Inverse-Reference Priors for Fisher Regularization of Bayesian Neural Networks

Keunseo Kim[1*], Eun-Yeol Ma[2], Jeongman Choi[2], Heeyoung Kim[2]

[1] Samsung Advanced Institute of Technology, Suwon, Republic of Korea
[2] Department of Industrial and Systems Engineering, KAIST, Daejeon, Republic of Korea
keunseo.kim@samsung.com, {eyma1127, jmchoi14, heeyoungkim}@kaist.ac.kr

Abstract

Recent studies have shown that the generalization ability of deep neural networks (DNNs) is closely related to the Fisher information matrix (FIM) calculated during the early training phase. Several methods have been proposed to regularize the FIM for increased generalization of DNNs. However, they cannot be used directly for Bayesian neural networks (BNNs) because the variable parameters of BNNs make it difficult to calculate the FIM. To address this problem, we achieve regularization of the FIM of BNNs by specifying a new suitable prior distribution called the inverse-reference (IR) prior. To regularize the FIM, the IR prior is derived as the inverse of the reference prior that imposes minimal prior knowledge on the parameters and maximizes the trace of the FIM. We demonstrate that the IR prior can enhance the generalization ability of BNNs for large-scale data over previously used priors while providing adequate uncertainty quantifications using various benchmark image datasets and BNN structures.

Introduction

The generalization of deep neural networks (DNNs) with a large number of parameters has received considerable attention in recent studies for complex data analysis (Neyshabur 2017; Jiang et al. 2019; Zhang et al. 2021). Widely used methods include implicit regularization techniques such as dropout (Srivastava et al. 2014), batch normalization (Ioffe and Szegedy 2015), and early stopping of training (LeCun et al. 2012), as well as explicit regularization techniques such as ℓ_2 regularization (Drucker and Le Cun 1992). However, for the past years it has been unclear why and how these regularization methods affect the generalization of DNNs.

Recent studies have found that the generalization of DNNs is highly affected by the local curvature of the loss function in the early training phase (Jastrzebski et al. 2019, 2021; Cohen et al. 2020). Jastrzebski et al. (2021) found that the trace of the Fisher information matrix (FIM) becomes excessively high during the early phase of training when the hyperparameters of stochastic gradient descent (SGD) are misspecified, which results in poor generalization. To prevent poor generalization, Jastrzebski et al. (2021) proposed the Fisher penalty, which regularizes the trace of the FIM. Previous methods that explicitly regularized the norm of the loss gradient (Varga, Csiszárik, and Zombori 2017; Barrett and Dherin 2020) can also be interpreted as regularizing the local curvature of the loss function.

Similar to DNNs, Bayesian neural networks (BNNs) can suffer from the poor generalization problem caused by the misspecification of the hyperparameters of SGD. However, the generalization issue has not been studied for BNNs, despite their wide use in risk-sensitive applications as they can quantify uncertainties in DNN predictions by placing prior distributions over the network weights (Jospin et al. 2022). BNNs are typically trained by maximizing the evidence lower bound (ELBO) using variational inference. However, the inherent noisy estimation of the ELBO during training makes the specification of the hyperparameters more critical for BNNs (Jospin et al. 2022). Moreover, their use of the ELBO for inference makes the direct application of the Fisher penalty (Jastrzebski et al. 2021) intractable.

Indeed, the generalization ability of BNNs can easily deteriorate for large-scale data depending on the specification of priors and the choice of hyperparameters such as the learning rate and batch size (Ghosh, Yao, and Doshi-Velez 2019; McGregor et al. 2019; Farquhar, Osborne, and Gal 2020). For instance, recent studies have shown that the Gaussian distribution, despite being the most frequently used, may be inappropriate as a prior distribution for BNNs (Nalisnick and Smyth 2018; Farquhar, Osborne, and Gal 2020) because it degrades the generalization of BNNs. In particular, Wenzel et al. (2020) pointed out that the Gaussian prior distribution is not suitable for BNNs because it is unintentionally informative and the unwanted effect of the prior amplifies as the network scale increases. We also empirically show in Section that the validation accuracy decreases significantly when a BNN with a Gaussian prior distribution is trained using a comparatively small learning rate. Although various priors have been proposed for better-performing BNNs (Ghosh, Yao, and Doshi-Velez 2019; McGregor et al. 2019; Farquhar, Osborne, and Gal 2020), identifying a prior that guarantees good generalization is still a problem to be solved.

In this study, we achieve regularization of the FIM of BNNs by specifying a new suitable prior distribution called the inverse-reference (IR) prior. Specifically, we derive the IR prior by manipulating the reference prior, inspired by the

[*] Work done while a student at KAIST.

fact that the closed-form reference prior is proportional to the determinant of the FIM (Consonni et al. 2018). The reference prior, which is a noninformative prior, maximizes the difference between a prior and the posterior (Bernardo 1979). The reference prior increases the trace of the FIM because it imposes minimal regularization on the model parameters by definition. Using this fact, we define the IR prior as the inverse of the reference prior to regularize the FIM.

The computation of the IR prior by taking the inverse of the reference prior is not tractable in general, because the closed-form of the reference prior is only available for the one-dimensional case. Instead, we compute the IR prior directly by obtaining a prior that minimizes the difference between the prior and posterior, inspired by the fact that the IR prior performs regularization in the direction opposite to that of the reference prior. Then, the IR prior is obtained as a prior that makes the prior and posterior distributions equal. Consequently, a variational posterior distribution can be used as an IR prior without an additional computational burden. We also provide a discussion on how the IR prior performs Fisher regularization.

We show that the IR prior helps BNNs achieve stable performance regardless of the model and hyperparameter specifications in Section . We also show that the IR prior can also be viewed as a suitable prior that improves the adversarial robustness of a BNN in Section . Here, adversarial robustness refers to the property of maintaining performance, even for adversarial examples, which are inputs intentionally crafted to deceive the model (Madry et al. 2018). It is widely known that BNNs, similar to DNNs, are vulnerable to adversarial examples (Goodfellow, Shlens, and Szegedy 2014; Yuan, Wicker, and Laurenti 2020). We describe the relationship between the FIM and the robustness of a BNN against adversarial examples and provide a detailed explanation of how the IR prior improves the adversarial robustness.

In summary, the contributions of this study are as follows:

- We propose a new prior, called the IR prior, that enhances the generalization ability of BNNs by regularizing the FIM.
- We validate the generalization ability of BNNs in terms of the validation accuracy and adversarial robustness compared to previously studied priors (Ghosh, Yao, and Doshi-Velez 2019; McGregor et al. 2019; Farquhar, Osborne, and Gal 2020) using various benchmark image datasets and BNN structures.

Background

Bayesian Neural Networks (BNNs)

A BNN is a probabilistic version of a neural network with a prior distribution on network weights. BNNs are trained by Bayesian inference, which estimates the posterior distribution of the network weights conditional on the data. To formulate the BNNs, we denote a training dataset consisting of n observations of random variables $(\mathbf{x}, y)$ as $\mathcal{D} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_n$ is the nth input feature and y_n is the corresponding label. We also denote the DNN embedded in the BNN as $f(\mathbf{x}, \theta)$, where θ is the set of the network weights with a prior distribution $p(\theta)$. Let $\ell(\mathbf{x}, y; \theta)$ denote the cross-entropy loss calculated for the input $\mathbf{x}$ and label y. We then define the log-likelihood for observing y given $\mathbf{x}$ and θ as the negative cross-entropy loss, i.e., $\log p(y|\mathbf{x}, \theta) = -\ell(\mathbf{x}, y; \theta)$. Bayesian inference estimates the posterior distribution $p(\theta|\mathcal{D})$, which can be obtained using Bayes' rule. However, the direct calculation of the posterior distribution is intractable, and variational inference is instead widely used for the inference of BNNs (Blundell et al. 2015; Nazarovs et al. 2021). Variational inference uses a variation posterior distribution $q_\phi(\theta)$ with variational parameters ϕ, which approximates the true posterior distribution $p(\theta|\mathcal{D})$. The variational parameters are estimated by maximizing the evidence lower bound (ELBO):

$$\begin{aligned} \log p(y|\mathbf{x}) &\geq E_{q_\phi(\theta)}[\log p(y|\mathbf{x}, \theta)] - KL(q_\phi(\theta)||p(\theta)) \\ &= ELBO, \end{aligned} \tag{1}$$

where KL denotes the Kullback-Leibler divergence.

Reference Prior

A reference prior is a type of noninformative prior that can be useful in the absence of prior information. Bernardo (1979) defined the reference prior as a prior that maximizes the KL divergence between the prior and the posterior, $E\left[\log \frac{p(\theta|\mathcal{D})}{p(\theta)}\right]$, and can be interpreted as a prior that maximizes the mutual information between the parameters and data as follows:

$$\begin{aligned} p^*(\theta) &= \arg\max_{p(\theta)} \int\int \log \frac{p(\theta|\mathcal{D})}{p(\theta)} p(\theta|\mathcal{D}) m(\mathcal{D}) d\theta d\mathcal{D} \\ &= \arg\max_{p(\theta)} I(\theta, \mathcal{D}), \end{aligned} \tag{2}$$

where $I(\theta, \mathcal{D})$ denotes the mutual information between θ and $\mathcal{D}$, and $m(\mathcal{D})$ denotes the marginal distribution of $\mathcal{D}$. The FIM is defined as the covariance of the score function, which is the gradient of the log-likelihood, as follows:

$$\text{FIM}(\theta) = E\left[\nabla_\theta \log p(y|\mathbf{x}, \theta) \nabla_\theta \log p(y|\mathbf{x}, \theta)^T\right]. \tag{3}$$

Bernardo (1979) showed that the reference prior is proportional to the positive square root of the determinant of the FIM, $p^*(\theta) \propto |\text{FIM}(\theta)|^{1/2}$.

Fisher Regularization

In general, explicit regularization methods introduce a specific regularization term to the objective function. For example, the ℓ_1 and ℓ_2 norms of the parameters have been widely used as regularization terms. Recently, gradient regularization, which regularizes the gradient norm of the likelihood with respect to its inputs, has gained significant attention (Drucker and Le Cun 1992; Czarnecki et al. 2017; Gulrajani et al. 2017; Varga, Csiszárik, and Zombori 2017). Specifically, gradient regularization penalizes the squared ℓ_p norm of the gradient of the likelihood, given by

$$\|\frac{\partial}{\partial \mathbf{x}} \log p(y|\mathbf{x}, \theta)\|_p^2. \tag{4}$$

Gradient regularization forces the model to produce similar outputs for nearby inputs. Thus, gradient regularization prevents the rapid change in the outputs in some directions and enforces the smoothness of the model outputs against the input noise (Varga, Csiszárik, and Zombori 2017).

Recently, Jastrzebski et al. (2021) proposed a regularization method that enforces smoothness in the optimization trajectory of the parameters. Jastrzebski et al. (2021) found that generalization of the DNNs is closely related to the optimization trajectory, which is affected by the local curvature of the loss surface. To regularize the optimization trajectory, Jastrzebski et al. (2021) introduced Fisher regularization, which regularizes the trace of the FIM, given by

$$Tr(\text{FIM}(\theta)) = E_{y^* \sim p(y^*|\mathbf{x},\theta)} \left[\| \frac{\partial}{\partial \theta} \log p(y^*|\mathbf{x},\theta) \|_2^2 \right], \tag{5}$$

where Tr denotes the trace.

Inverse-Reference Priors for Fisher Regularization

Previous studies have shown that Fisher regularization effectively enhances the generalization of DNNs and results in state-of-the-art generalization performance compared to other regularization methods. However, it is difficult to apply Fisher regularization directly to BNNs, because the parameters of BNNs are assumed to be random variables, which makes it difficult to differentiate the likelihood with respect to the parameters. Instead, we achieve Fisher regularization by imposing a suitable prior on the network parameters, rather than directly calculating the FIM.

We propose a new prior distribution, called the inverse-reference (IR) prior, that naturally achieves Fisher regularization for BNNs. We compute the IR prior by minimizing the mutual information between the parameters and data, based on the fact that the IR prior performs Fisher regularization as opposed to the reference prior that maximizes the mutual information, as follows:

$$\begin{aligned} p^{IR}(\theta) &= \arg\min_{p(\theta)} \int \int \log \frac{p(\theta|\mathcal{D})}{p(\theta)} p(\theta|\mathcal{D}) m(\mathcal{D}) d\theta d\mathcal{D} \\ &= \arg\min_{p(\theta)} E\left[KL(p(\theta|\mathcal{D})||p(\theta)) \right]. \end{aligned} \tag{6}$$

Because the KL divergence term in the last equality of Eq. (6) is always non-negative, we can solve the optimization problem by finding a prior that makes the KL divergence equal to zero. Then, the solution becomes the prior that makes the posterior distribution and the prior distribution equal, i.e., $p^{IR}(\theta) = p(\theta|\mathcal{D})$.

However, we cannot obtain the true posterior distribution $p(\theta|\mathcal{D})$ directly during training of BNNs. To circumvent the direct computation of the posterior distribution, we use a variational distribution q_ϕ as an approximation of the true posterior distribution as follows:

$$p^{IR}(\theta) = p(\theta|\mathcal{D}) = q_{\phi^*}(\theta), \tag{7}$$

where ϕ^* is obtained by maximizing the ELBO given by

$$ELBO = E_{q_\phi}\left[\log p(\mathcal{D}|\theta)\right] - KL(q_\phi(\theta)||p^{IR}(\theta)). \tag{8}$$

The training algorithm for a BNN with the IR prior is summarized in Algorithm 1.

The ELBO in Eq.(8) aids understanding of how the IR prior approximated by $q_{\phi^*}(\theta)$ regularizes the FIM. Let q_{ϕ^t} denote the variational distribution updated at training iteration t in the training process. At training iteration $t+1$, with $p^{IR}(\theta) = q_{\phi^t}(\theta)$, we find the variational distribution $q_{\phi^{t+1}}(\theta)$ that maximizes the ELBO given by

$$E_{q_{\phi^{t+1}}}\left[\log p(\mathcal{D}|\theta)\right] - KL(q_{\phi^{t+1}}(\theta)||q_{\phi^t}(\theta)), \tag{9}$$

where the KL divergence term can be reexpressed as

$$KL(q_{\phi^{t+1}}(\theta)||q_{\phi^t}(\theta)) = E_{q_{\phi^{t+1}}}\left[\log q_{\phi^{t+1}}(\theta) - \log q_{\phi^t}(\theta)\right]. \tag{10}$$

Using the Taylor expansion of $\log q_{\phi^{t+1}}(\theta)$ around $\phi^{t+1} = \phi^t$, we can approximate $KL(q_{\phi^{t+1}}(\theta)||q_{\phi^t}(\theta))$ in Eq.(10) as

$$(\Delta\phi)^T \text{FIM}(\phi^t)(\Delta\phi), \tag{11}$$

where $\text{FIM}(\phi^t) = E_{q_{\phi^t}}\left[(\nabla_{\phi^t} \log q_{\phi^t}(\theta))(\nabla_{\phi^t} \log q_{\phi^t}(\theta))^T\right]$ is the FIM of q_{ϕ^t} and $\Delta\phi = \phi^{t+1} - \phi^t$. From Eq. (8) to Eq.(11), we can see that the ELBO can be represented as the expectation of the log-likelihood regularized by the FIM of q_ϕ. We can also easily show the convergence of the training process for BNNs with the IR prior because the KL divergence term goes to zero as $q_{\phi^{t+1}}(\theta)$ approaches $q_{\phi^t}(\theta)$.

Experimental Evaluation

In this section, we evaluate the performance of the BNNs with the IR prior. Specifically, we analyze the results of experiments conducted using various benchmark datasets to demonstrate that the IR prior effectively reduces the trace of the FIM during training and improves the validation accuracy.

Experimental Setup

Datasets and Neural Network Architectures We used three benchmark image datasets, CIFAR-10, CIFAR-100, and SVHN, for the experiments. Each image was resized to 32 $\times$ 32 pixels with three channels. For the architectures of the BNNs, we employed three well-known image classification neural network structures: DenseNet (Huang et al. 2017), ResNet (He et al. 2016), and VGG (Liu and Deng 2015). Considering the size of the datasets, we used the DenseNet121, ResNet18, and VGG16 structures, which utilize approximately 7.8 million, 11 million, and 138 million parameters, respectively. We considered structures of various scales to verify whether the IR prior exhibits effective performance regardless of the scale of the BNNs.

Implementation Details We considered the same experimental settings as those in Nazarovs et al. (2021). To estimate the KL divergence term in the ELBO in Eq. (1), we used the graph reparameterization trick (Nazarovs et al. 2021), which is a scalable method to compute the ELBO of large-scale BNNs. We used a radial distribution (Farquhar, Osborne, and Gal 2020) as the variational posterior distribution and used the Adam optimizer (Kingma and Ba 2014)

Algorithm 1: Algorithm for training the BNN with the IR prior

Input: Input model $p(\mathcal{D}|\theta)$, variational distribution $q_\phi(\theta)$, number of iterations T
Output: Variational distribution $q_{\phi^T}(\theta)$
Initialize the variational parameters $\phi = \phi^0$ and prior distribution $p^{IR}(\theta) = q_{\phi^0}(\theta)$.
for $t = 0, \ldots, T-1$ iterations **do**
    Update ϕ^t to ϕ^{t+1} by maximizing the ELBO in Eq.(9).
    Update $p^{IR}(\theta) = q_{\phi^{t+1}}(\theta)$.
end for

with learning rate set to 0.001.

Alternative Priors for Comparison We employed three previously studied priors for large-scale BNNs as baselines for comparison: the horseshoe prior (Ghosh, Yao, and Doshi-Velez 2019), self-stabilizing prior (McGregor et al. 2019), and the classical Gaussian prior.

Experimental Results

Validation Accuracy Comparison with Other Priors We evaluated the validation accuracy of the BNNs with the four considered priors using the CIFAR-10, SVHN, and CIFAR-100 datasets. We compared the average validation accuracy over five repeated experiments using different initial values of the BNNs in Table 1 with standard errors in parentheses. We employed the validation accuracy as a metric of the generalization ability because it measures the performance on unseen data points. The validation accuracy reported in Table 1 represents the maximum validation accuracy recorded during 100 training epochs. The three image classification models, DenseNet121, ResNet18, and VGG16, were used to verify the performance of the BNNs with various scales of embedded neural networks.

In all the experiments, the IR prior achieved the best generalization performance. For example, when tested on the CIFAR-10 dataset, the BNN with the IR prior achieved the best validation accuracy of 85.5%, 84.6%, and 84.1% using DenseNet121, ResNet18, and VGG16, respectively, outperforming the other models. Similarly, the BNN with the IR prior achieved the best validation accuracy when tested on SVHN and CIFAR-100 as well (Table 1).

In addition, the IR prior showed the most stable validation accuracy in terms of standard error among the priors considered. For example, the standard error of validation accuracy of the DenseNet121-embedded BNN with the IR prior on the CIFAR-10 dataset (Table 1) was 0.49. In contrast, the standard errors of validation accuracy were 0.55, 0.86, and 0.60 when using the Gaussian, self-stabilizing, and horseshoe priors, respectively. Consistently, the standard errors of the validation accuracy were the lowest for the BNNs with the IR prior for all architectures and datasets considered.

Robustness against Hyperparameter Misspecification The generalization ability of DNNs easily deteriorates depending on the network architecture and specification of hyperparameters, such as the learning rate (Jastrzebski et al. 2021). Through additional experiments, we show that the same problem occurs with BNNs. Moreover, we empirically show that Fisher regularization with the IR prior can effectively solve this problem.

Figure 1 shows the validation accuracy of the BNNs based on DenseNet121 and ResNet18 with various priors trained using different learning rates. Regardless of the network architecture and learning rate, the BNN with the IR prior showed the best validation accuracy among the BNNs considered, suggesting that using the IR prior provides robust estimation regardless of model specifications. Even when the learning rate was misspecified as extremely low (e.g., 0.00001 or 0.00005), the BNN with the IR prior consistently achieved better validation accuracy than the BNNs with other priors. Furthermore, the BNN with the IR prior performed well for both relatively small (DenseNet121) and big (ResNet18) network capacities.

For an in-depth analysis, we analyzed the maximum value of the trace of the FIM during training with various learning rate values. The left and right panels of Figure 2 show the results of the ResNet18-embedded BNN and the DenseNet121-embedded BNN, respectively. As expected, for both ResNet- and DenseNet-embedded BNNs, the maximum value of the trace of the FIM increased as the learning rate decreased for all priors considered. However, whereas the trace of the FIM exploded when a small learning rate was used for all competing priors (Gaussian, self-stabilizing, horseshoe), the increase in the trace was significantly suppressed when the IR prior was used (green). The effective regularization of the trace of the FIM using the IR prior enabled good validation accuracy even with a small learning rate, as shown in Figure 1.

Furthermore, we compare the IR and the Gaussian priors in terms of the validation accuracy and trace of the FIM over different training epochs in Figure 3. As shown in the left panel, the BNN with the Gaussian prior exhibited significantly different performances depending on the specification of the learning rate (violet, cyan). In particular, the Gaussian prior resulted in a rapid increase in the trace of the FIM in the early training phase (right panel), which may have resulted in the inferior validation accuracy when the learning rate was as low as 0.0001 (left panel). In contrast, when the IR prior (orange) was used, the trace of the FIM remained stable even with a small learning rate of 0.0001 (right panel), which probably resulted in a significantly higher validation accuracy (left panel).

Uncertainty Quantification One of the main advantages of using BNNs is their ability to quantify uncertainty about

Priors	CIFAR-10 DenseNet	CIFAR-10 ResNet	CIFAR-10 VGG	CIFAR-100 DenseNet	CIFAR-100 ResNet	CIFAR-100 VGG	SVHN DenseNet	SVHN ResNet	SVHN VGG
Horseshoe	76.5 (0.60)	76.4 (1.01)	77.9 (0.69)	41.2 (0.80)	41.5 (0.91)	43.3 (0.82)	91.9 (0.32)	92.1 (0.35)	91.7 (0.45)
Self-stabilizing	77.2 (0.86)	76.4 (0.50)	79.4 (0.72)	47.9 (0.58)	47.6 (0.78)	47.3 (0.74)	92.5 (0.40)	91.8 (0.26)	91.8 (0.21)
Gaussian	81.9 (0.55)	82.6 (0.49)	81.1 (0.53)	48.3 (0.52)	48.9 (0.62)	48.1 (0.59)	94.9 (0.08)	94.8 (0.10)	94.3 (0.11)
IR	**85.5** (**0.49**)	**84.6** (**0.49**)	**84.1** (**0.48**)	**52.0** (**0.39**)	**51.1** (**0.28**)	**50.8** (**0.53**)	**96.1** (**0.08**)	**95.4** (**0.08**)	**95.4** (**0.08**)

Table 1: Validation accuracy (%) of the BNNs based on DenseNet121, ResNet18, and VGG16 tested on various datasets

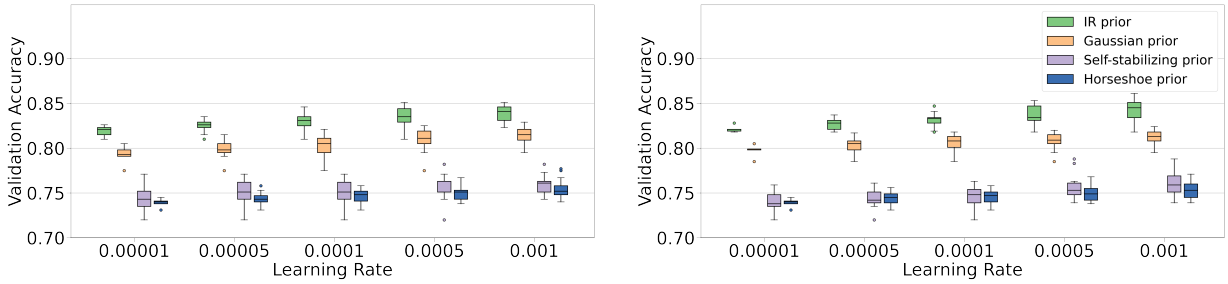

Figure 1: Validation accuracy on the CIFAR-10 dataset obtained using the ResNet18-embedded BNN (left panel) and the DenseNet121-embedded BNN (right panel) with various prior distributions. We measured the validation accuracy with various learning rates.

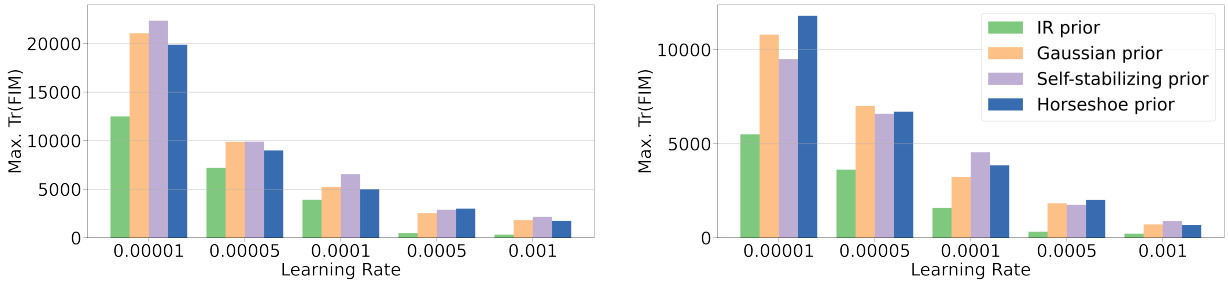

Figure 2: The maximum value of the trace of the FIM on the CIFAR-10 dataset obtained using the ResNet18-embedded BNN (left panel) and the DenseNet121-embedded BNN (right panel) with various prior distributions. We measured the maximum value of the trace with various learning rates.

their predictions. Figure 4 shows the epistemic uncertainty quantified by a 3-layer BNN with the Gaussian prior (orange) and a 3-layer BNN with the IR prior (green) on two toy regression datasets. Both networks similarly underestimated uncertainties as expected for BNNs trained with mean-field variational inference (Foong et al. 2019), while the BNN with the IR prior quantified the uncertainty to be slightly larger for unseen regions compared to the BNN with the Gaussian prior for both datasets. The IR prior tended to provide at least as good predictive uncertainty quantification as the popular Gaussian prior. Considering the superior predictive performance of the BNN with the IR prior (Table 1), the IR prior can be considered a well-suited prior that yields both good predictive performance and uncertainty quantification.

IR Priors Increase Robustness against Adversarial Attacks

In this section, we briefly describe the relationship between Fisher regularization using the IR prior and the adversarial robustness. In general, it is desirable that a model yields similar results for a clean sample and its perturbed version. However, it is well known that DNNs generally produce very different outputs when confronted with adversarial examples (Kurakin, Goodfellow, and Bengio 2016). To improve the

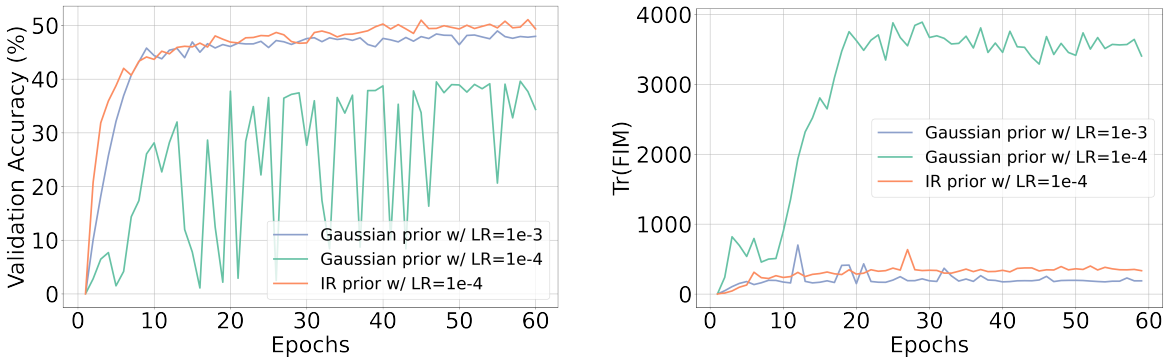

Figure 3: The validation accuracy (left panel) and the trace of the FIM (right panel) obtained using the ResNet18-embedded BNNs with the IR and Gaussian priors on the CIFAR-10 dataset with different learning rates (LR) over training epochs.

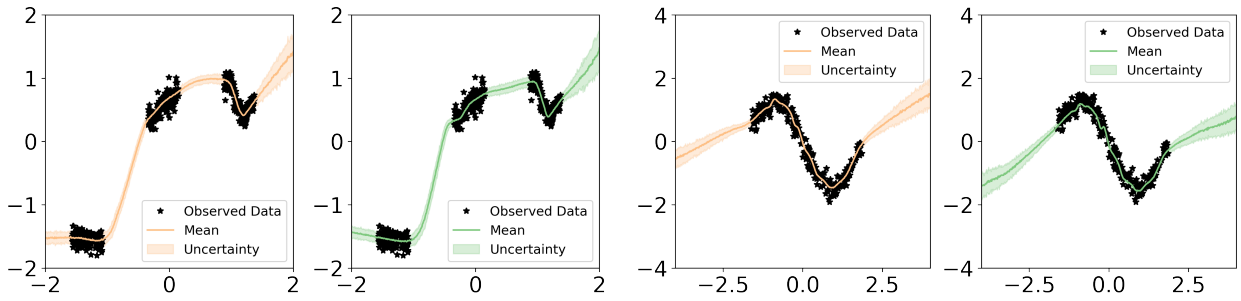

Figure 4: The epistemic uncertainty quantified by the BNNs with the Gaussian prior (orange) and IR prior (green) on two toy regression datasets.

adversarial robustness of DNNs, many studies have investigated the factors that affect adversarial robustness. In particular, Moosavi-Dezfooli et al. (2019) and Yu et al. (2018) showed that the trace of the FIM has a significant impact on adversarial robustness.

Adversarial robustness is defined as the difference in performance between a model with pure examples and a model with adversarial examples. Suppose that an adversarial example $\mathbf{x}'$ is generated by adding the ℓ_p norm-bounded perturbation δ to the pure example $\mathbf{x}$. The adversarial perturbation δ can be defined as follows:

$$\arg\min_{\delta} \log p(y|\mathbf{x}+\delta,\theta), \quad \text{where} \quad \|\delta\|_p < \epsilon, \tag{12}$$

where ϵ is a perturbation radius that controls the size of perturbation. Then, adversarial robustness can be measured as

$$\log p(y|\mathbf{x}',\theta) - \log p(y|\mathbf{x},\theta). \tag{13}$$

Using the Taylor expansion of $\log p(y|\mathbf{x}',\theta)$ around $\mathbf{x}'=\mathbf{x}$, we can approximate Eq. (13) as

$$\nabla \log p(y|\mathbf{x},\theta)^T\delta + \frac{1}{2}\delta^T \text{FIM}(\theta)\delta, \tag{14}$$

where $\delta = \mathbf{x}' - \mathbf{x}$ is the adversarial attack size. Note that in Eq.(14), the 0th order term disappears because it is canceled out by the negative 0th order term in Eq.(13). When the model is fully trained, the expectation of the gradient of the log-likelihood $\nabla \log p(y|\mathbf{x},\theta)$ converges to 0, simplifying Eq. (14) as $\delta^T\text{FIM}(\theta)\delta$. The $\hat{\delta}$ that maximizes the adversarial attack, i.e., $\hat{\delta} = \arg\max_\delta \delta^T\text{FIM}(\theta)\delta$, can be obtained by taking the derivative of the Lagrangian of $\delta^T\text{FIM}(\theta)\delta$ and equating it to 0. This yields $\text{FIM}(\theta)\hat{\delta} = \lambda\hat{\delta}$, where the Lagrangian λ is the eigenvector of $\text{FIM}(\theta)$ by definition. The adversarial attack $\hat{\delta}^T\text{FIM}(\theta)\hat{\delta}$ then becomes the maximum of the eigenvalues, which can be easily calculated through the trace of $\text{FIM}(\theta)$. Therefore, the IR prior regularizing the trace of the FIM can also be interpreted as the prior improving the adversarial robustness.

We conducted additional experiments to evaluate the robustness of using the IR prior against adversarial attacks. To evaluate the robustness, we evaluated the classification accuracy of BNNs for adversarial examples. We generated adversarial examples using the fast gradient sign method (Goodfellow, Shlens, and Szegedy 2014) by adding bounded perturbations of size $\epsilon = 0.001$ and $\epsilon = 0.0001$ to pure test examples.

We compared the mean (standard error) validation accuracy of the BNNs with the four different priors over five repeated experiments considered in Section using the adversarially manipulated CIFAR-10 and CIFAR-100 datasets

ϵ Size	Horseshoe	Self-stabilizing	Gaussian	IR
0.0001	75.6 (0.67)	76.7 (0.81)	80.1 (0.78)	**84.9** (**0.50**)
0.001	74.5 (0.71)	75.1 (0.86)	78.9 (0.65)	**84.7** (**0.70**)

Table 2: Validation accuracy (%) of DenseNet-embedded BNNs on CIFAR-10 dataset with adversarial perturbations of different sizes of $\epsilon = 0.0001$ and $\epsilon = 0.001$.

ϵ Size	Horseshoe	Self-stabilizing	Gaussian	IR
0.0001	39.8 (0.31)	45.4 (0.28)	47.6 (0.32)	**50.2** (**0.29**)
0.001	39.1 (0.55)	44.3 (0.33)	46.1 (0.41)	**49.9** (**0.35**)

Table 3: Validation accuracy (%) of DenseNet-embedded BNNs on CIFAR-100 dataset with adversarial perturbations of different sizes of $\epsilon = 0.0001$ and $\epsilon = 0.001$.

in Tables 2 and 3, respectively. For both datasets, the IR prior resulted in the highest accuracy among the priors. With the larger adversarial perturbation ($\epsilon = 0.001$), the accuracy decreased for all considered priors, as expected. For example, the accuracy with the IR prior decreased from 84.9% to 84.7% when the perturbation size increased from $\epsilon = 0.0001$ to $\epsilon = 0.001$. However, the decrease in the accuracy due to the larger perturbation was the smallest for the IR prior.

Related Work

Previous studies have highlighted that a right prior that reflects the characteristics of both the network structure and task purpose is essential in improving the predictive performance of the BNNs (Fortuin et al. 2022; Fortuin 2022). In general, Gaussian distributions have been a popular prior over the parameters of BNNs. By assuming independence between the parameters, a simple Gaussian prior can be introduced as $p(\theta_i) = \mathcal{N}(\theta_i; 0, \sigma^2)$, which can also be interpreted as the ℓ_2 norm regularization of the parameters. For more flexible prior distributions, Blundell et al. (2015) proposed a scale mixture prior, which is a mixture of two Gaussian distributions with varying standard deviations ($\sigma_1 \geq \sigma_2$ and $\sigma_2 \ll 1$). A near-zero choice of σ_2 forces the network weights to be near zero, which can be interpreted as imposing dropout (Srivastava et al. 2014).

However, recent studies have found that the typical choice of Gaussian priors can be problematic in many cases. For example, Gaussian priors may overinform the network, especially when the data size is small (Ghosh, Yao, and Doshi-Velez 2019). Moreover, Gaussian priors can result in high variance estimation, as BNNs are typically highly sensitive to prior and hyperparameter specifications (McGregor et al. 2019). Furthermore, conventional mean-field variational inference of Gaussian posteriors results in unrepresentative weights sampled from a "soap-bubble" or a narrow region distant from the mean, resulting in rapid deterioration of the validation performance of large-scale BNNs (Farquhar, Osborne, and Gal 2020).

To address these problems, various priors for BNNs have been proposed. Ghosh, Yao, and Doshi-Velez (2019) proposed the horseshoe prior as a shrinkage prior to alleviate the overparametrization in BNNs. McGregor et al. (2019) proposed the self-stabilizing prior, which updates the prior per gradient step to preserve the variance during the forward signal propagation and achieve robust training. Farquhar, Osborne, and Gal (2020) proposed the radial BNN, which samples network weights from a distribution in a hyperspherical coordinate system that ensures high probability densities around the mean. However, despite their efforts to solve specific problems induced by Gaussian priors, no prior study has addressed the essential problem of large-scale BNNs suffering from a lack of generalization ability. While noninformative priors may be the most appropriate priors when prior knowledge is insufficient (Nalisnick and Smyth 2017), no suitable prior for BNNs has yet been proposed to enhance generalization without imposing unnecessary information on the network parameters.

Conclusion

We proposed a new prior distribution for BNNs, called the IR prior, which achieves Fisher regularization and thus increases the generalization ability of BNNs. The IR prior was derived to regularize the FIM based on the fact that the closed-form reference prior is proportional to the determinant of the FIM. We empirically verified that the IR prior achieves superior validation accuracy compared with previously proposed priors through experiments using BNNs with DenseNet121, ResNet18, and VGG16 on the CIFAR-10, CIFAR-100, and SVHN datasets. Moreover, through experiments with various learning rate values, we demonstrated that the IR prior can effectively perform Fisher regularization, reducing the trace of the FIM in the training phase. In this study, we approximated the reference prior as a variational distribution. In future research, we will theoretically study the approximation error caused by the difference between the true and variational posterior distributions.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2018R1C1B6004511,2020R1A4A10187747).

References

Barrett, D.; and Dherin, B. 2020. Implicit Gradient Regularization. In *International Conference on Learning Representations*.

Bernardo, J. M. 1979. Reference posterior distributions for Bayesian inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2): 113–128.

Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; and Wierstra, D. 2015. Weight uncertainty in neural network. In *In-*

ternational Conference on Machine Learning, 1613–1622. PMLR.

Cohen, J.; Kaur, S.; Li, Y.; Kolter, J. Z.; and Talwalkar, A. 2020. Gradient Descent on Neural Networks Typically Occurs at the Edge of Stability. In *International Conference on Learning Representations*.

Consonni, G.; Fouskakis, D.; Liseo, B.; and Ntzoufras, I. 2018. Prior distributions for objective Bayesian analysis. *Bayesian Analysis*, 13(2): 627–679.

Czarnecki, W. M.; Osindero, S.; Jaderberg, M.; Swirszcz, G.; and Pascanu, R. 2017. Sobolev training for neural networks. *Advances in Neural Information Processing Systems*, 30.

Drucker, H.; and Le Cun, Y. 1992. Improving generalization performance using double backpropagation. *IEEE Transactions on Neural Networks*, 3(6): 991–997.

Farquhar, S.; Osborne, M. A.; and Gal, Y. 2020. Radial Bayesian neural networks: beyond discrete support in large-scale Bayesian deep learning. In *International Conference on Artificial Intelligence and Statistics*, 1352–1362. PMLR.

Foong, A. Y.; Li, Y.; Hernández-Lobato, J. M.; and Turner, R. E. 2019. 'In-Between'Uncertainty in Bayesian Neural Networks. In *Workshop on Uncertainty and Robustness in Deep Learning*.

Fortuin, V. 2022. Priors in bayesian deep learning: A review. *International Statistical Review*.

Fortuin, V.; Garriga-Alonso, A.; Ober, S. W.; Wenzel, F.; Ratsch, G.; Turner, R. E.; van der Wilk, M.; and Aitchison, L. 2022. Bayesian Neural Network Priors Revisited. In *International Conference on Learning Representations*.

Ghosh, S.; Yao, J.; and Doshi-Velez, F. 2019. Model Selection in Bayesian Neural Networks via Horseshoe Priors. *Journal of Machine Learning Research*, 20(182): 1–46.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of Wasserstein GANs. *Advances in Neural Information Processing Systems*, 30.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, 630–645. Springer.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 4700–4708.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 448–456. PMLR.

Jastrzebski, S.; Arpit, D.; Astrand, O.; Kerg, G. B.; Wang, H.; Xiong, C.; Socher, R.; Cho, K.; and Geras, K. J. 2021. Catastrophic fisher explosion: Early phase fisher matrix impacts generalization. In *International Conference on Machine Learning*, 4772–4784. PMLR.

Jastrzebski, S.; Szymczak, M.; Fort, S.; Arpit, D.; Tabor, J.; Cho, K.; and Geras, K. 2019. The Break-Even Point on Optimization Trajectories of Deep Neural Networks. In *International Conference on Learning Representations*.

Jiang, Y.; Neyshabur, B.; Mobahi, H.; Krishnan, D.; and Bengio, S. 2019. Fantastic Generalization Measures and Where to Find Them. In *International Conference on Learning Representations*.

Jospin, L. V.; Laga, H.; Boussaid, F.; Buntine, W.; and Bennamoun, M. 2022. Hands-on Bayesian neural networks—A tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17(2): 29–48.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial machine learning at scale. In *International Conference on Learning Representations*.

LeCun, Y. A.; Bottou, L.; Orr, G. B.; and Müller, K.-R. 2012. Efficient backprop. In *Neural networks: Tricks of the trade*, 9–48. Springer.

Liu, S.; and Deng, W. 2015. Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian conference on pattern recognition (ACPR)*, 730–734. IEEE.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

McGregor, F.; Pretorius, A.; Preez, J. d.; and Kroon, S. 2019. Stabilising priors for robust Bayesian deep learning. *arXiv preprint arXiv:1910.10386*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Uesato, J.; and Frossard, P. 2019. Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9078–9086.

Nalisnick, E.; and Smyth, P. 2017. Learning approximately objective priors. *arXiv preprint arXiv:1704.01168*.

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

Nazarovs, J.; Mehta, R. R.; Lokhande, V. S.; and Singh, V. 2021. Graph reparameterizations for enabling 1000+ Monte Carlo iterations in Bayesian deep neural networks. In *Uncertainty in Artificial Intelligence*, 118–128. PMLR.

Neyshabur, B. 2017. Implicit regularization in deep learning. *arXiv preprint arXiv:1709.01953*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1): 1929–1958.

Varga, D.; Csiszárik, A.; and Zombori, Z. 2017. Gradient regularization improves accuracy of discriminative models. *arXiv preprint arXiv:1712.09936*.

Wenzel, F.; Roth, K.; Veeling, B.; Swiatkowski, J.; Tran, L.; Mandt, S.; Snoek, J.; Salimans, T.; Jenatton, R.; and

Nowozin, S. 2020. How Good is the Bayes Posterior in Deep Neural Networks Really? In *International Conference on Machine Learning*, 10248–10259. PMLR.

Yu, F.; Liu, C.; Wang, Y.; Zhao, L.; and Chen, X. 2018. Interpreting adversarial robustness: A view from decision surface in input space. *arXiv preprint arXiv:1810.00144*.

Yuan, M.; Wicker, M.; and Laurenti, L. 2020. Gradient-Free Adversarial Attacks for Bayesian Neural Networks. *arXiv preprint arXiv:2012.12640*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2021. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3): 107–115.