CONFIDENCE CALIBRATION IN DEEP NEURAL NETWORKS THROUGH STOCHASTIC INFERENCES

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a generic framework to calibrate accuracy and confidence (score) of a prediction through stochastic inferences in deep neural networks. We first analyze relation between variation of multiple model parameters for a single example inference and variance of the corresponding prediction scores by Bayesian modeling of stochastic regularization. Our empirical observation shows that accuracy and score of a prediction is highly correlated with variance of multiple stochastic inferences given by stochastic depth or dropout. Motivated by these facts, we design a novel variance-weighted confidence-integrated loss function that is composed of the standard cross-entropy loss and KL-divergence from uniform distribution, where the two terms are balanced based on variance of stochastic prediction scores. The proposed algorithm presents outstanding confidence calibration performance and improved classification accuracy with two popular stochastic regularization techniques—stochastic depth and dropout—in multiple models and datasets; it alleviates overconfidence issue in deep neural networks significantly by training the networks to achieve prediction accuracy proportional to confidence of the prediction. To show generality of our approach, we also discuss how to interpret existing uncertainty estimation techniques in a principled way within our framework.

1 INTRODUCTION

Deep neural networks have achieved remarkable performance in various tasks, but have critical limitations in reliability of their predictions. One example is that inference results are often overly confident even for unseen or tricky examples; the maximum scores of individual predictions are very high even for out-of-distribution examples and consequently distort interpretation about the predictions. Since many practical applications including autonomous driving, medical diagnosis, and machine inspection require accurate uncertainty estimation as well as high prediction accuracy for each inference, such an overconfidence issue makes deep neural networks inappropriate to be deployed for real-world problems in spite of their impressive accuracy.

Regularization is a common technique in training deep neural networks to avoid overfitting problems and improve generalization accuracy Srivastava et al. (2014); Huang et al. (2016); Ioffe & Szegedy (2015). However, their objectives are not directly related to generating score distributions aligned with uncertainty of individual predictions. In other words, existing deep neural networks are inherently poor at calibrating prediction accuracy and confidence.

Our goal is to learn deep neural networks that are able to estimate accuracy and uncertainty of each prediction at the same time. Hence, we propose a generic framework to calibrate prediction score (confidence) with accuracy in deep neural networks. Our algorithm starts with an observation that variance of prediction scores measured from multiple stochastic inferences is highly correlated with accuracy and confidence of the prediction based on the average score, where we employ stochastic regularization techniques such as stochastic depth or dropout to obtain multiple stochastic inference results. We also interpret stochastic regularization as a Bayesian model, which shows relation between stochastic modeling and stochastic inferences of deep neural networks. By exploiting these properties, we design a loss function to enable deep neural network to predict confidence-calibrated scores based only on a single prediction, without stochastic inferences. Our contribution is summarized below:

- We provide a generic framework to estimate uncertainty of a prediction based on stochastic inferences in deep neural networks, which is motivated by empirical observation and theoretical analysis.
- We design a variance-weighted confidence-integrated loss function in a principled way without hyper-parameters, which enables deep neural networks to produce confidence-calibrated predictions even without stochastic inferences.
- The proposed framework presents outstanding performance to reduce overconfidence issue and estimate accurate uncertainty in various architectures and datasets.

The rest of the paper is organized as follows. We first discuss prior research related to our algorithm, and describe theoretical background for Bayesian interpretation of our approach in Section 2 and 3, respectively. Section 4 presents our confidence calibration algorithm through stochastic inferences, and Section 5 illustrates experimental results.

2 Related Work

Uncertainty estimation is a critical problem in deep neural networks and receives growing attention from machine learning community. Bayesian approach is a common tool to provide a mathematical framework for uncertainty estimation in deep neural networks. However, exact Bayesian inference is not tractable in deep neural networks due to its high computational cost, and various approximate inference techniques—MCMC Neal (1996), Laplace approximation MacKay (1992) and variational inference Barber & Bishop (1998); Graves (2011); Hoffman et al. (2013)–have been proposed. Recently, Bayesian interpretation of multiplicative noise is employed to estimate uncertainty in deep neural networks Gal & Ghahramani (2016); McClure & Kriegeskorte (2016). There are several approaches outside Bayesian modeling, which include post-processing Niculescu-Mizil & Caruana (2005); Platt (2000); Zadrozny & Elkan (2001); Guo et al. (2017) and deep ensembles Lakshminarayanan et al. (2017). All the post-processing methods require a hold-out validation set to adjust prediction scores after training, and the ensemble-based technique employs multiple models to estimate uncertainty.

Stochastic regularization is a common technique to improve generalization performance by injecting random noise to deep neural networks. The most notable method is Srivastava et al. (2014), which randomly drops their hidden units by multiplying Bernoulli random noise. There exist several variants, for example, dropping weights Wan et al. (2013) or skipping layers Huang et al. (2016). Most stochastic regularization methods exploit stochastic inferences during training, but perform deterministic inferences using the whole network during testing. On the contrary, we also use stochastic inferences to obtain diverse and reliable outputs during testing.

Although the following works do not address uncertainty estimation, their main idea is relevant to our objective. Label smoothing Szegedy et al. (2016) encourages models to be less confident, by preventing a network from assigning the full probability to a single class. The same loss function is discussed to train confidence-calibrated classifiers in Lee et al. (2018), but it focuses on how to discriminate in-distribution and out-of-distribution examples, rather than estimating uncertainty or alleviating miscalibration of in-distribution examples. On the other hand, Pereyra et al. (2017) claims that blind label smoothing and penalizing entropy enhances accuracy by integrating loss functions with the same concept with Szegedy et al. (2016); Lee et al. (2018), but improvement is marginal in practice.

3 Bayesian Interpretation of Stochastic Regularization

This section describes Bayesian interpretation of stochastic regularization in deep neural networks, and discusses relation between stochastic regularization and uncertainty modeling.

3.1 Stochastic Methods for Regularizations

Deep neural networks are prone to overfit due to their large number of parameters, and various regularization techniques including weight decay, dropout Srivastava et al. (2014), and batch normalization Ioffe & Szegedy (2015) have been employed to alleviate the issue. One popular class

of regularization techniques is stochastic regularization, which introduces random noise to a network for perturbing its inputs or weights. We focus on the multiplicative binary noise injection, where random binary noise is applied to the inputs or weights by elementwise multiplication since such stochastic regularization techniques are widely used Srivastava et al. (2014); Wan et al. (2013); Huang et al. (2016). Note that input perturbation can be reformulated as weight perturbation. For example, dropout—binary noise injection to activations—is intertpretable as weight perturbation that masks out all the weights associated with the dropped inputs. Therefore, if a classification network modeling $p(y|x, \theta)$ with parameters θ is trained with stochastic regularization methods by minimizing the cross entropy loss, its objective can be defined by

$$\mathcal{L}_{\mathrm{SR}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log p\left(y_i|x_i, \hat{\omega}_i\right) \tag{1}$$

where $\hat{\omega}_i = \theta \odot \epsilon_i$ is a set of perturbed parameters by elementwise multiplication with random noise sample $\epsilon_i \sim p(\epsilon)$, and $(x_i, y_i) \in \mathcal{D}$ is a pair of input and output in training dataset $\mathcal{D}$. Note that $\hat{\omega}_i$ is a random sample from $p(\omega)$ given by the product of the deterministic parameter θ and a random noise ϵ_i.

At inference time, the network is parameterized by the expectation of the perturbed parameters, $\Theta = \mathbb{E}[\omega] = \theta \odot \mathbb{E}[\epsilon]$, to predict an output $\hat{y}$, *i.e.*,

$$\hat{y} = \arg\max_{y} p\left(y|x, \Theta\right). \tag{2}$$

3.2 Bayesian Modeling

Given the dataset $\mathcal{D}$ with N examples, Bayesian objective is to estimate the posterior distribution of the model parameter, denoted by $p(\omega|\mathcal{D})$, to predict a label y for an input x, which is given by

$$p(y|x, \mathcal{D}) = \int_{\omega} p(y|x, \omega)p(\omega|D)d\omega. \tag{3}$$

A common technique for the posterior estimation is variational approximation, which introduces an approximate distribution $q_\theta(\omega)$ and minimizes Kullback-Leibler (KL) divergence with the true posterior $D_{\mathrm{KL}}(q_\theta(\omega)||p(\omega|D))$ as follows:

$$\mathcal{L}_{\mathrm{VA}}(\theta) = -\sum_{i=1}^{N}\int_{\omega} q_\theta(\omega) \log p(y_i|x_i, \omega)d\omega + D_{\mathrm{KL}}(q_\theta(\omega)||p(\omega)). \tag{4}$$

The intractable integral and summation over the entire dataset in Equation 4 is approximated by Monte Carlo method and mini-batch optimization resulting in

$$\hat{\mathcal{L}}_{\mathrm{VA}}(\theta) = -\frac{N}{MS}\sum_{i=1}^{M}\sum_{j=1}^{S} \log p\left(y_i|x_i, \hat{\omega}_{i,j}\right) + D_{\mathrm{KL}}\left(q_\theta(\omega)||p(\omega)\right), \tag{5}$$

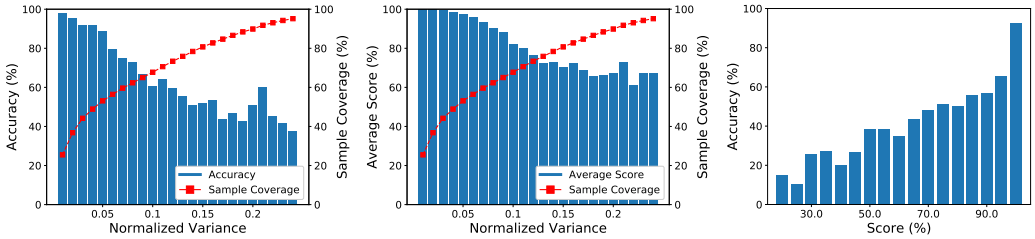
where $\hat{\omega}_{i,j} \sim q_\theta(\omega)$ is a sample from the approximate distribution, S is the number of samples, and M is the size of a mini-batch. Note that the first term is data likelihood and the second term is divergence of the approximate distribution with respect to the prior distribution.

3.3 Interpreting Stochastic Regularizations as Bayesian Model

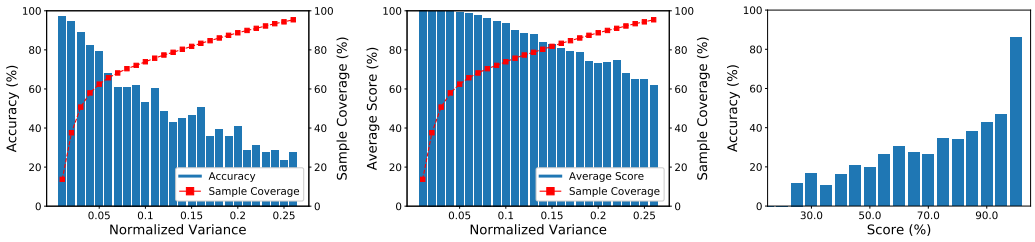
Suppose that we train a classifier with ℓ_2 regularization by a stochastic gradient descent method. Then, the loss function in Equation 1 is rewritten as

$$\hat{\mathcal{L}}_{\mathrm{SR}}(\theta) = -\frac{1}{M}\sum_{i=1}^{M} \log p\left(y_i|x_i, \hat{\omega}_i\right) + \lambda||\theta||_2^2, \tag{6}$$

where ℓ_2 regularization is applied to the deterministic parameters θ with weight λ. Optimizing this loss function is equivalent to optimizing Equation 5 if there exists a proper prior $p(\omega)$ and $q_\theta(\omega)$ is approximated as a Gaussian mixture distribution Gal & Ghahramani (2016). Note that Gal &



(a) Prediction uncertainty characteristics with stochastic depth in ResNet-34



(b) Prediction uncertainty characteristics with dropout in VGGNet with 16 layers

Figure 1: Uncertainty observed from multiple stochastic inferences with two stochastic regularization methods, (a) stochastic depth and (b) dropout. We present (left, middle) tendency of accuracy and score of the average prediction with respect to normalized variance of stochastic inferences and (right) relation between score and accuracy. In regularization methods, average accuracy and score drop gradually as normalized variance increases. The red lines indicate coverage (cumulative ratio) of examples. We present results from CIFAR-100.

Ghahramani (2016) casts dropout training as an approximate Bayesian inference. Thus, we can interpret training with stochastic depth Huang et al. (2016) within the same framework by simple modification. (See our supplementary document for detail.) Then, the predictive distribution of a model trained with stochastic regularization is approximately given by

$$\hat{p}(y|x,\mathcal{D}) = \int_{\omega} p(y|x,\omega) q_{\theta}(\omega) d\omega. \tag{7}$$

Following Gal & Ghahramani (2016) and Teye et al. (2018), we estimate the predictive mean and uncertainty by Monte Carlo approximation by drawing binary noise samples $\{\hat{\omega}_i\}_{i=1}^{T}$ as

$$\mathbb{E}_{\hat{p}}[y=c] \approx \frac{1}{T}\sum_{i=1}^{T} \hat{p}(y=c|x,\hat{\omega}_i) \quad \text{and} \quad \mathrm{Cov}_{\hat{p}}[\mathbf{y}] \approx \mathbb{E}_{\hat{p}}[\mathbf{y}^{\intercal}\mathbf{y}] - \mathbb{E}_{\hat{p}}[\mathbf{y}]^{\intercal}\mathbb{E}_{\hat{p}}[\mathbf{y}], \tag{8}$$

where $\mathbf{y} = (y_1, \ldots, y_C)^{\intercal}$ denotes a vector of C class labels. Note that the binary noise samples realize stochastic inferences such as stochastic depth and dropout by elementwise multiplication with model parameter θ. Equation 8 means that the average prediction and its variance can be computed directly from multiple stochastic inferences.

4 Confidence Calibration through Stochastic Inference

We present a novel confidence calibration technique for prediction in deep neural networks, which is given by a variance-weighted confidence-integrated loss function. We present our observation that variance of multiple stochastic inferences is closely related to accuracy and confidence of predictions, and provide an end-to-end training framework for confidence self-calibration. Then, prediction accuracy and uncertainty are directly accessible from the predicted scores obtained from a single forward pass. This section presents our observation from stochastic inferences and technical details about our confidence calibration technique.

4.1 EMPIRICAL OBSERVATIONS

Equation 8 suggests that variation of models[1] is correlated to variance of multiple stochastic predictions for a single example. In other words, by observing variation of multiple stochastic inferences, we can estimate accuracy and uncertainty of the prediction given by average of the stochastic inferences corresponding to an example.

Figure 1 presents how variance of multiple stochastic inferences given by stochastic depth or dropout is related to accuracy and confidence of the corresponding average prediction, where the confidence is measured by the maximum score of the average prediction. In the figure, accuracy and score of each bin are computed with the examples belonging to the corresponding bin of the normalized variance. We present results from CIFAR-100 with ResNet-34 and VGGNet with 16 layers. The histograms illustrate the strong correlation between the predicted variance and the reliability—accuracy and confidence—of a prediction, and between accuracy and prediction. These results suggest that one can disregard examples based on their prediction variances. Note that variance computation with more stochastic inferences provides more reliable estimation of accuracy and confidence.

4.2 CONFIDENCE-INTEGRATED LOSS

We first design a simple loss function for accuracy-score calibration by augmenting a confidence-integrated loss $\mathcal{L}_{\mathrm{U}}$ to the standard cross-entropy loss term, which is given by

$$\begin{aligned}\mathcal{L}_1(\theta) &= \mathcal{L}_{\mathrm{GT}}(\theta) + \beta\mathcal{L}_{\mathrm{U}}(\theta) \\ &= \sum_{i=1}^{N} H(p_{\mathrm{GT}}(y_i|x_i), p(y|x_i,\theta)) + \beta H(\mathcal{U}(y), p(y|x_i,\theta)) \\ &= \sum_{i=1}^{N} -\log p(y_i|x_i,\theta) + \beta D_{\mathrm{KL}}(\mathcal{U}(y)||p(y|x_i,\theta)) + \xi. \end{aligned} \tag{9}$$

where H is the cross entropy loss function, p_{GT} is the ground-truth distribution, $p(y|x_i,\theta)$ is the predicted distribution with model parameter θ, $\mathcal{U}(y)$ is the uniform distribution, and ξ is a constant. The loss denoted by $\mathcal{L}_1(\cdot)$ is determined based on cross-entropy with the ground-truths and KL-divergence from the uniform distribution. The main idea of this loss function is to regularize with the uniform distribution by expecting the score distributions of uncertain examples to be flattened first while the distributions of confident ones remain intact, where the impact of the confidence-integrated loss term is controlled by a global hyper-parameter β.

The proposed loss function is also employed in Pereyra et al. (2017) to regularize deep neural networks and improve classification accuracy. However, Pereyra et al. (2017) does not discuss confidence calibration issues. On the other hand, Lee et al. (2018) discusses the same loss function but focuses on differentiating between in-distribution and out-of-distribution examples by measuring loss of each example based only on one of the two loss terms depending on its origin.

Contrary to these approaches, we employ the loss function in Equation 9 for estimating prediction confidence in deep neural networks. Although the proposed loss makes sense intuitively, blind selection of a constant β limits its generality. Hence, we propose a more sophisticated confidence loss term by leveraging variance of multiple stochastic inferences.

4.3 VARIANCE-WEIGHTED CONFIDENCE-INTEGRATED LOSS

The strong correlation of accuracy and confidence with predicted variance observed in Figure 1 shows great potential to make confidence-calibrated prediction by stochastic inferences. However, variance computation involves multiple stochastic inferences by executing multiple forward passes. Note that this property incurs additional computational cost and may produce inconsistent results.

To overcome these limitations, we propose a generic framework for training accuracy-score calibration networks whose prediction score from a single forward pass directly provides confidence of the prediction. In this framework, we combine two complementary loss terms as in Equation 9, but they

[1]Deep neural networks with model variations can be realized by applying stochastic depth or dropout.

are balanced by the variance measured by multiple stochastic inferences. Our variance-weighted confidence-integrated loss $\mathcal{L}(\cdot)$ for the whole training data $(x_i, y_i) \in \mathcal{D}$ is defined by a linear interpolation of the standard cross-entropy loss with ground-truth $\mathcal{L}_{\mathrm{GT}}(\cdot)$ and the cross-entropy with the uniform distribution $\mathcal{L}_{\mathrm{U}}(\cdot)$, which is formally given by

$$\begin{aligned}\mathcal{L}(\theta) &= \sum_{i=1}^{N}(1-\alpha_i)\mathcal{L}_{\mathrm{GT}}^{(i)}(\theta) + \alpha_i\mathcal{L}_{\mathrm{U}}^{(i)}(\theta) \\ &= \frac{1}{T}\sum_{i=1}^{N}\sum_{j=1}^{T} -(1-\alpha_i)\log p(y_i|x_i,\hat{\omega}_{i,j}) + \alpha_i D_{\mathrm{KL}}(\mathcal{U}(y)||p(y|x_i,\hat{\omega}_{i,j})) + \xi_i \end{aligned} \tag{10}$$

where $\alpha_i \in [0,1]$ is a normalized variance, $\hat{\omega}_{i,j}(= \theta \odot \epsilon_{i,j})$ is a sampled model parameter with binary noise for stochastic prediction, T is the number of stochastic inferences, and ξ_i is a constant.

The two terms in our variance-weighted confidence-integrated loss pushes the network toward opposite directions; the first term encourages the network to produce a high score for the ground truth label while the second term forces the network to predict the uniform distribution. These terms are linearly interpolated by a balancing coefficient α_i, which is the normalized variance of individual example obtained by multiple stochastic inferences. Note that the normalized variance α_i is unique for each training example and is used to measure model uncertainty. Therefore, optimizing our loss function produces gradient signals, forcing the prediction toward the uniform distribution for the examples with high uncertainty derived by high variance while intensifying prediction confidence of the examples with low variance.

After training models in our framework, prediction of each testing example is made by a single forward pass. Unlike the ordinary models, however, a prediction score of our model is well-calibrated and represents confidence of the prediction, which means that we can rely more on the predictions with high scores.

4.4 Relation to Other Calibration Approaches

There are several score calibration techniques Guo et al. (2017); Zadrozny & Elkan (2002); Naeini et al. (2015); Niculescu-Mizil & Caruana (2005) by adjusting confidence scores through post-processing, among which Guo et al. (2017) proposes a method to calibrate confidence of predictions by scaling logits of a network using a global temperature τ. The scaling is performed before applying the softmax function, and τ is trained with validation dataset. As discussed in Guo et al. (2017), this simple technique is equivalent to maximize entropy of the output distribution $p(y_i|x_i)$. It is also identical to minimize KL-divergence $D_{\mathrm{KL}}(p(y_i|x_i)||\mathcal{U}(y))$ because

$$D_{\mathrm{KL}}(p(y_i|x_i)||\mathcal{U}(y)) = \sum_{c\in C} p(y_i^c|x_i)\log p(y_i^c|x_i) - p(y_i^c|x_i)\log \mathcal{U}(y^c) = -H(p(y_i|x_i)) + \xi_c \tag{11}$$

where ξ_c is a constant. We can formulate another confidence-integrated loss with the entropy as

$$\mathcal{L}_2(\theta) = \sum_{i=1}^{N} -\log p(y_i|x_i,\theta) - \gamma H(p(y_i|x_i,\theta)), \tag{12}$$

where γ is a constant. Equation 12 suggests that temperature scaling in Guo et al. (2017) is closely related to our framework.

5 Experiments

5.1 Experimental Setting

We choose two most widely adapted deep neural network architectures: ResNet and VGGNet. VGG architecture follows Simonyan & Zisserman (2015), where we employ dropout Srivastava et al. (2014) before every `fc` layer except for the classification layer. In ResNet, instead of stacking `conv` layers directly, outputs of residual blocks are added to the input feature representation by residual connections as proposed in He et al. (2016). Stochastic depth Huang et al. (2016) is used

Table 1: Classification accuracy and expected calibration error (ECE) of two models with Tiny ImageNet and CIFAR-100 datasets. We present results from baseline algorithm, CI and VWCI (ours) to understand their characteristics.

	ResNet-34				VGG-16			
	Tiny ImageNet		CIFAR-100		Tiny ImageNet		CIFAR-100	
	Acc. [%]	ECE	Acc. [%]	ECE	Acc. [%]	ECE	Acc. [%]	ECE
Baseline[det]	50.82	0.067	77.19	0.109	46.58	0.346	73.78	0.187
CI[$\beta = 0.01$]	49.16	0.119	77.53	0.074	47.11	0.259	73.78	0.163
CI[$\beta = 0.1$]	51.45	0.035	77.23	0.085	46.94	0.122	73.68	**0.083**
CI[$\beta = 1.0$]	50.77	0.255	77.48	0.295	45.40	0.130	73.62	0.291
VWCI	**52.80**	**0.027**	**77.74**	**0.038**	**48.03**	**0.053**	**73.87**	0.098

for stochastic regularization in ResNet. Note that, as discussed in Section 3.3, both dropout and stochastic depth inject multiplicative binary noise to within-layer activations or residual blocks, they are equivalent to noise injection into network weights. Hence, training with ℓ_2 regularization term enables us to interpret stochastic depth and dropout by Bayesian models.

We evaluate the proposed framework on two benchmarks, Tiny ImageNet and CIFAR-100. Tiny ImageNet contains 64×64 images with 200 object labels whereas CIFAR-100 has 32×32 images of 100 objects. There are 500 training images per class in both datasets. For testing, we use the validation set of Tiny ImageNet and the test set of CIFAR-100, which contain 50 and 100 images per class, respectively. To use the same network for two benchmarks, we resize images in Tiny ImageNet into 32×32.

All networks are trained with stochastic gradient decent with the momentum of 0.9 for 300 epochs. We set the initial learning rate to 0.1 and exponentially decay it with factor of 0.2 at epoch 60, 120, 160, 200 and 250. Each batch consists of 64 and 256 training examples for ResNet and VGG architectures, respectively. To train networks with the proposed variance-weighted confidence-integrated loss, we draw T samples for each input image by default, and compute the normalized variance α by running T forward passes. The number of samples T is set to 5. The normalized variance is estimated based on the variance of Bhattacharyya coefficients between individual predictions and the average prediction. The trained models with the variance-weighted confidence-integrated (VWCI) loss are compared to the models with confidence-integrated (CI) losses for several different constant β's.

5.2 Evaluation Metric

We measure classification accuracy and expected calibration error (ECE) of the trained models. While classification accuracy shows regularization effect of the confidence-integrated loss term, ECE summarizes miscalibration of a model by measuring discrepancy between confidence and accuracy. Specifically, let B_m be a set of indices of test examples whose scores for the ground-truth labels fall into the score interval $(\frac{m-1}{M}, \frac{m}{M}]$, where M is the number of bins. Then, ECE is formally defined by

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N'} \left|\text{acc}(B_m) - \text{conf}(B_m)\right| \tag{13}$$

where N' is the number of the test samples. Also, accuracy and confidence of each bin are given by

$$\text{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}(\hat{y}_i = y_i) \qquad \text{and} \qquad \text{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} p_i \tag{14}$$

where $\hat{y}_i$ and y_i are predicted and true label of the i-th example and p_i is its predicted confidence.

5.3 Results

Table 1 presents results of ResNet-34 and VGG-16 on both datasets. We observe that baseline methods with stochastic inferences reduce calibration error and the reduction becomes more significant in proportion to number of inferences. These results imply benefit of stochastic inference for confidence calibration, and reflect performance of methods by multiplicative noise in Gal & Ghahramani (2016); McClure & Kriegeskorte (2016).

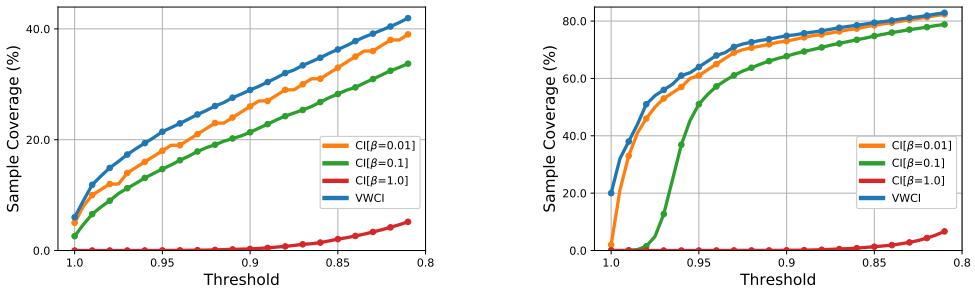


Figure 2: Coverage of ResNet-34 models with confidence interval on Tiny ImageNet (left) and CIFAR-100 (right). Coverage is computed by the portion of examples with higher accuracy and confidence than threshold, which is shown in x-axis.

The models trained with VWCI loss consistently outperform baselines and is competitive to models with CI loss on both classification accuracy and confidence calibration. Stochastic inference and variance-driven weight allow us to measure uncertainty for each instance, and enable two oppositional terms to be well balanced by the measured uncertainty. The confidence loss term regularizes the network by forcing predictions to uniform distribution, and the proper estimation of its coefficient leads to accuracy gain and confidence calibration. Note that, by assigning a low coefficient to the loss term for a confident example, the network allows the example to remain confident whereas a high weight for an uncertain example reduces confidence of the prediction.

The CI loss has a confidence loss term with fixed coefficient β. The networks trained with proper β show impressive improvement on both criteria, but their performance is sensitive to choice of β as this strategy ignores predictive uncertainty for confidence loss; an inappropriate choice of β even worsens accuracy and calibration error, *e.g.*, ResNet-34 trained with CI[$\beta = 0.01$] on Tiny ImageNet. Also, there seems to be no single β that is globally optimal across architectures and benchmark datasets. For instance, training the network with CI[$\beta = 0.01$] on Tiny ImageNet gives the worst accuracy with ResNet-34 and the best accuracy with VGG-16. In the experiments, the CI loss often works well on CIFAR-100 due to high accuracy. The majority of examples are classified correctly and the overconfident property of deep neural networks do little harm for confidence calibration. Specifically, CI loss sometimes achieves slightly better performance than VWCI with a certain fixed coefficient β because the measured normalized variance by stochastic inferences and its range are small. On the other hand, in Tiny ImageNet dataset, performance of VMCI is consistently better than CI because Tiny ImageNet is substantially more challenging than CIFAR-100.

A critical benefit of our variance-driven weight in the VWCI loss is the capability to maintain examples with high accuracy and high confidence. This is an important property for building real-world decision making systems with confidence interval, where the decisions should be both highly accurate and confident. Figure 2 illustrates coverage of test examples varying the confidence threshold, and VWCI shows better coverage than CI because CI pushes all instances to uniform with the same strength β regardless of their uncertainty unlike VWCI. It is also notable that β for the best coverage is different from that for the best accuracy and ECE whereas VWCI balances these based on the predictive uncertainty. These results suggest that using the predictive uncertainty for balancing the terms is preferable over setting a constant coefficient in our loss function. More experimental results are presented in the supplementary document.

6 Conclusion

We presented a generic framework for uncertainty estimation of a prediction in deep neural networks by calibrating accuracy and score based on stochastic inferences. Based on Bayesian interpretation of stochastic regularization and our empirical observation results, we claim that variation of multiple stochastic inferences for a single example is a crucial factor to estimate uncertainty of the average prediction. Motivated by this fact, we design the variance-weighted confidence-integrated loss to learn confidence-calibrated networks and enable uncertainty to be estimated by a single prediction. The proposed algorithm is also useful to understand existing confidence calibration methods in a

unified way, and we compared our algorithm with other variations within our framework to analyze their characteristics.

REFERENCES

D. Barber and Christopher Bishop. Ensemble learning for multi-layer networks. In *NIPS*, 1998.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016.

Alex Graves. Practical variational inference for neural networks. In *NIPS*, 2011.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *J. Mach. Learn. Res.*, 14(1):1303–1347, May 2013. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=2502581.2502622.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS*, pp. 6405–6416, 2017.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *ICLR*, 2018.

David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Comput.*, 4(3):448–472, May 1992. ISSN 0899-7667. doi: 10.1162/neco.1992.4.3.448. URL http://dx.doi.org/10.1162/neco.1992.4.3.448.

Patrick McClure and Nikolaus Kriegeskorte. Representation of uncertainty in deep neural networks through sampling. *CoRR*, abs/1611.01639, 2016.

Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *AAAI*, 2015.

Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, 1996. ISBN 0387947248.

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *ICML*, 2005.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. 10, 06 2000.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CVPR*, 2016.

Mattias Teye, Hossein Azizpour, and Kevin Smith. Bayesian uncertainty estimation for batch normalized deep networks. *arXiv preprint arXiv:1802.06455*, 2018.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *ICML*, 2013.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML*, 2001.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *KDD*, 2002.