

---

# Deep Probabilistic Ensembles: Approximate Variational Inference through KL Regularization

---

**Kashyap Chitta**  
Carnegie Mellon University  
kchitta@andrew.cmu.edu

**Jose M. Alvarez**  
NVIDIA  
josea@nvidia.com

**Adam Lesnikowski**  
NVIDIA  
alesnikowski@nvidia.com

## Abstract

In this paper, we introduce Deep Probabilistic Ensembles (DPEs), a scalable technique that uses a regularized ensemble to approximate a deep Bayesian Neural Network (BNN). We do so by incorporating a KL divergence penalty term into the training objective of an ensemble, derived from the evidence lower bound used in variational inference. We evaluate the uncertainty estimates obtained from our models for active learning on visual classification. Our approach steadily improves upon active learning baselines as the annotation budget is increased.

## 1 Introduction

Modeling uncertainty for deep neural networks has a wide range of potential applications: it can provide important information about the reliability of predictions, or better strategies for labeling data in order to improve performance. Bayesian methods, which provide an approach to do this, have recently gained momentum, and are beginning to find more widespread use in practice [6, 3, 5, 10].

The formulation of a Bayesian Neural Network (BNN) involves placing a prior distribution over all the parameters of a network, and obtaining the posterior given the observed data [13]. The distribution of predictions provided by a trained BNN helps capture the model’s uncertainty. However, training a BNN involves marginalization over all possible assignments of weights, which is intractable for deep BNNs without approximations [6, 3, 5]. Existing approximation algorithms limit their applicability, since they do not specifically address the fact that deep BNNs on large datasets are more difficult to optimize than deterministic networks, and require extensive parameter tuning to provide good performance and uncertainty estimates [14]. Furthermore, estimating uncertainty in BNNs requires drawing a large number of samples at test time, which can be extremely computationally demanding.

In practice, a common approach to estimate uncertainty is based on ensembles [12, 1]. Different models in a trained ensemble of networks are treated as if they were samples drawn directly from a BNN posterior. Ensembles are easy to optimize and fast to execute. However, they do not approximate uncertainty in the same manner as a BNN. For example, the parameters in the first kernel of the first layer of a convolutional neural network may serve a completely different purpose in different members of an ensemble. Therefore, the variance of the values of these parameters after training cannot be compared to the variance that would have been obtained in the first kernel of the first layer of a trained BNN with the same architecture.

In this paper, we propose Deep Probabilistic Ensembles (DPEs), a novel approach to approximate BNNs using ensembles. Specifically, we use variational inference [2], a popular technique for training BNNs, to derive a KL divergence regularization term for ensembles. DPEs are parallelizable, easy to implement, yield high performance, and in our experiments, provide better uncertainty estimates than existing methods when used for active learning on visual classification benchmarks.

## 2 Deep Probabilistic Ensembles

For inference in a BNN, we can consider the weights  $w$  to be latent variables with some prior distribution,  $p(w)$ . These weights relate to the observed dataset  $x$  through the likelihood,  $p(x|w)$ . We aim to compute the posterior  $p(w|x)$  that best explains the observed data. Variational inference involves restricting ourselves to a family of distributions  $D$  over the latent variables, and optimizing for the member of this family  $q^*(w)$  that is closest to the true posterior in terms of KL divergence,

$$q^*(w) = \arg \min_{q(w) \in D} KL(q(w)||p(w|x)). \quad (1)$$

Simplifying this minimization objective, we obtain the negative Evidence Lower Bound (ELBO) [2],

$$-ELBO = KL(q(w)||p(w)) - \mathbb{E}[\log p(x|w)], \quad (2)$$

where the first term is the KL divergence between  $q(w)$  and the chosen prior distribution  $p(w)$ ; and the second term is the expected negative log likelihood (NLL) of the data  $x$  based on the current parameters  $w$ . The optimization difficulty in variational inference arises partly due to this expectation of the NLL. In deterministic networks, fragile co-adaptations exist between different parameters, which can be crucial to their performance [17]. Features typically interact with each other in a complex way, such that the optimal setting for certain parameters is highly dependent on specific configurations of the other parameters in the network. Co-adaptation makes training easier, but can reduce the ability of deterministic networks to generalize [9]. Popular regularization techniques to improve generalization, such as Dropout, can be seen as a trade-off between co-adaptation and training difficulty [16]. An ensemble of deterministic networks exploits co-adaptation, as each network in the ensemble is optimized independently, making them easy to optimize.

For BNNs, the nature of the objective, an expectation over all possible assignments of  $q(w)$ , prevents the BNN from exploiting co-adaptations during training, since we seek to minimize the NLL for *any generic deterministic network* sampled from the BNN. While in theory, this should lead to great generalization, it becomes very difficult to tune BNNs to produce competitive results. In this paper, we propose a form of regularization to use the optimization simplicity of ensembles for training BNNs.

**KL regularization.** The standard approach to training neural networks involves regularizing each individual parameter with  $L_1$  or  $L_2$  penalty terms. We instead apply the KL divergence term in Eq. 2 as a regularization penalty  $\Omega$ , to the *set of values* of a given parameter takes over *all members in an ensemble*. If we choose the family of Gaussian functions for  $p(w)$  and  $q(w)$ , this term can be analytically computed by assuming mutual independence between the network parameters and factoring the term into individual Gaussians. The KL divergence between two Gaussians with means  $\mu_q$  and  $\mu_p$ , standard deviations  $\sigma_q$  and  $\sigma_p$  is given by

$$KL(q||p) = \frac{1}{2} \left( \log \frac{\sigma_q^2}{\sigma_p^2} + \frac{\sigma_p^2 + (\mu_q - \mu_p)^2}{\sigma_q^2} - 1 \right). \quad (3)$$

**Choice of prior.** In our experiments, we choose the network initialization technique proposed by He et al. [7] as a prior. For batch normalization parameters, we fix  $\sigma_p^2 = 0.01$ , and set  $\mu_p = 1$  for the weights and  $\mu_p = 0$  for the biases. For the weights in convolutional layers with the ReLU activation (with  $n_i$  input channels,  $n_o$  output channels, and kernel dimensions  $w \times h$ ) we set  $\mu_p = 0$  and  $\sigma_p^2 = \frac{2}{n_o w h}$ . Linear layers can be considered a special case of convolutions, where the kernel has the same dimensions as the input activation. The KL regularization of a layer  $\Omega^l$  is obtained by removing the terms independent of  $q$  and substituting for  $\mu_p$  and  $\sigma_p$  in Eq. 3,

$$\Omega^l = \sum_{i=1}^{n_i n_o w h} \left( \log \sigma_i^2 + \frac{2}{n_o w h \sigma_i^2} + \frac{\mu_i^2}{\sigma_i^2} \right), \quad (4)$$

where  $\mu_i$  and  $\sigma_i$  are the mean and standard deviation of the set of values taken by a parameter across different ensemble members. In Eq. 4, the first term prevents extremely large variances compared to the prior, so the ensemble members do not diverge completely from each other. The second term heavily penalizes variances less than the prior, promoting diversity between members. The third term closely resembles weight decay, keeping the mean of the weights close that of the prior, especially when their variance is also low.

Table 1: Validation accuracies (in %) on an active learning task. DPEs give consistent improvements in results over both random sampling and standard ensembles on both datasets. Relative performance to the upper bound performance for that data sampling strategy is given in (brackets).

| Task      | Data Sampling | Accuracy @8%         | Accuracy @16%        | Accuracy @32%        |
|-----------|---------------|----------------------|----------------------|----------------------|
| CIFAR-10  | Random        | 80.60 (84.66)        | 86.80 (91.18)        | 91.08 (95.67)        |
|           | Ensemble      | 82.41 (86.56)        | 90.05 (94.59)        | 94.13 (98.87)        |
|           | DPE (Ours)    | <b>82.88 (87.06)</b> | <b>90.15 (94.70)</b> | <b>94.33 (99.09)</b> |
| CIFAR-100 | Random        | 39.57 (50.18)        | 54.92 (69.64)        | 66.65 (84.51)        |
|           | Ensemble      | 40.49 (51.34)        | 56.89 (72.14)        | 69.68 (88.36)        |
|           | DPE (Ours)    | <b>40.87 (51.83)</b> | <b>56.94 (72.20)</b> | <b>70.12 (88.92)</b> |

**Objective function.** We can now rewrite the minimization objective from Eq. 2 for an ensemble as:

$$\Theta^* = \arg \min_{\Theta} \sum_{i=1}^N \sum_{e=1}^E \mathcal{H}(y^i, \mathcal{M}_e(\mathbf{x}^i, \Theta_e)) + \beta \Omega(\Theta), \quad (5)$$

where  $\{(\mathbf{x}^i, y^i)\}_{i=1}^N$  is the training data,  $E$  is the number of models in the ensemble, and  $\Theta_e$  refers to the parameters of the model  $\mathcal{M}_e$ .  $\mathcal{H}$  is the cross-entropy loss for classification and  $\Omega$  is our KL regularization penalty over all the parameters of the ensemble  $\Theta$ . We obtain  $\Omega$  by aggregating the penalty from Eq. 4 over all the layers in a network, and use a scaling term  $\beta$  to balance the regularizer with the likelihood loss. By summing the loss over each independent model, we are approximating the expectation of the ELBO’s NLL term in Eq. 2 over only the current ensemble configuration, a subset of all possible assignments of  $q(w)$ . This is the main distinction between our approach and traditional variational inference.

### 3 Experiments

Active learning allows a model to choose the data from which it learns, by iteratively using the partially trained model to decide which examples to annotate from a large unlabeled dataset [4]. In our approach, the data with the highest entropy for the average prediction across the ensemble members is added to the training set from the unlabeled pool. We experiment with active learning on the CIFAR dataset, which has two object classification tasks over natural images: one coarse-grained over 10 classes and one fine-grained over 100 classes [11]. There are 50k training images and 10k validation images of resolution  $32 \times 32$ .

We use eight models in our ensembles, each a pre-activation ResNet-18 [8]. Our first experiment involves initializing models by training with a random subset of 4% of the dataset, and then re-training the model at three additional intervals after adding more data (at 8%, 16% and 32% of the data). We evaluate three approaches for adding data to the training set, (1) Random: pick the required percentage of data through random sampling; (2) Ensemble: pick the samples with highest average entropy across our eight models with standard  $L_2$  regularization for each model; and (3) Deep Probabilistic Ensemble (DPE): pick samples with the highest average entropy across our eight models jointly regularized with our KL regularization. Our results are shown in Table 1. Our figures correspond to the mean accuracy of 3 experimental trials. As shown, active learning with DPEs clearly outperforms random baselines, and consistently provides better results compared to ensemble based active learning methods. Further, though the gap is small, it typically grows as the annotation budget (size of the labeled dataset) increases, which is a particularly appealing property for large-scale labeling projects.

In our second experiment, we compare our approach to state-of-the-art results on the CIFAR-10 dataset using 20% of the labeled training data. We include results for (1) the proposed DPE; (2) a deterministic network using the same architecture and  $L_2$  regularization; (3) Core-set selection with a single VGG-16 [15] and; (4) an ensemble of DenseNet-121 models [1]. We also include the upper bound of each experiment. That is, the accuracy obtained when the model is trained using all the data available. For the first two methods, we start training at 4% of the data and then re-train the model at four additional intervals (at 8%, 12%, 16% and 20%). For the others we use the performance reported in the corresponding papers. As shown in Table 2, our approach outperforms all the others, not only achieving higher accuracy with limited training data, but also reducing the gap to the corresponding upper bound.

Table 2: **CIFAR-10**: Comparison of our proposed approach to active learning baselines for learning with limited labels. DPEs show state-of-the-art performance. Relative performance refers to the performance for each method using a limited labeling budget compared to the performance using the whole dataset.

| Method                | Accuracy @20% | Accuracy @100% | Relative Performance |
|-----------------------|---------------|----------------|----------------------|
| Core-set [15]         | 74%           | 90%            | 82.2%                |
| Ensemble [1]          | 85%           | <b>95.5%</b>   | 89.0%                |
| Deterministic Network | 87.5%         | 94.4%          | 92.7%                |
| DPE (Ours)            | <b>92%</b>    | 95.2%          | <b>96.3%</b>         |

## 4 Conclusion

In this paper, we introduced Deep Probabilistic Ensembles (DPEs) for uncertainty estimation in deep neural networks. The key idea is to train ensembles with a novel KL regularization term as a means to approximate variational inference for BNNs. Our results demonstrate that DPEs improve performance on active learning tasks over baselines and state-of-the-art active learning techniques on two image classification datasets. Importantly, contrary to traditional Bayesian methods, our approach is simple to integrate in existing frameworks and scales to large models and datasets without impacting performance. We look forward to future work with these models for any downstream tasks requiring precise uncertainty information.

## References

- [1] William H. Beluch, Tim Genewein, Andreas Nürnberger, and Jan M. Köhler. The power of ensembles for active learning in image classification. In *CVPR*, 2018.
- [2] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational Inference: A Review for Statisticians. *ArXiv e-prints*, 2016.
- [3] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *ICML*, 2015.
- [4] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 1994.
- [5] Y. Gal and Z. Ghahramani. Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference. *ArXiv e-prints*, 2015.
- [6] Alex Graves. Practical variational inference for neural networks. In *NIPS*. 2011.
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [9] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *ArXiv e-prints*, 2012.
- [10] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS*, 2017.
- [11] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.
- [12] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS*, 2017.
- [13] Radford M. Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.
- [14] Ian Osband. Risk versus uncertainty in deep learning: Bayes, bootstrap and the dangers of dropout. In *NIPS Workshops*, 2016.
- [15] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *ICLR*, 2018.
- [16] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *ICML*, 2014.
- [17] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014.