Imperial College London

Learning Invariant Weights in Neural Networks

Oral presentation @ UAI 2022

Tycho F.A. van der Ouderaa, Mark van der Wilk

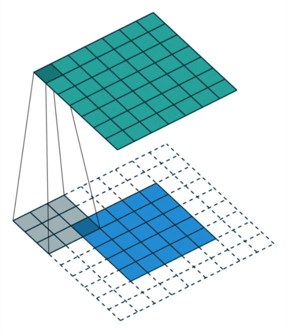
Imperial College London

Open pdf in Adobe Acrobat to enable animations.

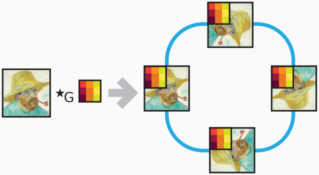
Invariances in Deep Learning

Symmetries in Deep Learning

Embedding symmetries into architectures leads to better models!



(a) Convolutions embed translation equivariance. *



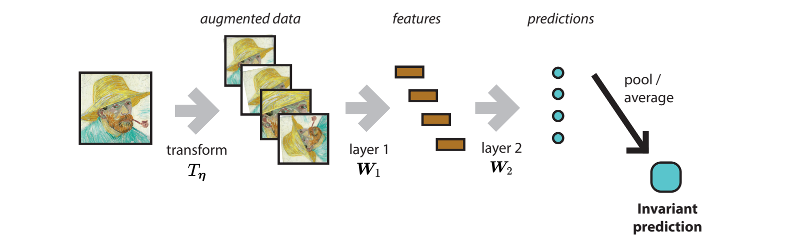
(b) Can be extended to other groups, such as rotation.

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

Can we *learn* the right invariances with gradients?

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

Constructing an invariant neural network



Consider a shallow network

$$g_{\boldsymbol{\theta}}(T(\mathsf{x})) = \sigma(\mathsf{W}_2 \circ \phi(\mathsf{W}_1 \circ T \circ \mathsf{x}))$$

integrated over a set of transformations

$$f_{\boldsymbol{\theta}}(\mathsf{x}; \boldsymbol{\eta}) = \int g_{\boldsymbol{\theta}}(T(\mathsf{x})) p_{\boldsymbol{\eta}}(T) \mathrm{d}T$$

We can apply transformations to the input or the weights first:

$$\mathsf{W}_1 \circ (T \circ \mathsf{x}) = (\mathsf{W}_1 \circ T) \circ \mathsf{x}$$

Parameterizing invariance

General parameterization of invariance

$$T = \exp\left(\sum_i \epsilon_i \eta_i \mathsf{G}_i\right), \qquad \boldsymbol{\epsilon} \sim U[-1, 1]^k$$

use set of affine generators

G_1 : translation x	G_4 : scale x
G_2 : translation y	G_5 : scale y
G_3 : rotation	G_6 : shear

Stochastic or deterministic sampling

The invariant predictor is described by an integral, which is intractable.

However, we can predict with an approximation from MC samples:

$$\hat{f}_{\boldsymbol{\theta}}(\mathsf{x}; \boldsymbol{\eta}) = \frac{1}{S} \sum_{i=1}^{S} g_{\boldsymbol{\theta}}(T_i(\mathsf{x}))$$

to get an unbiased estimate

$$f_{\boldsymbol{\theta}}(\mathsf{x}; \boldsymbol{\eta}) = \mathbb{E}_T \left[\hat{f}_{\boldsymbol{\theta}}(\mathsf{x}; \boldsymbol{\eta})\right]$$

Bayesian Model Selection (Murphy §5.3, MacKay §28, Bishop §3.4)

Normally, we would use cross-validation to learn hyperparameters.

Finding optimal hyper-parameters $\boldsymbol{\eta}$ with Bayesian model selection

$$p(\boldsymbol{\theta}, \boldsymbol{\eta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\eta})p(\boldsymbol{\theta}|\boldsymbol{\eta})p(\boldsymbol{\eta})}{p(\mathcal{D}|\boldsymbol{\eta})} \qquad \text{(Full Bayes)}$$

$$\hat{\boldsymbol{\eta}} = \arg\max_{\boldsymbol{\eta}} \int p(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\eta})p(\boldsymbol{\theta}|\boldsymbol{\eta})\mathrm{d}\boldsymbol{\theta} \qquad \text{(Empirical Bayes)}$$

Well known procedure (Empirical Bayes, Type-II ML).
Used to learning invariances in GPs [van der Wilk; 2018].

Marginal likelihood is intractable for neural networks.

We derive a lower bound using multi-sample Jensen's inequality:

$$\log p(\mathcal{D}) \geq \mathbb{E}_{q(\boldsymbol{\theta})}\left[\mathbb{E}_{\prod_{i=1}^{N} p_{\boldsymbol{\eta}}(T_i)}\left[\log p(\mathsf{y}|\hat{f}_{\boldsymbol{\eta}}(\mathsf{x}, \boldsymbol{\eta}))\right]\right] - \mathrm{KL}(q(\boldsymbol{\theta}|\boldsymbol{\mu}, \boldsymbol{\Sigma})||p(\boldsymbol{\theta}))$$

Similar to Nabarro et al., 2022 and Schwöbel et al., 2022.

Results

Learning affine invariances

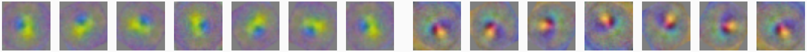


Figure 1: Visualisation of filter banks trained on rotated CIFAR-10 data.

(a) Trained on regular MNIST.

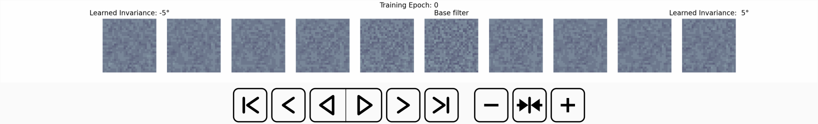
(b) Trained on rotated MNIST.

(c) Trained on scaled MNIST.

(d) Trained on translated MNIST.

Figure 2: The same model capable of learning affine invariances learns filter banks with different invariances corresponding the data it was trained on.

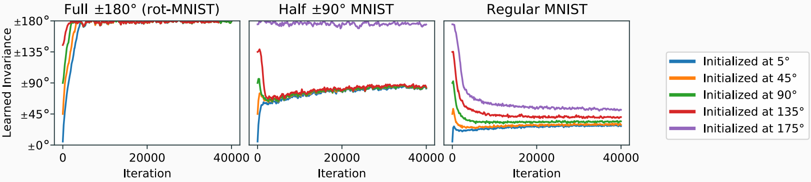
Learned invariant filter banks



Recovering invariances

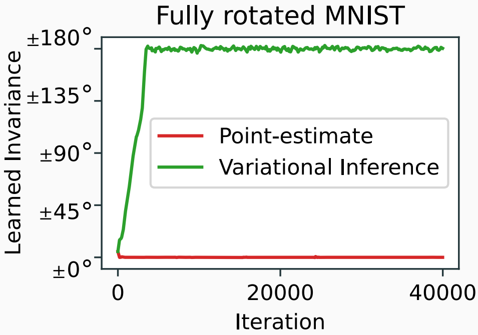
Relaxed invariance is not limited by the 'closure' axiom of groups.

We can learn the right amount of invariance from training data.



The necessity of the Bayesian approach

The marginal likelihood balances data fit and model complexity.



This proves to be very useful beyond predictive uncertainty.

Conclusion

Conclusion

We can learn invariant weights from data!
But, only in shallow networks

Exciting work in progress...

- Scaling the objective to deep networks
  "Invariance Learning in Deep Neural Networks with Differentiable Laplace Approximations",
  A Immer, TFA van der Ouderaa, V Fortuin, G Rätsch, M van der Wilk (2022)
- Parameterization of learnable layer-wise equivariance
  "Relaxing Equivariance Constraints with Non-stationary Continuous Filters",
  TFA van der Ouderaa, M van der Wilk (2022)

Very happy to engage in discussions on the topic!

Follow on @tychovdo

Come chat at our poster ID# 419