

Learning Layer-wise Equivariances Automatically using Gradients

NeurIPS 2023 (Awarded with spotlight)

Tycho F. A. van der Ouderaa¹, Alexander Immer^{2,3}, Mark van der Wilk^{1,4}

¹ Imperial College London, UK

² ETH Zurich, Switzerland

³ Max Planck Institute for Intelligent Systems, Germany

⁴ University of Oxford, UK

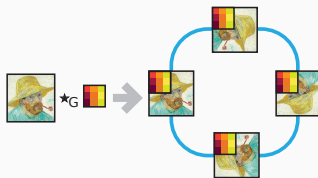
Table of contents

1. Symmetries in Deep Learning
2. Differentiable Equivariance
3. Objective to learn symmetry constraints
4. Results
5. Conclusion

Symmetries in Deep Learning

Symmetries in neural networks

Embedding symmetries into architectures leads to better models!



(a) Convolutions embed translation equivariance. *

(b) Can be extended to other groups, such as rotation.

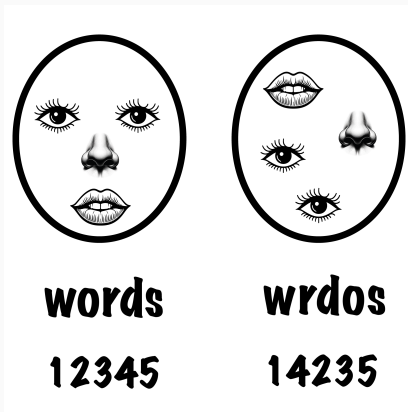
Symmetries need to be chosen or selected with cross-validation.

Can we *learn* the right equivariances with gradients?

* Animation by: Vincent Dumoulin, Francesco Visin - A guide to convolution arithmetic

Part-whole hierarchies

Approximate equivariance exchanges information from place-coded to rate-coded features (relates to capsule networks).



Learning layer-wise equivariances is about *determining the right amount of disentanglement* in part-whole hierarchies.

Why symmetry learning is hard

Learning symmetries is hard...

- *Objective* Equivariances constrain functions a neural net can represent. Consequently, optimising data fit does not encourage the use of symmetry.
- *Parameterisation* It is not clear how to parameterise differentiable symmetry constraints effectively.

We tackle both issues.

Differentiable Equivariance

Equivariant subspace

To obtain **differentiable equivariance constraints**, we relax layer-wise equivariances. Starting from a linear layer:

$$\underbrace{\sum_c \sum_{x,y} \mathbf{x}(c, x, y) \boldsymbol{\theta}(c', c, x', y', x, y)}_{\text{fully-connected FC}} \quad (\text{generalises to groups in paper})$$

for discrete spaces this is a classic FC layer $\mathbf{y} = \mathbf{W}\mathbf{x}$, with $\mathbf{W} = \text{vec}(\boldsymbol{\theta})$. We separately parameterise the equivariant subspace:

$$\mathbf{y}(c', x', y') = \underbrace{\sum_c \sum_{x,y} \mathbf{x}(c, x, y) \boldsymbol{\theta}(c', c, x', y', x, y)}_{\text{fully-connected FC}} + \underbrace{\sum_c \sum_{x,y} \mathbf{x}(c, x, y) \bar{\boldsymbol{\theta}}(c', c, \bar{x}, \bar{y})}_{\text{equivariant CONV}}$$

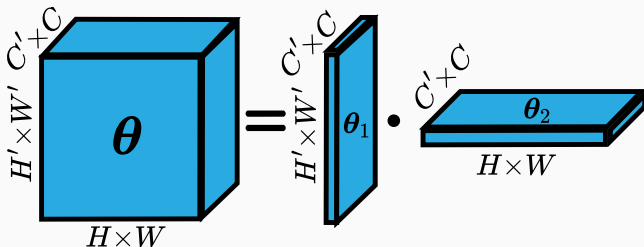
Non-stationary $\boldsymbol{\theta}$ and convolutional filter $\bar{\boldsymbol{\theta}}(c', c, \bar{x}, \bar{y})$ with stationary $(\bar{x}, \bar{y}) = (x' - x, y' - y)$. Similar to ‘residual pathways’ [Finzi et al., 2021].

Factorising layers

Relaxed equivariance can be hard to parameterise in a reasonable parameter count, especially for larger group sizes $|G|$.

Some that scale to (image size) translation groups:

- Non-stationary filters [van der Ouderaa et al., 2022]
- Residual pathways [Finzi et al., 2021a] (C^2S^4)

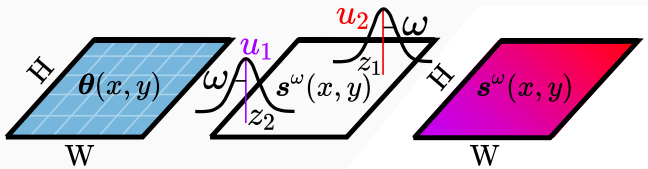


Factorisation simple trick to reduce $C^2S^4 \rightarrow 2C^2S^2$:

Spatial sparsification

Further, reduce spatial dimension in a small set of P basis functions

$$\mathbb{C}^2 \textcolor{brown}{S}^2 \rightarrow \mathbb{C}^2 \textcolor{green}{P}$$



Casts symmetry discovery as automatic relevance determination.

Shown effective on groups with b-splines [Bekkers, 2019] and exponential basis features [van der Ouderaa and van der Wilk, 2023].

Equivariance as controllable soft constraint

To automatically learn equivariance in each layer l , we interpolate non-equivariant/equivariant solutions with $\boldsymbol{\eta} = \{\sigma_l^2\}_{l=1}^L$:

$$\sigma_l^2 = 0 \implies \text{strict equivariance} \quad \sigma_l^2 > 0 \implies \text{relaxed equivariance}$$

To satisfy this, we may consider:

$$\underbrace{\text{residual pathway prior: } \mathcal{N}(\boldsymbol{\theta}|0, \sigma_l^2)}_{\text{Finzi et al., 2021}} \quad \underbrace{\text{non-stationary filter: } \mathcal{N}(\omega_l^2|0, \sigma_l^2)}_{\text{van der Ouderaa et al., 2023}}$$

We consider both. Instead of setting or tuning prior variances that control equivariance constraints, we propose to infer the amount of equivariance from train data with approximate empirical Bayes.

Objective to learn symmetry
constraints

Inspired by the marginal likelihood

Inspired by optimising *marginal likelihood*: (Type-II ML)

$$p(\mathcal{D}|\boldsymbol{\eta}) = \int_{\boldsymbol{\theta}} p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\eta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}$$

Estimate with modern linearised Laplace approximations (KFAC $\mathbf{H}_{\boldsymbol{\theta}_*}$)

$$p(\mathcal{D}|\boldsymbol{\eta}) \approx \underbrace{-\log p(\mathcal{D}|\boldsymbol{\theta}_*, \boldsymbol{\eta})}_{\text{NLL / Data fit}} - \underbrace{\log p(\boldsymbol{\theta}_*) - \frac{P}{2} \log(2\pi) + \frac{1}{2} \log |\mathbf{H}_{\boldsymbol{\theta}_*}|}_{\text{Occam's factor}}$$

Effectively, we optimize equivariance constraints of the architecture that yield wide minima in the loss landscape.


Proven effective for invariance learning in [Immer, et. al, 2022]

Results

Toy problem

Problems engineered such that strict symmetry constraints are either good or bad.

Example:



(3, upper left)

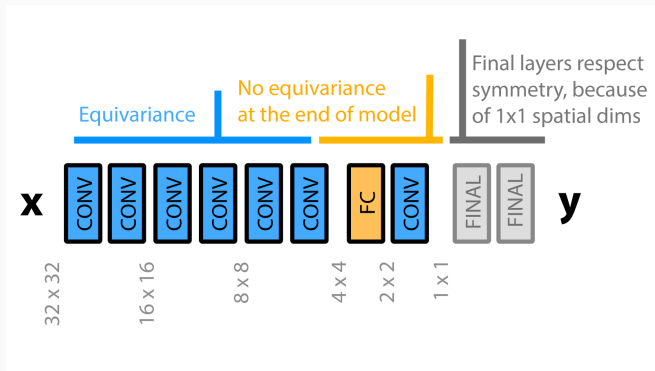
Symmetry	Prediction task	MAP		Diff. Laplace		Learned (ours)
		FC	CONV	FC	CONV	
		112.7 M	0.4 M	112.7 M	0.4 M	1.8 M
Strict symmetry	(digit)	95.73	99.38	97.04	99.23	98.86
Partial symmetry	(digit, quadrant)	95.10	24.68	95.46	24.50	99.00

Table 1: Preventing symmetry misspecification. Test accuracy for non-symmetric FC, strictly symmetric CONV, and learnable symmetry F-FC+CONV models on a translation invariant task and a task that can not be solved under strict translation symmetry. Unlike the FC and CONV baselines, the proposed model with learnable symmetry constraints achieves high test performance on both tasks.

Our method capable of learning symmetry performs well on both.

Learning to become convolutional

On CIFAR-10 classification, we train a network with flexible layers that can automatically adapt symmetry constraints:



Most layers **learn to become convolutional**, but in last layers it learns to break symmetry. This is analogous to many successful hand-engineered architectures.

Multiple symmetry groups

Automatic relevance determination between *multiple groups*.

Dataset	# Params	MAP		Learned with Differentiable Laplace (ours)			Rel. Effective Num. of Param.					
		Test NLL (\downarrow)	Test accuracy (\uparrow)	Test NLL (\downarrow)	Test accuracy (\uparrow)	Approx. MargLik (\downarrow)	FC (%)	CONV (%)	GCONV (%)			
MNIST	1.2 M	0.172	97.59	0.023	99.21	0.328	10 (0-46)	15 (0-98)	75 (1-100)			
Translated MNIST	1.2 M	0.812	90.78	0.053	98.27	0.216	0 (0-0)	23 (0-99)	77 (1-100)			
Rotated MNIST	1.2 M	0.819	91.02	0.136	95.55	0.896	8 (0-20)	8 (0-47)	83 (47-100)			
CIFAR-10	1.2 M	3.540	68.33	0.552	80.94	0.926	0 (0-1)	44 (0-99)	56 (0-100)			
Rotated CIFAR-10	1.2 M	5.953	48.30	1.236	55.68	1.630	4 (0-22)	14 (0-41)	82 (58-99)			

Table 3: Selecting from multiple symmetry groups. Negative log likelihood (NNL) and Laplace learned symmetries measured by *mean (min-max)* relative effective number of parameters over layers.

Learning symmetry improves final test performance.

Conclusion

ELLA: Equivariance Learning with Laplace Approximations

ELLA for automatic discovery of layer-wise equivariances from data.

- Differentiable objective that can learn symmetry constraints
- Scalable parameterisation of flexible layer-wise equivariance

Demonstrated the principle of learning layer-wise equivariances.

Come visit our poster **ID: 71150**

Or check out:

- Paper: NeurIPS 2023 (Awarded with spotlight)
- Code: <https://github.com/tychovdo/ella>
- Twitter/X: @tychovdo, @a1mmer, @markvanderwilk

Be sure to come by our poster!

- ID 71150 -