for computer vision applications where symmetries are present (e.g. cell identification, tissue segmentation, object pose estimation). Currently, only ResNet-style architectures are implemented and trained; we plan to add equivariant vision transformers (introduced by Romero and Cordonnier (2020)), and are open to supporting other models.

References

Sourya Basu, Pulkit Katdare, Prasanna Sattigeri, Vijil Chenthamarakshan, Katherine Driggs-Campbell, Payel Das, and Lav R Varshney. Equivariant few-shot learning from pretrained models. *arXiv preprint arXiv:2305.09900*, 2023a.

Sourya Basu, Prasanna Sattigeri, Karthikeyan Natesan Ramamurthy, Vijil Chenthamarakshan, Kush R Varshney, Lav R Varshney, and Payel Das. Equi-tuning: Group equivariant fine-tuning of pretrained models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6788–6796, 2023b.

Gabriele Cesa, Leon Lang, and Maurice Weiler. A program to build E(N)-equivariant steerable CNNs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=WE4qe9xlnQw.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Bryn Elesedy and Sheheryar Zaidi. Provably strict generalisation benefit for equivariant models. In *International Conference on Machine Learning*, pages 2959–2969. PMLR, 2021.

Carlos Esteves, Yinshuang Xu, Christine Allen-Blanchette, and Kostas Daniilidis. Equivariant multi-view networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1568–1577, 2019.

William A Falcon. Pytorch lightning. *GitHub*, 3, 2019.

Nate Gruver, Marc Finzi, Micah Goldblum, and Andrew Gordon Wilson. The lie derivative for measuring learned equivariance. *arXiv preprint arXiv:2210.02984*, 2022.

Jiaming Han, Jian Ding, Nan Xue, and Gui-Song Xia. Redet: A rotation-equivariant detector for aerial object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2786–2795, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

David M Klee, Ondrej Biza, Robert Platt, and Robin Walters. Image to sphere: Learning equivariant features for efficient pose prediction. *arXiv preprint arXiv:2302.13926*, 2023.

Jasper Linmans, Jim Winkens, Bastiaan S Veeling, Taco S Cohen, and Max Welling. Sample efficient semantic segmentation using rotation equivariant convolutional networks. *arXiv preprint arXiv:1807.00583*, 2018.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

David W Romero and Jean-Baptiste Cordonnier. Group equivariant stand-alone self-attention for vision. *arXiv preprint arXiv:2010.00977*, 2020.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

Dian Wang, Jung Yeon Park, Neel Sortur, Lawson LS Wong, Robin Walters, and Robert Platt. The surprising effectiveness of equivariant models in domains with latent symmetry. *arXiv preprint arXiv:2211.09231*, 2022a.

Dian Wang, Robin Walters, and Robert Platt. SO(2)-equivariant reinforcement learning. *arXiv preprint arXiv:2203.04439*, 2022b.

Rui Wang, Robin Walters, and Rose Yu. Approximately equivariant networks for imperfectly symmetric dynamics. In *International Conference on Machine Learning*, pages 23078–23091. PMLR, 2022c.

Maurice Weiler and Gabriele Cesa. General E(2)-Equivariant Steerable CNNs. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Richard Zhang. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pages 7324–7334. PMLR, 2019.

Appendix A. Loading Pre-Trained Models

Our goal with this work is to make it easy for other researchers to test out equivariant models for vision applications. To this end, we provide a simple interface (inspired by torchvision[3]) to instantiate the models and load pre-trained weights. In the example shown in Listing 1, the pre-trained D_1-equivariant ResNet18 model is loaded and a forward pass is performed to generate the final feature map contained regular representations of the group.

```python
from equivision.models import d1resnet18

# load model with weights
model = d1resnet18(pretrained=True)
model.eval()

# generate final feature map
fmap = model.forward_features(img_tensor)
```

Listing 1: Loading Pretrained Model to Predict Equivariant Feature Map

All equivariant models in Table 1 are available on our Github page (link will be available in final version).

Appendix B. Additional Results

Table 3: ResNet18 on rotated ImageNet1K with restricted compute budget.

	Group	Params	Epochs	Acc@1	Acc@5
	$\{e\}$	11.7M	90	0.640	0.853
Reduced Epochs	C_4	11.5M	42	0.686	0.883
	D_4	11.7M	27	0.681	0.883
Reduced Params	C_4	2.0M	90	0.640	0.853
	D_4	1.0M	90	0.520	0.770

[3] https://pytorch.org/vision/stable/models.html