Learning approximate invariance requires far fewer data

Jean-Michel A. Sarr [1 2] **Alassane Bah** [1] **Christophe Cambier** [2]

Abstract

Invariance has been conjectured to be the key to learning with small datasets because it improves generalization. This paper introduces a new algorithm to learn approximate invariant priors in the framework of Nalisnick & Smyth (2018) requiring only a small fraction of the dataset. Furthermore, the algorithm is shown to yield stable generalization improvements when compared with two flavors of data augmentation (DA) in the context of Bayesian posterior learning. Besides, it is shown that invariance inducing techniques enhance uncertainty in Bayesian Neural Networks.

1. Introduction

The current success of deep supervised learning has been largely attributed to the advent of powerful computer infrastructure and big datasets (Goodfellow et al., 2016). Yet when it comes to small datasets the current paradigm falls short when compared to human learning. Hence, one of the remaining challenges of machine learning is to be able to learn with fewer data. It has been conjectured that the key to learning with few examples is invariance (Anselmi et al., 2014), the authors argued that most of the complexity in recognition tasks is due to viewpoint or illumination nuisances that swaps the intrinsic characteristics of objects. This suggests that if an oracle could factor out all the transformations concerning viewpoint, scale, position, etc, the problem of categorization would become easy and could be done with very few labeled examples. The nuisances as described by Anselmi et al. (2014) are also called uninformative factors of variation by Bengio et al. (2013) who view the factor of variations in two categories: the factor of variations that are uninformative and informative to the task at hand. An ideal model would be sensitive to informative factors of variations and invariant to uninformative factors of variations. Informative factors of variation can affect the label predicted. For instance, in a classification setting, cats and dogs have two eyes, varying the number of eyes tells a lot in the object to identify, hence the number of eyes is informative. Uninformative factors of variation are label preserving, for instance, we could apply rotations, blurs, or translations on a cat image, the resulting images would still represent a cat for human perception but could fool a model that is not invariant to such transformations, therefore invariance is a highly desired property.

Two ways to learn invariant models are considered: using data augmentation (DA) (Fawzi & Frossard, 2015; Dao et al., 2019), or learning approximate invariant features (Nalisnick & Smyth, 2018). There are various ways to learn invariant features directly, for instance, Simard et al. (1992) added a regularization term ensuring that the model has a zero derivative in the direction of the transformation. Another alternative is to take advantage of the topology of the data (Ranzato et al., 2007; LeCun, 2012). Recently, Dao et al. (2019) have shown a correspondence between applying augmentation on the data space and the feature space of a model. The authors have shown that the effect of DA during training can be transferred by perturbing the weight of the model in a specific manner and keeping the data without augmentation. This links to the work of Nalisnick & Smyth (2018) who have developed a general method to learn approximate invariant priors in Bayesian neural networks (BNNs), Indeed, DA applied directly onto the weights of a neural network create a weight distribution, this is typical of BNNs which are designed to involve a distribution over its weights. The Bayesian framework allows naturally formulating the technique of DA because it allows to treat uninformative factors of variation as prior knowledge and let the posterior learn only informative factor of variation. This factorization of factors of variation allows us to naturally regularize the model by learning representations that are only compatible with the invariances. Indeed, the prior can be seen as a regularization term in BNNs, for instance, there is a direct mapping between weight decay and Gaussian priors (MacKay, 1992). Furthermore, DA has been shown to regularize models (Leen, 1995; Dao et al., 2019). It means that we can compare the generalization effects of approximate invariant prior and DA as both techniques induce invariance and regularization. However, in the case the modeler train a

[1]UMI UMMISCO Université Cheikh Anta Diop Dakar, Sénégal [2]UMI UMMISCO Sorbonne Université, IRD, F-75006 Paris, France. Correspondence to: Jean-Michel A. Sarr <jm.amath@icloud.com>.

Presented at the ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning. Copyright 2020 by the author(s).

BNN with DA, posterior learning would require to decipher informative and uninformative factors of variation at the same time.

Our main contribution in section 4.2 is to propose an algorithm in the framework of Nalisnick and Smyth (2018) that requires only a small fraction of the whole dataset to learn approximate invariant priors. Furthermore, we compared its stability with two flavors of DA in section 5.1. Stability as in perturbation analysis refers to the variability of the learning outcome obtained by a change in the learning procedure (Bousquet & Elisseeff, 2002). This research varies the invariance inducing technique and studies how it impacts generalization gains. Moreover, we also show that invariance inducing techniques enhance uncertainty in posterior learning in Appendix B.2.

2. Data Augmentation

DA has been widely used to improve the generalization performance of deep learning models (Dao et al., 2019), it has allowed getting state of the art results in numerous domain such as computer vision (Shorten & Khoshgoftaar, 2019) and speech processing (Ko et al., 2015; Park et al., 2019). When the dataset at hand is small, DA is often applied offline, it means that one or multiple transformations are applied to the original dataset multiple times to augment the dataset size. The amount of augmentation is referred to as the augmentation factor, for instance when the total size of the dataset is made 16 times larger, the augmentation factor is 16. The caveat with this approach is its memory consumption, for instance, a big dataset like Imagenet (Deng et al., 2009) could hardly be augmented by a factor of 64, because it would take too much place into the memory of personal laptops. When the dataset is large an alternative is to apply DA online, meaning that each batch of data is augmented 'on the fly' before being fed into the model. As a result, online DA is memory efficient, but it takes longer to train as it adds a sampling operation in the training loop and often reads data from the disk. Also, online augmentation results in a noisier data distribution than offline augmentation, because of the constant change during training. To the best of our knowledge, we have not yet seen any comparison of online and offline DA. Moreover, because invariance is often induced in standard neural networks, its effect on uncertainty has not been studied yet. Indeed, except with ensemble methods (Lakshminarayanan et al., 2017), or Monte Carlo dropout (Gal & Ghahramani, 2016), neural networks don't exhibit uncertainty.

3. Bayesian Neural Networks

One way to picture how BNNs work is by comparing them with standard neural networks. The first difference is that BNNs involve two neural networks with the same architecture, one for the prior and one for the posterior. They are related by the Bayes formula:

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{p(X)}$$

Where X is a dataset, $p(X)$ is the evidence, θ are the parameters of the network, $p(X|\theta)$ is the likelihood, finally $p(\theta)$ and $p(\theta|X)$ are respectively the prior and the posterior. The second difference is that each weight is represented by a probability distribution instead of a single unit. They are many ways to select a distribution for a learning problem, the reader interested might refer to appendix A. Often the prior is chosen with heuristics and does not involve learning, and only posterior learning is considered. The algorithm presented in section 4.2 is focused on learning the prior and is typically done before learning the posterior.

4. Learning Approximate Invariant Priors

4.1. Preliminaries

Most augmentation schemes can be cast as perturbation processes (PP) that are parametrized by a given magnitude, for instance, an image could be perturbed by a rotation between -20 degrees to 20 degrees to get a new image. More generally, if x is an example, a transformation $\hat{x}$ can be sampled from a parametrized perturbation process q_η and can be written as follow: $\hat{x} \sim q(\hat{x}; x, \eta)$, where η parametrize the PP. Supervised learning focuses on learning functions, mapping high dimensional input data $x \in \mathbb{R}^d$ coming from an unknwown distribution $p_\mathcal{D}$ to output labels y. In that setting, parametric models can often be framed as probabilistic ones to easily compute their likelihood $p(y|x, \theta)$, where θ the parameters of the model typically depends on the architecture of the model. Such models can be described as invariant to a perturbation process q_η if and only if $p(y|x) = p(y|\hat{x}) \, \forall \hat{x} \in \mathcal{X}_{q_\eta}(x)$. With $\mathcal{X}_{q_\eta}(x)$ the set of all transformed examples from q_η. However this constraint might be too strong, and can be weakened with an approximate invariance constraint by ensuring $p(y|x) = \mathbb{E}_{\hat{x}\sim q_\eta}[p(y|\hat{x})]$. Thus an optimization objective might be to minimize the Kullback-Leibler (KL) divergence between those two terms. The KL divergence is well known to measure the discrepancy between probability distributions in generative models (Kingma & Welling, 2013) or in BNNs (Graves, 2011; Blundell et al., 2015). It is defined as follows for two probability distribution p, q defined on a domain $\mathcal{X}$: $KL(p||q) = \int_\mathcal{X} p(x) \log(\frac{p(x)}{q(x)}) dx$.

Furthermore the prior can be exposed by expressing a model prediction $p(y|x)$ as a marginalized likelihood $p(y|x) = \mathbb{E}_{\theta\sim p_\lambda}[p(y|x, \theta)]$, where p_λ is the parametrized prior distribution. For instance in case the prior distribution is a Gaussian $\mathcal{N}(\mu, \sigma^2)$, then $\lambda = \{\mu, \sigma^2\}$. Finally, the approxi-

mate invariance constraint can be formulated with an upper bound thanks to the Jensen inequality and the convexity of the KL divergence:

$$KL(p(y|x)||\mathbb{E}_{\hat{x}\sim q_\eta}[p(y|\hat{x})]) \leq \mathbb{E}_{\hat{x}\sim q_\eta}\mathbb{E}_{\theta\sim p_\lambda} KL(p(y|x,\theta)||p(y|\hat{x},\theta))$$

Minimizing this upper bound will result in learning approximate invariant priors, however directly optimizing this criterion would result in priors learning spurious solutions, in other words, solutions that would not be helpful in the second stage of Bayesian inference during posterior learning. This has been shown analytically by Nalisnick & Smyth (2018). Thus, the proper optimization objective should be enhanced by referring to the maximum entropy principle: the appropriate distribution for representing prior beliefs is one that obeys known constraints and has maximum entropy otherwise (Nalisnick & Smyth, 2018). Hence, the following formulation of the objective function:

$$\mathcal{J}(\lambda, x) = \mathbb{H}_\lambda[\theta] - \mathbb{E}_{\hat{x}\sim q_\eta}\mathbb{E}_{\theta\sim p_\lambda} KL(p(y|x,\theta)||p(y|\hat{x},\theta))$$

With $\mathbb{H}_\lambda[\theta] = -\int_\theta p_\lambda(\theta)\log(p_\lambda(\theta))d\theta$ the entropy of the prior. Hence, the optimisation objective is to maximize the objective function for each example from the data distribution $p_\mathcal{D}$, which we write as an expectation in the following optimisation problem:

$$p_{\lambda^*}(\theta) = \arg\max_\lambda \mathbb{E}_{x\sim p_\mathcal{D}}\mathcal{J}(\lambda, x)$$

This optimization problem makes sense with respect to the goal of finding high entropy priors that obeys to the approximate invariance constraint because minimizing the upper bound to the approximate invariance objective is equivalent to maximizing its opposite. Finally, unbiased estimates of gradients with respect to the expectation are found using the reparametrization trick (Kingma & Welling, 2013).

4.2. A lightweight formulation

A new approximate invariant learning technique is presented in algorithm 1 extending the work of Nalisnick and Smyth (2018). The algorithm uses only a small fraction of the full dataset and only a finite set of augmentation (or augmentation factor) for the whole learning process. Furthermore, we only sample as many weights from the prior as there are augmentations in the finite set. In the end, the process requires only $O(N)$ samples and $O(MN)$ operations at each epoch, with M the dataset size and N the augmentation factor. Also, we added an annealing term to the entropy, as we found that the optimization could easily favor maximizing the entropy to a point where the expected KL would increase. While, the paper of Nalisnick and Smyth (2018) presents a novel and elegant mathematical framework to learn approximate invariant priors,

Algorithm 1 Lightweight approximate invariant prior

Input: data distribution $p_\mathcal{D}$, prior p_λ, perturbation process q_η, sample size M, augmentation factor N
Sample $\{x^{(1)}, \cdots, x^{(M)}\}$ from $p_\mathcal{D}$
for $i = 1$ **to** M **do**
  Sample $\{\hat{x}_1^{(i)}, \cdots, \hat{x}_N^{(i)}\}$ from q_η
end for
repeat
  Sample $\{\theta_1, \cdots, \theta_N\}$ from p_λ
  div $\leftarrow \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{K} KL(p(y^{(m)}|x^{(m)},\theta_n)||p(y^{(m)}|\hat{x}_n^{(m)},\theta_n)$
  $\mathcal{L}(\lambda) \leftarrow \beta\cdot\mathbb{H}_\lambda[\theta] -$ div
  Compute $\nabla_\lambda\mathcal{L}(\lambda)$
  $\lambda \leftarrow \lambda + \alpha\nabla_\lambda\mathcal{L}(\lambda)$
until convergence

it still involves the calculation of a complex expectation i.e: $\mathbb{E}_{x\sim p_\mathcal{D}}\mathbb{E}_{\hat{x}\sim q_\eta}\mathbb{E}_{\theta\sim p_\lambda} KL(p(y|x,\theta)||p(y|\hat{x},\theta))$. The authors used nested Monte Carlo to estimate it. The dataset is always fixed, so we cannot sample indefinitely from the data distribution $p_\mathcal{D}$. However, it is possible to sample indefinitely from the perturbation process and the prior. The original formulation used the whole dataset and sampled new perturbations and new prior weights for each example at each epoch. So if the dataset has size M, and the nested Monte Carlo estimation requires N sample from the prior p_λ and N sample from the perturbation process q_η, then, the KL estimate would require $O(MN)$ samples and $O(M\cdot N^2)$ operations at each epoch.

5. Experiments

In all the experiments described below, we used the same architecture, a Bayesian neural network with one hidden layer of 512 neurons. The optimization algorithm Adam with a learning rate of 0.001 was used for all training purposes. The experiment described in section 5.1 evaluated the effect of online, offline augmentation on posterior learning of Bayesian Neural Networks. The experiment described in appendix B.1 investigated different scenarios to learn invariant priors, all of them were fully factorized Gaussian. Posterior learning was done using variational inference with fully factorized Gaussian as approximation distribution. Finally, an experiment has been conducted to evaluate the uncertainty gains by each invariance inducing technique after posterior learning. It is described in B.2. All the experiments were conducted with Pytorch (Paszke et al., 2019). [1]

[1]The code is available at https://github.com/jmamath/learning-approximate-invariance-requires-far-fewer-data

5.1. Stability of invariance inducing techniques on generalization gains

In this experiment, we investigated the generalization gains and stability of posterior learning under three invariance inducing techniques: invariant prior, online, and offline DA. To evaluate the effect of the different forms of augmentation, four BNNs were used. The first three BNNs involved a fully factorized standard Gaussian as a prior $\mathcal{N}(0, 1)$ and the posteriors were trained respectively with: no augmentation (vanilla), offline augmentation with an augmentation factor of 8, and online augmentation respectively. The last BNN used an approximate invariant prior and its posterior was trained with no augmentation at all. The approximate invariant prior was learned for 2000 epochs with an early stopping after 500 epochs without progress. The sample size was set to 1024, and the augmentation factor to 64. Finally, the annealing term for the entropy was $\frac{i}{2000}$, where i was the current epoch. Three augmentation procedures were selected: rotation, uniform background noise, and a combination of both. The rotation of the image was sample between $[-20, 20]$; uniform noise between $[0, 255]$ was added to the image. The last augmentation was simply a composition of rotation and noise. The four Bayesian neural networks were compared on the MNIST and Fashion MNIST datasets (Deng, 2012; Xiao et al., 2017).

Moreover, to have a basis of comparison we fixed the amount of computation available for each model in terms of mini-batch processed. Therefore, 800 epochs were used for each model except for offline DA that was trained with only 100 epochs, because its dataset was 8 times larger. The batch size for every model was set to 1024, and we computed 30 Monte Carlo pass per batch of data to compute the expected lower bound (ELBO). Finally, each model was tested with the respective test set of the MNIST and Fashion MNIST datasets. To get the prediction on the test set, 100 passes per batch of data were made, and we ran the evaluation 10 times to get the mean and standard deviation for each model.

The results are summarized in table 1. We observe that using approximate invariant priors never performed worse than the control Vanilla BNN. Overall it yielded the best generalization improvements. Also, it takes roughly as long to train as it takes for the vanilla BNN. Online augmentation takes the longest to train and is the least stable method, for instance, it failed multiple times: noise on Fashion MNIST, rotation+noise on MNIST, and Fashion MNIST. We explain that online augmentation dramatically shifts the data distribution because much more noise is added to the learning process, which leads the optimizer to more likely be trapped in local minima. Hence online DA might require a more careful optimization strategy to yield good results. Yet it got the best generalization accuracy for noise on Fashion MNIST. Offline learning is faster to train and more stable when compared to online learning, it achieves the best result for rotation MNIST.

Table 1: Comparison of the different settings for posterior learning.

Rotation - MNIST			
Prior	DA	Accuracy (%)	Time (s)
$\mathcal{N}(0, 1)$	Online	96.7 ± 0.2	11726
$\mathcal{N}(0, 1)$	Offline	**96.9 ± 0.1**	7649
$\mathcal{N}(0, 1)$	N.A	96.8 ± 0.1	10483
Invariant	N.A	96.8 ± 0.1	10715
Noise - MNIST			
Prior	DA	Accuracy (%)	Time (s)
$\mathcal{N}(0, 1)$	Online	95.3 ± 0.1	17276
$\mathcal{N}(0, 1)$	Offline	79.5 ± 0.1	7660
$\mathcal{N}(0, 1)$	N.A	96.8 ± 0.1	10483
Invariant	N.A	**97.0 ± 0.1**	10411
Rotation+Noise - MNIST			
Prior	DA	Accuracy (%)	Time (s)
$\mathcal{N}(0, 1)$	Online	19.8 ± 0.6	18934
$\mathcal{N}(0, 1)$	Offline	82.2 ± 0.2	7643
$\mathcal{N}(0, 1)$	N.A	96.8 ± 0.1	10483
Invariant	N.A	**97.0 ± 0.1**	10431
Rotation - Fashion MNIST			
Prior	DA	Accuracy (%)	Time (s)
$\mathcal{N}(0, 1)$	Online	**77.5 ± 0.3**	11529
$\mathcal{N}(0, 1)$	Offline	76.4 ± 0.5	7554
$\mathcal{N}(0, 1)$	N.A	44.3 ± 0.2	10476
Invariant	N.A	74.7 ± 0.2	10659
Noise - Fashion MNIST			
Prior	DA	Accuracy (%)	Time (s)
$\mathcal{N}(0, 1)$	Online	26.5 ± 0.3	16951
$\mathcal{N}(0, 1)$	Offline	78.4 ± 0.3	7593
$\mathcal{N}(0, 1)$	N.A	44.3 ± 0.2	10476
Invariant	N.A	**83.3 ± 0.3**	10221
Rotation+Noise - Fashion MNIST			
Prior	DA	Accuracy (%)	Time (s)
$\mathcal{N}(0, 1)$	Online	21.1 ± 0.3	18600
$\mathcal{N}(0, 1)$	Offline	70.6 ± 0.6	7570
$\mathcal{N}(0, 1)$	N.A	44.3 ± 0.2	10476
Invariant	N.A	**83.8 ± 0.2**	10290

6. Discussion

It should be pointed out that the approximation distribution used for prior and posterior learning is very rudimentary, indeed fully factorized Gaussians are among the simpler approximation distributions one could choose. Related theoretical development could be to extend the work of Sokolic et al. (2016) to the case of approximate invariant classifiers. A better understanding of approximate invariant classifiers could help to design an optimal nested Monte Carlo estimation following the approach of Rainforth et al. (2017). Future experimental work might be to investigate how abstract those invariant priors are to by investigating their transferability to other domains.

7. Conclusion

This paper support evidences that approximate invariant priors learned with only a fraction of the full dataset can get stable generalization improvements when learning the posterior of BNNs. Furthermore, posterior training with invariant prior is faster than posterior learning with online DA, and do not require to inflate the memory as with offline DA. Moreover, it is reported that offline DA is more stable than online DA.

References

Anselmi, F., Leibo, J. Z., Rosasco, L., Mutch, J., Tacchetti, A., and Poggio, T. Unsupervised learning of invariant representations with low sample complexity: the magic of sensory cortex or a new framework for machine learning? 2014. Publisher: Center for Brains, Minds and Machines (CBMM), arXiv.

Balan, A. K., Rathod, V., Murphy, K. P., and Welling, M. Bayesian dark knowledge. In *Advances in Neural Information Processing Systems*, pp. 3438–3446, 2015.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013. Publisher: IEEE.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Bousquet, O. and Elisseeff, A. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002.

Chen, T., Fox, E., and Guestrin, C. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pp. 1683–1691, 2014.

Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. EMNIST: Extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 113–123, 2019.

Dao, T., Gu, A., Ratner, A. J., Smith, V., De Sa, C., and RÃl', C. A kernel theory of modern data augmentation. *Proceedings of machine learning research*, 97:1528, 2019. Publisher: NIH Public Access.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Deng, L. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. Publisher: IEEE.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010.

Fawzi, A. and Frossard, P. Manitest: Are classifiers really invariant? *arXiv preprint arXiv:1507.06535*, 2015.

Federici, M., Ullrich, K., and Welling, M. Improved bayesian compression. *arXiv preprint arXiv:1711.06494*, 2017.

Gal, Y. Uncertainty in deep learning. *University of Cambridge*, 1:3, 2016.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. The MIT Press, 2016. ISBN 978-0-262-03561-3.

Graves, A. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pp. 2348–2356, 2011.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Ko, T., Peddinti, V., Povey, D., and Khudanpur, S. Audio augmentation for speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pp. 6402–6413, 2017.

LeCun, Y. Learning invariant feature hierarchies. In *European conference on computer vision*, pp. 496–505. Springer, 2012.

Leen, T. K. From data distributions to regularization in invariant learning. In *Advances in Neural information processing systems*, pp. 223–230, 1995.

Lemley, J., Bazrafkan, S., and Corcoran, P. Smart augmentation learning an optimal data augmentation strategy. *Ieee Access*, 5:5858–5869, 2017. Publisher: IEEE.

Lim, S., Kim, I., Kim, T., Kim, C., and Kim, S. Fast autoaugment. In *Advances in Neural Information Processing Systems*, pp. 6662–6672, 2019.

Louizos, C. and Welling, M. Structured and efficient variational deep learning with matrix gaussian posteriors. In *International Conference on Machine Learning*, pp. 1708–1716, 2016.

Louizos, C. and Welling, M. Multiplicative normalizing flows for variational bayesian neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2218–2227. JMLR. org, 2017.

Louizos, C., Ullrich, K., and Welling, M. Bayesian compression for deep learning. In *Advances in Neural Information Processing Systems*, pp. 3288–3298, 2017.

MacKay, D. J. *Bayesian methods for adaptive models*. PhD Thesis, California Institute of Technology, 1992.

Mescheder, L., Nowozin, S., and Geiger, A. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2391–2400. JMLR. org, 2017.

Molchanov, D., Ashukha, A., and Vetrov, D. Variational dropout sparsifies deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2498–2507. JMLR. org, 2017.

Nalisnick, E. and Smyth, P. Learning priors for invariance. In *International Conference on Artificial Intelligence and Statistics*, pp. 366–375, 2018.

Neklyudov, K., Molchanov, D., Ashukha, A., and Vetrov, D. P. Structured bayesian pruning via log-normal multiplicative noise. In *Advances in Neural Information Processing Systems*, pp. 6775–6784, 2017.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., and Antiga, L. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.

Pawlowski, N., Brock, A., Lee, M. C., Rajchl, M., and Glocker, B. Implicit weight uncertainty in neural networks. *arXiv preprint arXiv:1711.01297*, 2017.

Rainforth, T., Cornish, R., Yang, H., Warrington, A., and Wood, F. On nesting monte carlo estimators. *arXiv preprint arXiv:1709.06181*, 2017.

Ranzato, M., Huang, F. J., Boureau, Y.-L., and LeCun, Y. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *2007 IEEE conference on computer vision and pattern recognition*, pp. 1–8. IEEE, 2007.

Shi, J., Sun, S., and Zhu, J. Kernel implicit variational inference. *arXiv preprint arXiv:1705.10119*, 2017.

Shorten, C. and Khoshgoftaar, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019. Publisher: Springer.

Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., and Webb, R. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2107–2116, 2017.

Simard, P., Victorri, B., LeCun, Y., and Denker, J. Tangent prop-a formalism for specifying selected invariances in an adaptive network. In *Advances in neural information processing systems*, pp. 895–903, 1992.

Sokolic, J., Giryes, R., Sapiro, G., and Rodrigues, M. R. Generalization error of invariant classifiers. *arXiv preprint arXiv:1610.04574*, 2016.

Swendsen, R. H. and Wang, J.-S. Nonuniversal critical dynamics in Monte Carlo simulations. *Physical review letters*, 58(2):86, 1987. Publisher: APS.

Tanner, M. A. and Wong, W. H. The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association*, 82(398):528–540, 1987. Publisher: Taylor & Francis.

Tran, T., Pham, T., Carneiro, G., Palmer, L., and Reid, I. A bayesian data augmentation approach for learning deep models. In *Advances in neural information processing systems*, pp. 2797–2806, 2017.

Ullrich, K., Meeds, E., and Welling, M. Soft weight-sharing for neural network compression. *arXiv preprint arXiv:1702.04008*, 2017.

Van Dyk, D. A. and Meng, X.-L. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001. Publisher: Taylor & Francis.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., and Le, Q. V. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*, 2019.

A. Related Works

Bayesian neural networks can model uncertainty by learning a posterior distribution of the weights given the data. However the true posterior is often intractable, this has led to much technical development, Markov Chain Monte Carlo techniques (MCMC) afford the best theoretical guarantees to find the true posterior distribution, but are often impractical with big data as they need the full dataset to learn (Chen et al., 2014; Balan et al., 2015). Variational Inference is another method for Bayesian inference, the core idea is to use a parametric probability distribution, and to optimize

the KL between the posterior and the variational approximation. This technique has attracted a lot of attention recently, and variational approximation has been used in the context of BNNs. The first approaches used fully factorized Gaussian (Graves, 2011; Blundell et al., 2015). However, more sophisticated approximation distribution have been developed for variational posterior learning for instance Louizos & Welling (2016) used a full matrix, Gal & Ghahramani (2016) introduced Dropout as a Bayesian approximation. Later, Louizos & Welling (2017) used Normalizing flows to allow a more flexible posterior approximation. Recently other authors (Mescheder et al., 2017) have argued that implicit methods might offer more flexible approximation than invertible methods such as Normalizing Flows. For instance, Pawlowski et al. (2017) used a Hypernetwork to model the posterior, similarly, Shi et al. (2017) introduced a more efficient algorithm called Matrix Multiplication Neural Network (MMNN) to sample large matrix to model the posterior. In summary, there are many other approximation distributions that one could use (1) to learn approximate invariant priors and (2) for posterior learning.

One of the advantages of BNNs is their ability to leverage small datasets by incorporating prior knowledge. Prior knowledge can express a range of hypotheses about the data generating function. Moreover, several hypotheses have found a proper encoding directly into neural networks' weights, these include inducing compression: with sparsity over the weights (Molchanov et al., 2017; Neklyudov et al., 2017) , quantization (Ullrich et al., 2017) or a mix of both (Louizos et al., 2017; Federici et al., 2017).This paper focused on incorporating invariance to improve generalization following the work of Nalisnick & Smyth (2018). As the prior induce a regularization term in the expression of the ELBO, we see a link between prior learning and unsupervised pre-training (Erhan et al., 2010) as a form of semi-supervised learning in the non-Bayesian context. Both methods are applied before training and induce regularization. Unsupervised pre-training regularize by selecting initialization parameters taking into account the data distribution before the classification task. In other words, unsupervised pre-training initialize deep architectures in regions from which better basins of attraction can be reached, in terms of generalization improvements. Unsupervised pre-training favors hidden units that better explain informative factors of variation within the data X, and those factors of variation are useful in the task of predicting Y.

Several theoretical approaches have been provided about DA. The first who coined the term DA (Tanner & Wong, 1987) used it to simulate the posterior distribution in Bayesian inference. Their interpretation was to use augmented data as latent variables and provided an iterative MCMC algorithm. As remarked by (Van Dyk & Meng, 2001), the term evolved in the physics literature and was further referred to as the method of auxiliary variable, it has been used for instance to sample from the Ising and Post models (Swendsen & Wang, 1987). However, the technique of DA as it is used today in the machine learning literature has been well formulated by Dao et al. (2019) who have described its regularization effect by robustness induction with task invariance and model complexity penalization. Another line of research focuses on finding good or optimal augmentation schemes with the data at hand. Those augmentation policies rely on generative models (Tran et al., 2017; Lemley et al., 2017; Shrivastava et al., 2017) or reinforcement learning (Lim et al., 2019); (Cubuk et al., 2019). Searching good augmentation policies is similar to searching for uninformative factors of variation within the data. This can be seen as an orthogonal task to the extraction of informative factors of variation via semi-supervised learning. Furthermore, the combination of both approaches was successfully used to obtain state-of-the-art results in text classification and computer vision (Xie et al., 2019).

B. Additional experiments

B.1. Sample size and augmentation factor

This experiment investigated the importance of the sample size and the augmentation factor when learning approximate invariant priors. The augmentations were the same as in the experiment 5.1: rotation, noise, and rotation + noise with the original MNIST training set. Three sample sizes were used (256, 512, 1024) along with three augmentation factors (16, 32, 64). The models were trained for 2000 epochs with an early stopping after 500 epochs without progress. Finally, the annealing term for the entropy was $\frac{i}{2000}$, where i was the current epoch.

To evaluate learning, we used the whole 60,000 examples from the original MNIST training set and implemented a loader that sampled a batch of data X and the same batch augmented $\hat{X} \sim q(\hat{X}, X, \eta)$, with one set of weights from the learned prior $\theta \sim p_\lambda$ to compute the Expected KL. This procedure was performed on the whole dataset and iterated 10 times to have descriptive statistics.

The results of the experiment are summarized in figure 1. There is a clear pattern, for noise and rotation+noise, augmenting the sample size and the augmentation factor leads to better priors. However, we can see that this is not the case for rotation. What is surprising is that the network does not need the full dataset, nor infinite variation of the augmentation to learn invariant representations. This suggests that augmenting the full dataset might not be necessary to get the generalization improvement from DA. An additional question is: do these approximate invariant priors lead to a better generalization when learning posteriors? This is what was considered in section 5.1.

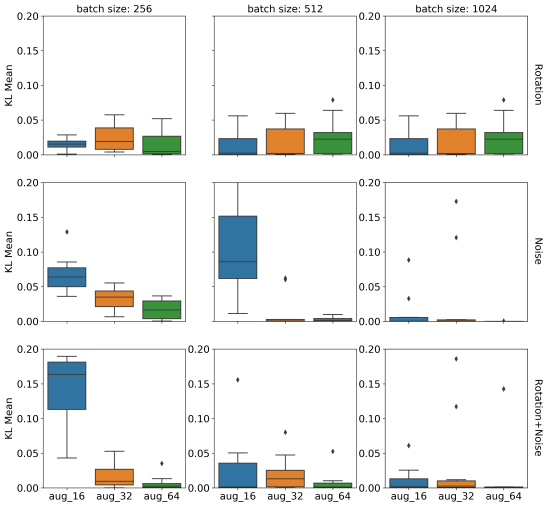

Figure 1: Comparison of different choices for the nested Monte Carlo approximation. Each column denotes a different sample size between 256, 512, and 1024. Each row refers to a different augmentation scheme: rotation, noise, and rotation+noise. Finally each plot as the same structure, different augmentation factors are compared between 16, 32, and 64. The y-axis has the same scale and computed the expected KL divergence.

B.2. Effect of invariance inducing technique on uncertainty

One weakness of neural networks is that they cannot model uncertainty, and this can lead to overconfident prediction even when they are wrong, this is known as overfitting. Uncertainty is critical if neural networks are deployed in applications such as medicine, autonomous vehicles, nuclear power plant, or high-frequency trading (Gal, 2016). BNNs can model uncertainty by learning a posterior distribution of the weights given the data, in the following experiment we evaluate the effect of approximate invariant priors, online and offline DA on posterior uncertainty after learning. What is typically done to evaluate uncertainty, is to train a model on a given dataset and evaluate the confidence of its prediction on another dataset involving different classes.

To do that, we used the networks trained in experiment 5.1 as in all cases except online DA with rotation+noise, the models have learned reasonably well on MNIST (see table 1). Furthermore, we evaluate them on the EMNIST dataset (Cohen et al., 2017) which involves letters instead of digits. The first ten letters of the alphabet: a-j were selected to emulate a ten class classification problem. The dataset has 4,800 examples per class which leads to a total of 48,000 examples for the ten classes. To evaluate the uncertainty in the prediction, the empirical cumulative distribution function (CDF) of the predictive entropy was used as in (Louizos & Welling, 2017).

The results are displayed in figure 2. To interpret these graph it is helpful to notice that the CDF of the predictive entropy is a global measure of uncertainty because it displays the proportion of items with high entropy within a given bound. For instance, in the case of the upper graph, all items from vanilla have entropy less or equal than 2, whereas only 25% of items have entropy less or equal than 2 in the case of online DA rotation. It means that 75% of items classified with the online DA rotation have entropy higher than 2. In other words, the network does not know how to classify the items from an unknown dataset, thus its output is close to a uniform distribution.

Overall, online and offline DA have a greater effect on the uncertainty of the trained posterior than the invariant prior, whatever the augmentation considered, moreover, all invariant inducing techniques display higher uncertainty than the vanilla BNN. Another detail to notice is that items which display the minimal amount of entropy depend not only on the invariance inducing technique but also on the augmentation. This is displayed with DA, for instance, if we consider offline DA, the minima entropy in the case of rotation is 0.5 but is around 1.25 in the case of rotation+noise. The same phenomenon is observed for online DA, which suggests that combining multiple augmentations directly on data increases the uncertainty of the model, yet this phenomenon is not observed for the BNN trained with an approximate invariant prior. Moreover, the only model that failed to learn MNIST: online DA trained with rotation+noise exhibit maximum uncertainty, it displays a uniform distribution for every item of EMNIST. To conclude, all invariant inducing techniques enhance the ability of a model to be uncertain, also, DA has a greater effect on model uncertainty than approximate invariant priors.

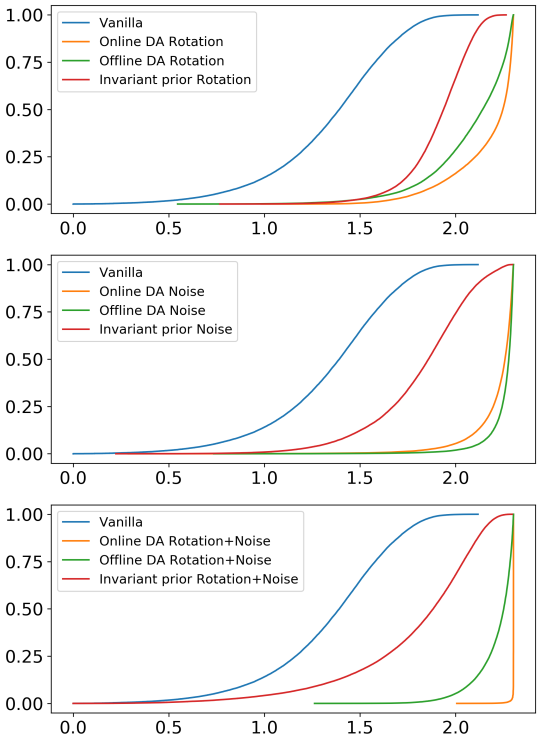

Figure 2: CDF comparison of predictive entropies. The graphs from up to bottom represent models trained respectively with rotation, noise and mix of rotation and noise.