OUT-OF-DISTRIBUTION IMAGE DETECTION WITH DEEP NEURAL NETWORKS

V. Sreeramdass, S. Sarawagi
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Powai, India
{varshiths,sunita}@cse.iitb.ac.in

ABSTRACT

Out-of-distribution detection is vital in the deployment of Deep Neural Networks in practical applications. Several methods exist that range from temperature scaling of logits to developing an ensemble of models. This work surveys existing methods for out-of-distribution detection that either use features extracted by pre-trained models or propose minimal modifications to the architecture, analyses their performance, shortcomings and proposes a few simple enhancements with that produce comparable results.

1 INTRODUCTION

Understanding and estimating the uncertainty in a model is vital to studying the deployment of deep neural networks, especially in practical settings. Uncertainty estimates are crucial in some cases (DeVries & Taylor (2018b)) to judge if a human evaluator has to step in. In other words, the model has to know what it doesn't know.

As Guo et al. (2017) demonstrated, several deep networks make wrong predictions with very high confidence. This is the issue of mis-calibration where the confidence of the prediction does not represent the likelihood of the prediction. Several quite effective methods for calibrating neural networks exist in literature (Lakshminarayanan et al. (2017), Pereyra et al. (2017), Kumar et al. (2018)) that work with regularization techniques, priors on weights and several others.

The problem of calibrating neural networks is related to out-of-distribution (OOD) detection where the model is capable of detecting samples that are drawn from distributions different from the training distribution.
It is possible to train a simple binary classifier to distinguish samples from in- and out- distributions. However, these models tend to be biased towards the chosen out-distribution and cannot perform well with samples from arbitrary distributions.

Lee et al. (2017) demonstrates the effectiveness of using synthesized OOD samples as a reference out-distribution by modeling in-distributions using GANs Goodfellow et al. (2014) and sampling from the boundary regions of the distribution. This method however suffers from training instability and all other issues related to GANs. Moreover, GANs have been very effective in image applications but not so much in other domains (Rajeswar et al. (2017)).

It is thus required to perform OOD detection in the absence of a reference out-distribution.
Few methods use features extracted from the input using a pre-trained model (Hendrycks & Gimpel (2016), Lee et al. (2018), Gal & Ghahramani (2016)) while others use these features in conjunction with the learned priors (Alemi et al. (2018)), to derive an OOD confidence estimate. It is however possible that the extracted features are not always characteristic of the in-distribution. Either features from the initial layers of the pre-trained models can be used or dedicated models like VAEs (Kingma & Welling (2013)) can be used to obtain useful latent space representations of the images.

Contribution: This work implements a few methods for OOD detection in the context of image classification and provides insights into those that are promising and proposes a few enhancements that improve their OOD detection performance.

2 Problem Setting

Given IID samples $(x_i, y_i) \in D$, where D is the in-distribution and $y \in \{1, 2...k\}$, and a test example $\tilde{x}$, our task is to predict the label $\tilde{y}$ and a confidence score $r \in [0, 1]$ that indicates the if the input is in- or out-of-distribution. $r = 1$ indicates that the input is in-distribution with 100% confidence while $r = 0$ indicates otherwise.

3 Methods

This work explores the following baseline methods and a enhancements for a few of them.

Baseline and Temperature Scaling Baseline methods in Hendrycks & Gimpel (2016), suggest using the softmax score of the predicted class $S(\tilde{y}, \tilde{x}; T)$ as the confidence value r after scaling the logits by temperature T. Temperature helps distribute the confidence values to reflect high likelihood while the softmax score indicates the confidence of the predicted class.

$$S(y, \tilde{x}; T) = \frac{exp(p_{\tilde{x},y}/T)}{\Sigma_{j=1}^{k} exp(p_{\tilde{x},j}/T)} \tag{1}$$

where $p_{\tilde{x},y}$ are the logits for each of the classes.

ODIN Liang et al. (2017) proposes using the softmax score of the predicted class on perturbed input $S(\tilde{y}, \tilde{x}^{'})$. Inputs are perturbed to enhance the confidence score on the particular sample. The increase in the softmax score is expected to be negligible if the sample is OOD, and considerable otherwise, assisting better separation in the confidence scores for in- and out-of-distribution samples.

$$\tilde{x}^{'} = \tilde{x} - \epsilon sign(-\nabla_{\tilde{x}} log S(\tilde{y}, \tilde{x}; T)) \tag{2}$$

BIN Baseline and ODIN expect the model to assign flat distributions for OOD samples. It is reasonable to model the task as a multi-label classification with binary classifiers for each of the classes so that the confidence scores represent the absolute likelihood each of the classes and the values are not constrained to sum to one. The softmax activation is replaced with a sigmoid. Confidence r is the maximum of probabilities across classes.
To mitigate the effect of class imbalance for each binary classifier (the positive and negative samples for each of the classifiers are in the ratio k-1:1), the term for the negative examples is weighted by a tune-able hyperparameter ζ.

$$L(\theta) = \underset{(x_i,1)\in D}{\Sigma} \frac{exp(p_{x_i})}{1 + exp(p_{x_i})} + \zeta \underset{(x_i,0)\in D}{\Sigma} \frac{1}{1 + exp(p_{x_i})} \tag{3}$$

BINC The use of softmax is motivated by better accuracy as against sigmoid activation. To preserve accuracy performance while improving the OOD performance, we add a separate branch that performs multi-label classification like the above method.
To implement this, the size of the output space of the primary model is increased to z. Two sets of *logits* are mapped to from this output space independently. One of them is trained with a regular cross entropy with a softmax activation, while the other is trained as BIN, with trainable calibration loss (Kumar et al. (2018)) for each binary classifier added to the objective.

$$L(\theta) = K(\theta) + L_{BIN}(\theta) + \lambda MMCE(\theta) \tag{4}$$

K is the standard cross entropy loss used for classification with one set of logits; L_{BIN} is the same objective used in the BIN method for the second set of logits; $MMCE$ is the re-weighted estimate from Kumar et al. (2018).

VIB Alemi et al. (2016) models the task as variational inference where a latent distribution $P_{\theta_{\tilde{x}}}$ over $\mathbb{R}^z$ is obtained for each input from the primary model. Samples from latent distribution are used to obtain confidence scores for each of the classes. The information in the latent distribution is constrained by bottleneck-ing it through a prior P_ϕ. The model and the prior are thus trained to most efficiently capture the distribution for all training samples.

Alemi et al. (2018) subsequently demonstrated the utility of KL-divergence between the prior distribution and the predicted distribution of the latent representation of the input in OOD detection. The confidence score r is obtained from KL-divergence as below.

$$r(\tilde{x}) = exp(-KL(P_{\theta_{\tilde{x}}}, P_{\phi})) \tag{5}$$

VIBY It might be *difficult* for the prior P_{ϕ} to capture the entire in-distribution. Considering the diversity in the features across classes in the training set, separate priors P_{ϕ_y} for each class are maintained. Confidence is calculated using two methods: one where the KL-divergence is the weighted average of KL-divergences across classes, weighted by the predicted confidences and the other where the KL-divergence is the minimum across classes.

$$r_I(\tilde{x}) = exp(-\underset{y}{\Sigma} P(y|\tilde{x}) KL(P_{\theta_{\tilde{x}}}, P_{\phi_y})) \tag{6}$$

$$r_{II}(\tilde{x}) = exp(-\underset{y}{min} KL(P_{\theta_{\tilde{x}}}, P_{\phi_y})) \tag{7}$$

4 Evaluation Metrics

We adopt the following four metrics to measure the effectiveness of a neural network in distinguishing in- and out-of-distribution images.

FPR at 95% TPR can be interpreted as the probability that a negative example is mis-classified as positive when the the true positive rate (TPR) is as high as 95%. True positive rate can be computed by TPR = TP/(TP+FN) where TP and FN denote true positives and false negatives respectively. The false positive rate (FPR) can be computed by FPR = FP / (FP + TN), where FP and TN denote false positives and true negatives respectively.

Detection Error , i.e., P_e measures the mis-classification probability when TPR is 95%. The definition of P_e is given by P_e = 0.5(1-TPR) + 0.5FPR, where we assume that both positive and negative examples have the equal probability of appearing in the test set.

AUROC is the Area under the Receiver-Operator Characteristic, which is also a threshold-independent metric. The ROC curve depicts the relationship between TPR and FPR. The AUROC can be interpreted as the probability that a positive example is assigned a higher detection score than a negative example. A perfect detector corresponds to an AUROC score of 100%.

AUPR is the Area under the Precision-Recall curve, which is another threshold independent metric. The PR curve is a graph showing the precision=TP/(TP+FP) and recall=TP/(TP+FN) against each other. The metric AUPR-In and AUPR-Out denote the area under the precision-recall curve where in-distribution and out-of-distribution images are specified as positives, respectively.

5 Experiments

5.1 Architecture and training configurations

For the primary model, we adopt Wide ResNet (Zagoruyko & Komodakis (2016)) architecture with depth 28 and width 10 and no dropout layers. The hyper parameters are set identical to the original Wide ResNet (Zagoruyko & Komodakis (2016)). The networks are trained with stochastic gradient descent with Nestrov momentum for 200 epochs with batch size 128 and momentum 0.9. The learning rate starts at 0.1 and is dropped by a factor of 0.1 at 50% and 75% of the training progress. The model is implemented in Tensorflow and takes 6 hours to train on a Titan X GPU.

5.2 Datasets

The in-distribution dataset is CIFAR10 which is split into train/validation/test sets with 45k/5k/10k images in each. The various out-distributions used for validation and testing are listed below.

Out-of-distribution dataset	Method	FPR & 95% TPR ↓	Detection Error ↓	AUROC ↑	AUPR-IN ↑	AUPR-OUT ↑
Gaussian Noise	Baseline	65.7	35.37	90.92	95.87	87.33
	Baseline+T	0.0	2.54	99.83	99.87	99.77
	ODIN	0.0	2.54	99.83	99.87	99.77
	BIN	62.1	33.6	93.16	95.76	84.23
	BINC	68.0	36.54	93.33	95.93	85.45
	VIB	17.2	11.12	88.93	99.92	81.89
	VIBY-I	0.5	2.78	98.92	99.85	97.96
	VIBY-II	0.2	2.59	99.44	99.9	98.96
Tiny ImageNet (crop)	Baseline	61.4	33.22	70.06	85.88	86.49
	Baseline+T	41.8	23.41	93.03	94.38	91.2
	ODIN	25.8	15.41	95.85	96.51	95.23
	BIN	59.6	32.29	89.35	91.32	86.34
	BINC	50.3	27.70	93.15	95.00	89.65
	VIB	38.2	21.62	85.37	92.54	79.12
	VIBY-I	43.5	24.28	92.54	93.69	90.21
	VIBY-II	43.2	24.09	92.85	93.78	91.16
Tiny ImageNet (resize)	Baseline	64.8	34.92	63.74	81.5	85.31
	Baseline+T	55.5	30.22	90.05	91.87	87.86
	ODIN	46.4	25.65	91.71	93.01	90.17
	BIN	62.0	33.55	86.34	86.77	84.18
	BINC	58.8	31.94	90.49	92.49	86.91
	VIB	52.1	28.57	83.45	88.75	77.52
	VIBY-I	54.1	29.55	89.77	90.55	87.48
	VIBY-II	52.2	28.59	90.03	90.59	88.31

Table 1: Out-of-distribution detection performance for various OOD sets

Gaussian Noise The synthetic Gaussian noise dataset consists of 1500 random 2D Gaussian noise images, where each RGB value of every pixel is sampled from an i.i.d Gaussian distribution with mean 0.5 and unit variance which are further clipped into the range [0, 1].

Tiny Image Net The Tiny ImageNet dataset consists of a subset of ImageNet images. It contains 10,000 test images from 200 different classes. We sample 1500 random images and construct two datasets TinyImageNet (crop) and TinyImageNet (resize), by either randomly cropping image patches of size 32 x 32 or downsampling each image to size 32 x 32.

The test/val split for the above datasets is 1000/500.

5.3 Hyper-parameter tuning

For **Baseline+T** and **ODIN**, T (for both) and ϵ (for ODIN) are tuned to minimize FPR-at-95%-TPR on the OOD validation set. For **BIN** and **BINC**, ζ is set to 0.05. For **BINC**, λ is set to 0.5. For **VIB**, β is set to 0.005 and for **VIBY**, β is set to 0.01. The same βs surprisingly minimize FPR-at-95%-TPR for all OOD validation sets. z used for **BINC**, **VIB** and **VIBY** is set to 64.

6 Results

Accuracy The accuracy drops with the use of BIN (2). This is a result of the change in activation of the logits. The accuracy for other models is close to the baseline and not affected much.

Out-of-distribution detection Among all the methods, **ODIN** has the best performance among all methods.

Baseline+T is understandably better than **Baseline** due to temperature scaling and better calibrated confidence scores. It is however worse than **ODIN** due to lack of input perturbation and enhance-

Method	Accuracy
Baseline	93.76
BIN	92.54
BINC	93.66
VIB	93.71
VIBY	93.63

Table 2: Classification performance of various methods

ment. This effect is not observed for Gaussian noise due to the perturbation resulting in a similar input.

BINC is a slight improvement over **BIN**. This is presumably because each of the binary classifiers are better calibrated and represent confidences with high likelihood.

VIB proves to be a decent method for OOD detection, however not the best. **VIBY** are better than **VIB** for all datasets in terms of non-threshold based metrics. This is justified by the higher amount of information retained in the class wise priors as against the information retained in the single prior. **VIBY-II** performs better as compared to **VIBY-I** due to possible mis-calibration of the confidences that weight the KL-divergences.

7 Conclusion

The proposed enhancements seem to work in improving the respective primitive methods for the few datasets that have been experimented with. However more rigorous experimentation is required with a wider range of OOD sets.
Even though the proposed methods are far behind the state of the art **ODIN**, perturbations to the input are only intuitive in the context of images.

Even though other methods that perform better than the proposed methods do exist (Lee et al. (2018)), they can be too resource intensive. The proposed methods are more compact, general and applicable to other domains as well and are expected to perform better. The pursuit of this work is to develop a general technique that works for all deep neural networks and these methods offer a greater promise. We expect future research to provide us with more compact methods that tackle OOD detection along the lines of the proposed methods.

References

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Alexander A Alemi, Ian Fischer, and Joshua V Dillon. Uncertainty in the variational information bottleneck. *arXiv preprint arXiv:1807.00906*, 2018.

Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018a.

Terrance DeVries and Graham W Taylor. Leveraging uncertainty estimates for predicting segmentation quality. *arXiv preprint arXiv:1807.00502*, 2018b.

Yarin Gal. Uncertainty in deep learning.

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

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*, 2017.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

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

Aviral Kumar, Suita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pp. 2810–2819, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 6402–6413, 2017.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pp. 7165–7175, 2018.

Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

Amit Mandelbaum and Daphna Weinshall. Distance-based confidence score for neural network classifiers. *arXiv preprint arXiv:1709.09844*, 2017.

Marc Masana, Idoia Ruiz, Joan Serrat, Joost van de Weijer, and Antonio M Lopez. Metric learning for novelty and anomaly detection. *arXiv preprint arXiv:1808.05492*, 2018.

Azadeh Sadat Mozafari, Hugo Siqueira Gomes, Steeven Janny, and Christian Gagné. A new loss function for temperature scaling to have better calibrated deep networks. *arXiv preprint arXiv:1810.11586*, 2018.

Marcin Możejko, Mateusz Susik, and Rafał Karczewski. Inhibited softmax for uncertainty estimation in neural networks. *arXiv preprint arXiv:1810.01861*, 2018.

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.

Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Pal, and Aaron Courville. Adversarial generation of natural language. *arXiv preprint arXiv:1705.10929*, 2017.

Apoorv Vyas13, 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.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.