Thinkback: Task-Specific Out-of-Distribution Detection

Lixuan YANG [1] **Dario ROSSI** [1]

Abstract

The increased success of Deep Learning (DL) has recently sparked large-scale deployment of DL models in many diverse industry segments. Yet, a crucial weakness of supervised model is the inherent difficulty in handling out-of-distribution samples, i.e., samples belonging to classes that were not presented to the model at training time.

We propose in this paper a novel way to formulate the out-of-distribution detection problem, tailored for DL models. Our method does not require fine tuning process on training data, yet is significantly more accurate than the state of the art for out-of-distribution detection.

1. Introduction

Irrespectively of the specific application, classes and inputs, classification engines needs to perform two functions: namely an *identification function* $f(x)$ and an *open-set detection function* $g(x)$. Shortly, the goal of $\ell = f(x)$ is to determine from an input x, a label $\ell \in [1, K]$ among a set of K known classes. Supervised ML/DL techniques are well suited to learn the function $f(x)$, in a process referred to as training, where the parameters W of the function $f(x)$ are adapted (e.g., through backpropagation in case of DL models). The goal of $g(x)$ is instead to detect open-set inputs x', i.e., inputs that do not belong to any of the K classes known to $f(x)$.

The problem tackled by $g(x)$ is usually referred to as *out of distribution* (OOD) detection, and has recently gathered significant attention in the literature for a variety of domains, ranging from network(Yang et al., 2021), natural language processing(Miok et al., 2020) to medical field (Cao et al., 2020). Given this variety of applications domains, ideally OOD techniques should be broadly applicable to any DL model deployed in production, and require as few assistance as possible. Additionally, OOD techniques computational complexity should be small, as if $g(x)$ is slower than $f(x)$, then this would either limit the classification rate of $f(x)$, or means that OOD could be applied only to some samples, i.e., trading off classification rate performance with OOD detection accuracy. Conscious of the above constraints, in this paper we propose "Thinkback", a novel gradient-based method for task-specific OOD that does not require any specific tuning, is computationally very lightweight, yet very accurate. In a nutshell, after having classified a new instance x with the feed-forward network inference $f(x) = \ell$, our method infer the plausibility of ℓ by pretending ℓ to be a ground truth label, and assessing how much this new label would change model weights, by performing the initial steps of a backpropagation – where we expect weights to change little if ℓ is a plausible class for x, and change much in case x is an OOD sample.

In the reminder of this short paper, we overview the state of the art (Sec. 2), we present (Sec. 3) our Thinkback method and benchmark (Sec. 4) our proposed Thinkback with the state of the art methods and summarize our findings (Sec. 5).

2. Related work

We summarize the relevant literature with the help of Fig. 1, that depicts a visual synoptic where related work is divided into three categories, based on whether OOD detection is performed on input x, output $\ell = f(x)$ or in the inner-stages of the DL model.

2.1. Input

Works have recently proposed to apply transformations on the input so to control models outputs regulated by mechanisms such as Mahalanobis based score (Lee et al., 2018) or temperature scaled SoftMax score (Liang et al., 2018). There are two drawback of those proposals: i) their known additional computational cost makes them unappealing from practical perspective and ii) the fact that input transformation should be concerted prior to DL model deployment, for which work in the remaining classes is more interesting.

[1]Huawei Technologies, France. Correspondence to: Lixuan YANG <lixuan.yang@huawei.com>, Dario ROSSI <dario.rossi@huawei.com>.

Presented at the ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning., Copyright 2021 by the author(s).

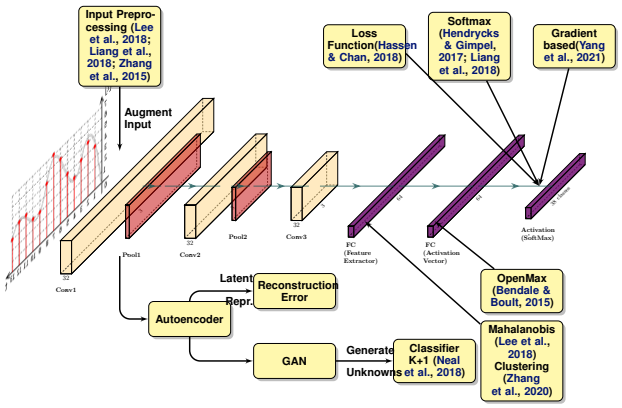

Figure 1. Synoptic of OOD detection wrt an exemplary convolutional DL model: we classify OOD techniques as either working at the input/output of the DL model, or by requiring modifications to the inner models.

2.2. Inner

At their core, DL methods project input data into a *latent space* where is easier to separate data based on class labels. A set of work then proposes specific ways to alter this latent space to purposely simplify open-set recognition.

For instance, (Zhao et al., 2019) uses AutoEncoders (AE) to transform input data, and apply clustering to the transformed input, while (Yoshihashi et al., 2019) uses latent representation along with OpenMax (Bendale & Boult, 2015) activation vectors. Other works instead rely on Generative Adversarial Networks (GAN) to explore the latent space in order to generate "unknown classes" data to train a classifier for the class $\ell = 0$ i.e., a K+1 classifier. For instance, (Ge et al., 2017) generate unknown classes by mixing the latent representation of known classes, while (Neal et al., 2018) uses optimisation methods to create counterfactual samples that are close to training samples but do not belong to training data. All these methods require specific architectures (so they are hardly deployable) and extra training (so their computational complexity can be high).

Other work propose to alter activation (Jang & Kim, 2020) or loss functions (Aljalbout et al., 2018; Hassen & Chan, 2018). In (Jang & Kim, 2020) authors replace the SoftMax activation with a sigmoid, and fit a Weibull distribution for each activation output to revise the output activation. Special clustering loss functions (Aljalbout et al., 2018; Hassen & Chan, 2018) can be used to further constraint points of the same class to be close to each other, so that unknown classes are expected to be projected into sparse region which is far from known classes. However, all these methods constrain to use special DL architectures and cannot be used on existing models; additionally such architectural modifications can alter the accuracy of the supervised classification task. As such, we deem it difficult for techniques of this class to actually broadly deployable, and we disregard them in what follows.

2.3. Output

The most common approach for OOD detection at output stage is thresholding SoftMax values (Hendrycks & Gimpel, 2017). OpenMax (Bendale & Boult, 2015) revises SoftMax activation vectors adding a special "synthetic" unknown class, by using weighting induced by Weibull modeling of input data. Alternative approaches include the use of Extreme Value Machine (EVM) (Rudd et al., 2018), based of Extreme Value Theory (EVT), and, more recently, clustering on the CNN feature vectors with a PCA reduction of dimension (Zhang et al., 2020). The authors (Wang et al., 2020) adjusts to few shot classification by using SPP (Hendrycks & Gimpel, 2017) and Mahalanobis distance (Lee et al., 2018). Finally, as a collection of energy values could be turned into a probability density though Gibbs distribution, (Liu et al., 2020) formulates the energy function by the denominator of the SoftMax activation, so that energy scores align with the probability density. This method shows nearly perfect performance after fine tuning on the OOD data – however, in reality collecting OOD data in advance is impossible.

Our proposed methods, based on evaluating gradients change via backpropagation, also fit in this class: this makes techniques in this class particularly relevant and worth considering for a direct performance comparison.

We select the best-in class approach, that represents the state of the art and namely Energy score (Liu et al., 2020), that authors shows to have superior performance to (Hendrycks et al., 2019; Lee et al., 2018; Liang et al., 2018). To better assess the added value of Energy score and our method, we additionally consider classic SoftMax outputs (Hendrycks & Gimpel, 2017) as a reference. Based on a preliminary evaluation on additional techniques that were not considered in (Liu et al., 2020), we instead discard OpenMax (Bendale & Boult, 2015), which we find to be significantly less accurate than Energy score. We also discard (Zhang et al., 2020), which in reason of the PCA and clustering step, is significantly more complex from a computational standpoint. Thus, we resort to comparing Energy score, SoftMax and our proposed Thinkback method, for which accuracy comparison can be done on a fair ground in terms of complexity.

3. Methodology

Since the unknown data may take infinite forms, modeling unknown data without any assumption is difficult; conversely, constraining unknown data by specific assumptions may results in weak OOD detection capabilities in the general case.

To circumvent this conudrum, we let a DL model trained

on D_{in}^{train} to *introspect* its feed-forward decision based on the information extracted solely out of model weights W. In a nutshell, we ask the model to *think back* about its feed-forward decision by leveraging *backpropagation*, assessing the magnitude of the change that would happen to model weights if they were altered W' by training also on test data D_{test}. Thus, we transform the OOD problem into compute the probability $p(W'|D_{in}^{train})$ of getting W' with a model trained on D_{in}^{train}, as we detail next.

3.1. Task specific novelty detection

In more details, Elastic weight consolidation (Kirkpatrick et al., 2016) approximates the posterior probability of $p(w_i|D_{train})$ as a Gaussian distribution with mean given by μ_i and a diagonal precision p_i given by the diagonal of the Fisher information matrix F. After training on D_{in}^{train}, the weights w_i are stable around the μ_i. The probability of a weight, given the training dataset D_{in}^{train} can be written as:

$$p(w_i|D_{in}^{train}) = \sqrt{\frac{F_i}{2\pi}} \exp\left(-\frac{1}{2}(w_i - \mu_i)^2 F_i)\right) \quad (1)$$

When new data $x \in D_{test}$ comes, in addition to performing feed-forward inference $\ell = f(x)$, we let the model think back about this decision. To do so, we initiate a (fake) backpropagation step, as if the classification outcome ℓ was a ground truth label (but without altering the model weights): the new data would induce changes on model weights as $(w_i)' = w_i - \delta w_i$. We then assess the probability of this newly changed weight when x is an in-distribution sample, which can be written as:

$$p({w_i}'|D_{in}^{train}) = \sqrt{\frac{F_i}{2\pi}} \exp\left(-\frac{1}{2}(\delta w_i)^2 F_i)\right) \quad (2)$$

Since the weights distribution are independent, we have:

$$p({w_1}', {w_2}', ..., {w_n}'|D_{IN}^{train}) = \prod_i p({w_i}'|D_{in}^{train}) \quad (3)$$

so that the whole weights distribution is proportional to:

$$p(W'|D_{in}^{train}) \propto \prod_i \exp\left(-\frac{1}{2}(\delta w_i)^2 F_i\right) \propto -\sum_i (\delta w_i)^2 \quad (4)$$

To limit computation complexity, we limit backpropagation to the penultimate layer, which contains most information concerning the classes. The intuition is that shall the new sample belong to an OOD class that was never seen at training, weights of the penultimate layer which is task-specific should have large changes. Ultimately, we define the unknown level of a sample x as the reverse of the above probability:

$$g(x \in D_{out}^{test}) \propto \sum_i (\delta w_i)^2 \quad (5)$$

While our method do not require to fine-tune on OOD data, as for instance (Liu et al., 2020) (which is however unavailable at training time) it can however benefit from in-distribution training data (which is easily accessible at training time). In particular, during the training phase, despite the local minimum has been reached, the training data still generate gradients. In order to reduce the gradient that follows the in-distribution training data's gradient trend, the gradient is divided by the expected gradient of the training.

$$g(x \in D_{out}^{test}) = \sum_i \frac{(\delta w_i)^2}{\epsilon + E(\delta w_i^2|D_{in}^{train})} \quad (6)$$

where ϵ is a technicality to avoid division by zero in the rescaling. Additionally, as suggested by ODIN (Liang et al., 2018), temperature scaled Softmax score helps to separate the in- and out-of-distribution: we thus backpropagate the scaled Softmax by temperature T.

$$\delta w_i = \frac{\partial L}{\partial w_i} = -\frac{\partial \sum_{i=1}^{K} y_i \log(Softmax(z_i/T))}{\partial w_i} \quad (7)$$

By trusting the network's prediction y_i, the gradient is the partial derivative of the loss function L on the network softmax activation z_i scaled by T over the total number of classes K.

4. Evaluation

4.1. Settings

We study OOD performance using classic models and datasets for image recognition. In particular for the $f(x)$ classification function, we resort to WideResNet (Zagoruyko & Komodakis, 2016), which is known to provide state-of-the-art results on CIFAR and significant improvements on ImageNet, with 16-layer-deep wide residual network outperforming in accuracy even thousand-layer-deep networks. To thoroughly assess OOD performance, we therefore train WideResNet on CIFAR10 (Krizhevsky, 2009) as in-distribution, and use multiple datasets as out-of-distribution – namely, Textures (Cimpoi et al., 2013), SVHN (Netzer et al., 2011), LSUN-Crop (Yu et al., 2015), LSUN-Resize (Yu et al., 2015), iSUN (Xu et al., 2015) that are also used in (Liu et al., 2020).

To carry on a fair algorithmic comparison, we do not allow fine-tuning based on OOD datasets. As such, we compare with Energy score without the fine tuning process –as otherwise the OOD data would be known at training time, questioning whether it would be more adviseable to include such data for training– and use the default setting proposed in (Hendrycks & Gimpel, 2017). For our implementation, we have selected the temperature which has the smallest standard deviation on in-distribution's validation data from $T \in [1, 5]$ (specifically $T = 5$), and set $\epsilon = 10^{-16}$.

Table 1. OOD detection performance of Thinkback, our proposed gradient-based method, compared with vanilla Softmax (Hendrycks & Gimpel, 2017) and state of the art Energy Score (Liu et al., 2020). $\uparrow, +$ denote metrics for which larger values are better, while $\downarrow, -$ indicate that smaller values are better. Δ shows the relative performance w.r.t Softmax.

Dataset	Method	TPR10 ↑	ΔTPR10 +	FPR95 ↓	ΔFPR95 −	AUROC ↑	ΔAUROC +	AUPR ↑ +	Δ AUPR
Mean Result	Softmax	71.83		30.81		91.43		66.45	
	Energy	80.51	*+8.68%*	36.62	*+5.81%*	92.32	*+0.89%*	75.75	*+9.30%*
	Thinkback	**84.37**	*+12.54%*	**22.65**	*-8.16%*	**94.17**	*+2.74%*	**92.42**	*+25.97%*
Textures	Softmax	62.00		43.23		88.66		59.04	
	Energy	65.15	*+3.15%*	68.79	*+25.56%*	85.03	*-3.63%*	59.80	*+0.76%*
	Thinkback	**78.23**	*+16.23%*	**32.09**	*-11.14%*	**92.35**	*+3.69%*	**91.73**	*+32.69%*
SVHN	Softmax	71.30		28.47		91.83		66.54	
	Energy	78.25	*+6.95%*	42.29	*+13.82%*	91.07	*-0.76%*	70.87	*+4.37%*
	Thinkback	**83.82**	*+12.52%*	**23.41**	*-5.06%*	**94.06**	*+2.23%*	**91.90**	*+25.4%*
LSUN Crop	Softmax	89.15		15.18		95.64		79.78	
	Energy	**96.35**	*+7.20%*	**6.72**	*-8.46%*	**98.44**	*+2.90%*	**93.55**	*+13.77%*
	Thinkback	89.14	*-0.01%*	14.09	*-1.09%*	95.12	*-0.52%*	91.86	*+12.08%*
LSUN Resize	Softmax	70.20		30.80		91.27		64.88	
	Energy	83.25	*+13.05%*	29.96	*-0.83%*	93.99	*+2.72%*	78.93	*+14.05%*
	Thinkback	**87.86**	*+17.66%*	**20.73**	*-10.07%*	**94.88**	*+3.61%*	**93.47**	*+28.59%*
iSUN	Softmax	66.50		36.37		89.76		61.99	
	Energy	79.55	*+13.05%*	35.34	*-1.03%*	93.06	*+3.30%*	75.63	*+13.64%*
	Thinkback	**82.82**	*+16.31%*	**22.95**	*-13.41%*	**94.42**	*+4.66%*	**93.13**	*+31.14%*

4.2. Metrics

As rightly observed in (Hendrycks et al., 2019), OOD methods should be evaluated on their ability to detect OOD points i.e., to focus on this capability, we consider OOD points as positive. Thus, a True Positive for the $g(x)$ function equals to a correctly detected out-of-distribution sample (i.e., a correct rejection of a wrong $f(x) = \ell$ classification), whereas a False Positive equals to a wrongly rejected in-distribution sample (i.e., a wrong rejection of a correct $f(x) = \ell$ classification).

We then evaluate OOD detection capabilities using the standard metrics of information retrieval, as follows. **AUROC** is the Area Under the Receiver Operating Characteristic (ROC) curve of False Positive Rate (FPR) and True Positive Rate (TPR) and **AUPR** is the area under the precision-recall curve. AUROC and AUPR evaluate the overall method capabilities over all operational settings. Additionally, we investigate the system ability to achieve high recall for a low false alarm rate by considering **TPR10**, i.e., the TPR when FPR=10%. This metric has a high practical relevance, since $g(x)$ OOD detection capabilities should not reject correct DL prediction, which would (doubly) waste computational power of $f(x)$ (and $g(x)$). Finally, we consider the false alarm rate at high recall rate **FPR95**, i.e., the FPR at TPR=95%. Again, in practical settings, this metrics well reflect the scenario where one may wants $g(x)$ to accurately collect as much OOD samples as possible, in order to update the $f(x)$ model.

4.3. Results

Experimental results are reported in Tab 1. In general, Thinkback outperforms the SoftMax baseline (Energy score state of the art) for TPR10, FRP95, AUROC and AUPR. The average improvement over SoftMax (Energy score) is 2.74% (1.85%) for AUROC and 25.97% (16.67%) for AUPR. By looking closer, at low FPR rate, Thinkback allows to detect 12.54% (3.86%) more OOD data than the SoftMax (Energy score). In order to detect 95% of OOD, Thinkback raises 8.16% (14.03%) less false alarm than the SoftMax baseline (Energy score). The gains are consistent across all datasets with the exception of LSUN Crop, for which Energy score provide better results.

As a side note, we report on the time complexity of the $g(x)$ methods: our non-optimized implementation on Pytorch v1.8, Thinkback takes 1.5 ms/sample on average (i.e., equivalent to 667 samples/sec), which is slightly slower but still comparable to the SoftMax and Energy score (both take 1.2 ms/sample or 833 samples/sec).

5. Discussion

We introduce Thinkback, a gradient-based method for OOD samples detection, fit to work with existing models (as it does not require to alter the network architecture), and in general settings (as it does not require fine-tuning on OOD data). Experimental results show Thinkback to be fast and accurate, providing superior results to the state of the art.

References

Aljalbout, E., Golkov, V., Siddiqui, Y., and Cremers, D. Clustering with deep learning: Taxonomy and new methods. *CoRR*, abs/1801.07648, 2018. URL http://arxiv.org/abs/1801.07648.

Bendale, A. and Boult, T. E. Towards open set deep networks. In *IEEE CVPR*, 2015.

Cao, T., Huang, C.-W., Hui, D. Y.-T., and Cohen, J. P. A Benchmark of Medical Out of Distribution Detection. In *Workshop on UDL*, 2020.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. *CoRR*, abs/1311.3618, 2013.

Ge, Z., Demyanov, S., Chen, Z., and Garnavi, R. Generative openmax for multi-class open set classification. *CoRR*, abs/1707.07418, 2017. URL http://arxiv.org/abs/1707.07418.

Hassen, M. and Chan, P. K. Learning a neural-network-based representation for open set recognition. *CoRR*, abs/1802.04365, 2018. URL http://arxiv.org/abs/1802.04365.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ICLR*, 2017.

Hendrycks, D., Mazeika, M., and Dietterich, T. Deep anomaly detection with outlier exposure. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

Jang, J. and Kim, C. O. One-vs-rest network-based deep probability model for open set recognition. *CoRR*, abs/2004.08067, 2020. URL https://arxiv.org/abs/2004.08067.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N. C., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016. URL http://arxiv.org/abs/1612.00796.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, 2009.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. 2018.

Liang, S., Li, Y., and Srikant, R. Principled detection of out-of-distribution examples in neural networks. *ICLR*, 2018.

Liu, W., Wang, X., Owens, J., and Li, Y. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21464–21475. Curran Associates, Inc., 2020.

Miok, K., Skrlj, B., Zaharie, D., and Sikonja, M. R. Bayesian BERT for Trustful Hate Speech Detection . In *Workshop on UDL*, 2020.

Neal, L., Olson, M., Fern, X., Wong, W.-K., and Li, F. Open set learning with counterfactual images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

Rudd, E. M., Jain, L. P., Scheirer, W. J., and Boult, T. E. The extreme value machine. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):762–768, 2018.

Wang, K.-C., Vicol, P., Triantafillou, E., and Zemel, R. Few-shot Out-of-Distribution Detection. In *Workshop on UDL*, 2020.

Xu, P., Ehinger, K. A., Zhang, Y., Finkelstein, A., Kulkarni, S. R., and Xiao, J. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *CoRR*, abs/1504.06755, 2015.

Yang, L., FINAMORE, A., FENG, J., and ROSSI, D. Deep learning for encrypted zero-day traffic classification. *arXiv 2104.03182*, 2021.

Yoshihashi, R., Shao, W., Kawakami, R., You, S., Iida, M., and Naemura, T. Classification-reconstruction learning for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Yu, F., Zhang, Y., Song, S., Seff, A., and Xiao, J. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *BMVC*, 2016.

Zhang, J., Chen, X., Xiang, Y., Zhou, W., and Wu, J. Robust network traffic classification. *IEEE/ACM Transactions on Networking*, 23(04):1257–1270, jul 2015. ISSN 1558-2566. doi: 10.1109/TNET.2014.2320577.

Zhang, J., Li, F., Ye, F., and Wu, H. Autonomous unknown-application filtering and labeling for dl-based traffic classifier update. In *INFOCOM 2020*, 2020.

Zhao, S., Zhang, Y., and Sang, Y. Towards unknown traffic identification via embeddings and deep autoencoders. In *2019 26th International Conference on Telecommunications (ICT)*, pp. 85–89, 2019.