BATCH-ENSEMBLE STOCHASTIC NEURAL NETWORKS
FOR OUT-OF-DISTRIBUTION DETECTION

Xiongjie Chen*, Yunpeng Li†, and Yongxin Yang‡

*University of Surrey, UK
{∗xiongjie.chen, yunpeng.li}@surrey.ac.uk
†Queen Mary University of London, UK
¶yongxin.yang@qmul.ac.uk

ABSTRACT
Out-of-distribution (OOD) detection has recently received much attention from the machine learning community because it is important for deploying machine learning models in real-world applications. In this paper we propose an uncertainty quantification approach by modeling data distributions in feature spaces. We further incorporate an efficient ensemble mechanism, namely batch-ensemble, to construct the batch-ensemble stochastic neural networks (BE-SNNs) and overcome the feature collapse problem. We compare the performance of the proposed BE-SNNs with the other state-of-the-art approaches and show that BE-SNNs yield superior performance on several OOD detection benchmarks, such as the Two-Moons dataset, the FashionMNIST vs MNIST dataset, FashionMNIST vs NotMNIST dataset, and the CIFAR10 vs SVHN dataset.

Index Terms — Out-of-distribution detection, ensemble methods, stochastic neural networks.

1. INTRODUCTION
Out-of-distribution (OOD) detection is a critical and challenging task for applying deep learning models in real-world scenarios. Various OOD detection methods have been proposed to enhance the reliability of deep learning models [1, 2, 3, 4, 5, 6]. One main branch of current OOD detection approaches is developed based on uncertainty quantification techniques. Existing uncertainty quantification techniques can be roughly categorized into two classes. The first class relies on ensembles of deep neural networks [7, 8, 9, 10], where the outputs of multiple individually trained models are combined to estimate the uncertainty. The second family of uncertainty estimation approaches aims to measure the predictive uncertainty using deterministic single forward pass neural networks [11, 12], where the uncertainty is estimated by modeling the distribution of data features.

Despite their success on benchmark datasets, both deep ensemble methods and deterministic single forward pass neural networks have their respective limitations. An obvious disadvantage of deep ensemble methods is their computational costs [8, 13, 14]. Particularly, ensemble methods are limited in practice since each ensemble member requires an independent copy of neural network weights and they need to be trained separately. Therefore, their computational and memory costs increase linearly with the ensemble size in both training and testing [8]. In contrast, single forward-pass methods have shown to be efficient in modeling uncertainties, but they suffered from the so-called feature collapse problem, which refers to the phenomenon that features of out-of-distribution samples are mapped onto the region of in-distribution sample features [11].

To take the best from both worlds, we propose the batch-ensemble stochastic neural networks (BE-SNNs), an OOD detection method consisting of a novel single forward-pass approach and an efficient ensemble method inspired by [8]. Our main contributions are: (i) The proposed BE-SNNs can overcome the undesirable feature collapse in single forward-pass methods while also maintaining their low computational cost. We achieve this by constructing an ensemble of single forward pass models with the efficient batch-ensemble mechanism [8]. (ii) A novel single forward pass model is proposed in this paper. The proposed single forward pass model captures class distributions in a flexible way by approximating them with sets of feature vectors. The constructed empirical distributions are inherently multi-modal and can better model data with complex structures than simple distributions such as Gaussian distributions with diagonal covariance matrices. (iii) We introduce a novel OOD score that can better distinguish between OOD samples and in-distribution samples. The proposed OOD score is computed based on an adaptively tempered version of softmax outputs, which can be interpreted as applying data-adaptive temperature to softmax activation functions. (iv) We demonstrate the effectiveness of the BE-SNNs on several OOD detection benchmarks, including the Two-Moons dataset, the FashionMNIST vs MNIST dataset, FashionMNIST vs NotMNIST, and the CIFAR10 vs SVHN dataset [15, 16, 17, 18].

2. PRELIMINARY
Batch-ensemble (BE) is a parameter efficient variant of deep ensemble that construct ensembles over a rank-1 subspace of networks’ weights, which only brings little extra computational overhead compared to storing full weight matrices [8]. Denote by $W \in \mathbb{R}^{m \times n}$ the weight matrix of a neural network layer, where $m$ is the input dimension and $n$ is the output dimension. Each ensemble member in the batch-ensemble is assigned with a pair of trainable vectors $s_{n_e} \in \mathbb{R}^m$ and $r_{n_e} \in \mathbb{R}^n$, where $n_e \in \{1, \cdots, N_e\}$ and $N_e$ is the ensemble size. The weights $\overline{W}_{n_e}$ of ensemble layers are then obtained by:

$$\overline{W}_{n_e} = W \circ F_{n_e},$$

(1)

where $\circ$ refers to the element-wise product, $W$ is shared across ensemble members, and $F_{n_e} = r_{n_e} s_{n_e}^\top \in \mathbb{R}^{m \times n}$ is a rank-one matrix. Given an input $x$, the output $y_{n_e}$ of the $n_e$-th ensemble

1Code to reproduce experiment results is available at: https://github.com/xiongjiechen/BE-SNNs.
where

The distribution of the
Framework of BE-SNNs. BE-SNN-1 consists of a data feature extractor
one-hot class label
Euclidean distance by the set of feature vectors
complex distributions than pre-defined simple distributions such
as well as BE-SNNs in general, is represented by a set of feature
neural networks. The distribution of class features in the BE-SNN-1,
generator
ping the input sample
Layer can be computed by:
\[ y_{nc} = \phi \left( \frac{1}{N} \sum_{i=1}^{N} x_i \right) = \phi \left( \frac{1}{N} \sum_{i=1}^{N} s_{nc}^{(i)} \right). \]
(2)
where \( \phi \) denotes the activation function of the ensemble layer.

3. BATCH-ENSEMBLE STOCHASTIC NEURAL NETWORKS

Framework of BE-SNNs. We first introduce the framework of the BE-SNN-1, i.e., an individual realization of BE-SNNs. The BE-SNN-1 consists of a data feature extractor \( \Phi_{\theta}(\cdot) \) mapping the input sample \( x \in \mathbb{R}^d \) to a feature space, and a class feature generator \( \Psi_{\theta}(\cdot) \) modeling the distribution of feature representations of each class. In this work, both \( \Phi_{\theta}(\cdot) \) and \( \Psi_{\theta}(\cdot) \) are built with neural networks. The distribution of class features in the BE-SNN-1, as well as BE-SNNs in general, is represented by a set of feature vectors for each class, which enables us to approximate more complex distributions than pre-defined simple distributions such as Gaussian distributions with diagonal covariance matrices.

The input of the class feature generator \( \Psi_{\theta} : \mathbb{R}^{d+C} \rightarrow \mathbb{R}^d \) is a one-hot class label \( h_c \in \mathbb{R}^d \) and a random vector \( \epsilon_m \in \mathbb{R}^{d+C} \), and the output of it is \( d \)-dimensional class-dependent feature vectors \( e_{c,m} = \Psi_{\theta}(h_c, \epsilon_m) \), for \( m \in \{1, \ldots, M\} \) and \( c \in \{1, \ldots, C\} \), where \( \epsilon_m \sim \mathcal{N}(\mathbf{0}, I_{d+C}) \). The ensemble size, yielding \( N_e \) different weights in \( N_e \) ensemble members as introduced in Section 2. Therefore, in BE-SNNs, the set of data feature extractors \( \{\Phi_{\theta_{n_e}}(\cdot)\}_{n_e=1}^{N_e} \) produces \( N_e \) different feature vectors for a given input \( x \), and the set of class feature generators \( \{\Psi_{\theta_{n_e}}(\cdot)\}_{n_e=1}^{N_e} \) produces \( N_e \times M \) feature vectors for each class. To make a prediction for a given input \( x \), the BE-SNNs classify \( x \) as the class with the maximum average kernel value:

\[
\argmax_c \left\{ \frac{1}{N_e} \sum_{n_e=1}^{N_e} \exp \left( - \frac{1}{M} \sum_{m=1}^{M} \left\| \Phi_{\theta_{n_e}}(x) - e_{n_e,c,m} \right\|^2 \right) \right\}. \quad (4)
\]

Loss functions of BE-SNNs. We first introduce the loss function for an individual ensemble member of BE-SNNs. The loss function of BE-SNN-1 consists of two parts, a classification loss and regularization terms. For the classification loss \( L_{\text{cl}}(y, \hat{y}) \), we use the binary cross entropy loss between the one-hot ground-truth label \( y \) and the one-hot prediction label \( \hat{y} \) as in [5]:

\[
L_{\text{cl}}(y, \hat{y}) = -\sum_{c=1}^{C} \left[ y_c \log(\hat{y}_c) + (1-y_c) \log(1-\hat{y}_c) \right],
\]

where \( y_c \) and \( \hat{y}_c \) are \( c \)-th element of the one-hot ground-truth and predictions, respectively.

To prevent class feature vectors \( e_{c,m} \) from converging to almost the same value for all \( m \in \{1, \ldots, M\} \), we add a regularization term \( R_c(\cdot) \) to the loss function. Particularly, we design the regularization term \( R_c(\cdot) \) to encourage the entropy of the \( c \)-th class to be proportional to the entropy of data features of the \( c \)-th class. Note by \( y_c(x) \) the \( c \)-th element of the one-hot label of \( x \), we approximate the entropy \( \hat{H}(\Phi_{\theta}(\cdot)|y_c(x)=1) \) of the data sample feature for all \( c \in \{1, \ldots, C\} \) using the \( k \)-nearest neighbor entropy estimator proposed in [19]. The class-dependent entropy

\[
\hat{H}(\Phi_{\theta}(\cdot)|y_c(x)=1)
\]

is then used as a threshold to penalize class features whose entropy is smaller than this threshold. The described
We minimize the average of the loss function where the adaptive tempering constant is used during training.

\[
R_c(x, e) = \max \left(0, \frac{1}{M} \sum_{m=1}^{M} \mathbb{I}(\Phi_e(x)|y_c(x) = 1) - \mathbb{H}(\epsilon_{c,m}) \right),
\]

(5)

In addition, the loss function of BE-SNNs also includes a gradient penalty term \(L_{gp}(\hat{y})\) to encourage sensitivity of the classifier as in [5]:

\[
L_{gp}(\hat{y}) = \left\| \nabla_x \sum_c \hat{y}_c \right\|_2^2 - 1, \quad \text{where } \nabla_x \sum_c \hat{y}_c \text{ is the derivative of } \sum_c \hat{y}_c \text{ w.r.t to the input } x.
\]

For the \(n_c\)-th ensemble member, the loss function of it is as follows:

\[
L_{n_c} = \sum_{c} \lambda_1 R_c(x, e) + \lambda_2 L_{gp}(\hat{y}).
\]

(6)

We minimize the average of the loss function \(L_{n_c}\) defined in Eq. (6) across all ensemble members to train BE-SNNs:

\[
L_{BE-SNN} = \frac{1}{N_c} \sum_{n_c=1}^{N_c} L_{n_c}.
\]

**OOD detection score.** Denote by \(d(x) = [d_1(x), \ldots, d_C(x)]\) the distance vector in BE-SNN-1, we compute a categorical distribution through softmax activations \(\sigma(\cdot)\) to classify a given input \(x\) with the tempered negative distance as the softmax input:

\[
\xi(x) = \frac{-d(x)}{\exp(\min_c(d_c(x))},
\]

(7)

where \(\min(d_c(x))\) is the smallest distance between class features and input features. It is worth mentioning that \(\sigma(\xi(x))\) can be seen as a tempered softmax prediction as in [20], whereas the tempering constant \(\exp(\min_c(d_c(x)))\) is data-adaptive. Note that the adaptive tempered softmax is only used in testing, but not for training BE-SNNs and making predictions for in-distribution samples. The motivation of the adaptively tempered softmax \(\exp(\min_c(d_c(x)))\) is that input features that are far from class features should be assigned with more uniform categorical distributions. A similar idea of using adaptive tempering constants was proposed in a concurrent work [21], where the adaptive tempering constant is used during training.

We then propose to use the entropy of the averaged tempered softmax prediction \(\mathbb{H}(\xi(x)) = -\sum_c \xi_c(x) \log \xi_c(x)\) as the OOD score, where \(\gamma(x) = \{\hat{\gamma}_1, \ldots, \hat{\gamma}_C\}, \hat{\gamma}_c(x) = \frac{1}{N_c} \sum_{n_c=1}^{N_c} \sigma_c(\xi_n(x))\), and \(\xi_n(x)\) is the tempered negative distance of the \(n_c\)-th ensemble member computed in Equation (7). An overview of the structure of the proposed BE-SNNs is presented in Figure 1.

### 4. EXPERIMENTS

In this section, we evaluate the performance of BE-SNNs on several OOD benchmarks adopted by previous works [5, 6]. We choose regularization coefficients \(\lambda_1\) and \(\lambda_2\) from a candidate set \{0.0, 0.1, 0.5, 1.0\}, and empirically found that \(\lambda_1 = 1.0\) and \(\lambda_2 = 0.5\) lead to the best validation performance of BE-SNNs.

#### 4.1. Two Moons dataset

Following the setup in [5], we first visualize the performance of the BE-SNN-1 on the Two-Moons dataset and compare it with deterministic uncertainty quantification (DUQ) [5] and a deep ensemble of softmax networks with an ensemble size of 4. In this experiment, while the evaluated dataset is a toy dataset, we do not focus on the classification accuracy since all evaluated models can achieve 100% accuracy, but how well these models can assign proper confidence scores to their predictions. The experiment results presented in Figure 2 indicate that, even with deep ensembles, softmax networks still produce overconfident predictions on OOD samples. One potential reason is that the Two-Moons dataset is too simple for the networks to converge to different local optima, thus all ensemble members are almost the same after training. As a comparison, both the BE-SNN-1 and the DUQ are able to assign proper confidence to their predictions. It can be observed from Figure 2 that the DUQ and the BE-SNN-1 only produce high confidence predictions on the in-distribution, and show gradually decreasing confidence in areas that are far away from the in-distribution.

**4.2. Image classification datasets**

In this section, we evaluate BE-SNNs’ ability to detect OOD samples on image OOD detection datasets. Three baselines are compared with the BE-SNNs, including a vanilla softmax neural network, the DUQ [5], and the Gaussian classifier [6]. We compute the entropy of the adaptively tempered softmax prediction as the OOD score for the proposed BE-SNNs. The performance of compared methods are evaluated using the following OOD detection metrics work [4, 3, 5, 1]: (i) FPR95, the false positive rate of classifying OOD examples when the true positive rate (recall) of in-distribution is 95%; (ii) the area under the ROC curve (AUROC); (iii) the area under the precision-recall curve (AUPRC). Lower FPR95, higher AUROC and AUPRC indicate better performance.

In the FashionMNIST vs MNIST & NotMNIST dataset experiment, we train evaluated models on the FashionMNIST dataset, and test their ability to detect OOD samples from the MNIST and NotMNIST datasets. The experiment results of BE-SNNs shown in Table 1 are achieved by using \(N_c = 4\) ensemble members.

It can be observed from Table 1 that all the evaluated methods produced similar classification accuracy, while their OOD detection metrics vary from method to method. The softmax network leads to the worst OOD detection metrics in both experiment setups. The BE-SNNs achieved better OOD detection performance in most settings than evaluated baselines regarding the reported OOD detection metrics. To draw a fair comparison, all evaluated methods have approximately the same number of learnable parameters as shown.
Table 1. Experiments results on FashionMNIST vs MNIST, FashionMNIST vs NotMNIST, and CIFAR-10 vs SVHN datasets. Compared baselines are the DUQ [5], and the Gaussian classifier [6]. Reported results are computed over 5 random seeds.

Table 2. Number of parameters and runtime of the evaluated methods. The runtime refers to the computational time of one forward propagation for a mini-batch containing 500 samples, and is computed based on a computer with an Intel(R) Core(TM) i9 @ 2.50GHz, 2496MHz 8 core processor, and a RTX 3090 graphic card with 64GB RAM and 24GB GPU memory.

Table 3. Experiments results of BE-SNNs with different ensemble sizes \( N_e \in \{2, 4, 8\} \) on the FashionMNIST vs MNIST and FashionMNIST vs NotMNIST datasets. Increasing the ensemble size \( N_e \) will ideally leads to improved performance, we suspect that the decreasing batch size is the reason for the similar performances of \( N_e = 4 \) and \( N_e = 8 \). The mean and standard deviation are calculated over 5 random seeds.

4.3. Ablation Study

We conduct an ablation study on the FashionMNIST vs MNIST dataset to investigate the effect of the adaptively tempered softmax and the gradient penalty regularization on the performance of BE-SNNs. From Figure 3, we can observe that BE-SNNs equipped with the adaptively tempered softmax consistently outperform BE-SNNs with standard softmax activations, implying that the proposed adaptively tempered softmax can indeed improve the OOD detection performance of BE-SNNs. Regarding the gradient penalty regularization coefficient \( \lambda_2 \), from Figure 3, we found that the best performance of BE-SNNs occurs when \( \lambda_2 = 0.5 \), and the performance of BE-SNNs is not sensitive to the value of \( \lambda_2 \) in the candidate set \( \{0.0, 0.1, 0.5, 1.0\} \). In addition, from Figure 3, we observe that all positive values of \( \lambda_2 \) lead to better performance of BE-SNNs than \( \lambda_2 = 0.0 \), showing that gradient penalty can enhance BE-SNNs’ ability to detect OOD samples.

5. CONCLUSION

In this work, we proposed the batch-ensemble stochastic neural networks (BE-SNNs), an OOD detection approach that incorporates the batch-ensemble mechanism with a novel single forward pass uncertainty quantification framework. By aggregating the predictions given by different ensemble members, BE-SNNs are algorithmically designed to overcome the feature collapse problem with deterministic single-forward pass models. Besides, BE-SNNs are memory efficient and have low computational cost in comparison with related approaches. We evaluated the performance of BE-SNNs on several OOD detection benchmarks and compared BE-SNNs with other state-of-the-art OOD detection approaches. Experiment results showed that BE-SNNs demonstrate superior performance on both toy and real-world image datasets over the other evaluated methods.
6. REFERENCES

[1] J. Ren et al., “Likelihood ratios for out-of-distribution detection,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), Vancouver, Canada, Dec. 2019.

[2] Z. Zhou et al., “Step: Out-of-distribution detection in the presence of limited in-distribution labeled data,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), Dec. 2021.

[3] Y. Sun, C. Guo, and Y. Li, “React: Out-of-distribution detection with rectified activations,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), Dec. 2021.

[4] W. Liu, X. Wang, J. Owens, and Y. Li, “Energy-based out-of-distribution detection,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), Dec. 2020, pp. 21464–21475.

[5] J. van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal, “Uncertainty estimation using a single deep deterministic neural network,” in Proc. International Conference on Machine Learning (ICML), July 2020.

[6] W. Wan, Y. Zhong, T. Li, and J. Chen, “Rethinking feature distribution for loss functions in image classification,” in Proc. IEEE conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, USA, June 2018.

[7] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), Long Beach, USA, Dec. 2017.

[8] Y. Wen, D. Tran, and J. Ba, “BatchEnsemble: an alternative approach to efficient ensemble and lifelong learning,” in Proc. International Conference on Learning Representations (ICLR), New Orleans, USA, May 2019.

[9] Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L. Willke, “Out-of-distribution detection using an ensemble of self supervised leave-out classifiers,” in Proc. European Conference on Computer Vision (ECCV), Munich, Germany, Sep. 2018.

[10] H. Choi, E. Jang, and A. A. Alemi, “Waic, but why? generative ensembles for robust anomaly detection,” arXiv preprint arXiv:1810.01392, 2018.

[11] J. Mukhoti, J. van Amersfoort, P. HS Torr, and Y. Gal, “Deep deterministic uncertainty for semantic segmentation,” in ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning, July 2021.

[12] J. van Amersfoort, L. Smith, A. Jesson, O. Key, and Y. Gal, “On feature collapse and deep kernel learning for single forward pass uncertainty,” in NeurIPS workshop on Bayesian Deep Learning workshop, Dec. 2021.

[13] N. Durasov, T. Bagaudinov, P. Baque, and P. Fua, “Masksem- bles for uncertainty estimation,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2021.

[14] M. Dusenberry et al., “Efficient and scalable Bayesian neural nets with rank-1 factors,” in Proc. International Conference on Machine Learning (ICML), July 2020.

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

[16] A. Krizhevsky, “Learning multiple layers of features from tiny images,” Tech Report, 2009.

[17] N. Yuval, W. Tao, C. Adam, B. Alessandro, W. Bo, and Andrew Y. N., “Reading digits in natural images with unsupervised feature learning,” in NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning, Granada, Spain, Dec. 2011.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[19] D. Lombardi and S. Pant, “Nonparametric k-nearest-neighbor entropy estimator,” Physical Review E, vol. 93, no. 1, pp. 013310, 2016.

[20] C. Guo et al., “On calibration of modern neural networks,” in Proc. International Conference on Machine Learning (ICML), Sydney, Australia, Aug. 2017.

[21] H. Wei, R. Xie, H. Cheng, L. Feng, B. An, and Y. Li, “Mitigating neural network overconfidence with logit normalization,” in Proc. International Conference on Machine Learning (ICML), Baltimore, USA, July 2022.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, USA, June 2016.