Probabilistic Modeling of Deep Features for Out-of-Distribution and Adversarial Detection

Nilesh.A.Ahuja Intel
Ibrahima J. Ndiour Intel
Trushant Kalyanpur
Omesh Tickoo Intel

Abstract

We present a low-complexity approach for detecting out-of-distribution (OOD) and adversarial samples in deep neural networks. We model the outputs of the various layers (deep features) with parametric probability distributions (Gaussian and Guassian Mixture Models) once training is completed. At inference, the likelihoods of the observed features w.r.t the learnt distributions are calculated and used to discriminate in-distribution samples from OOD samples. We demonstrate that using our approach on the detection of OOD images and adversarially-generated images on MNIST and CIFAR10 datasets results in an improvement of up to 13 percentage points in AUPR and AUROC metrics over a state-of-the-art baseline.

1 Introduction

An important area of active research in the machine-learning is the ability of deep neural-networks (DNN) to estimate confidence or uncertainty measures, which quantify how much trust should be placed in DNN results. It has been shown that Softmax scores at the output of a deep-network tend to result in overconfident predictions, especially when the input does not resemble the training data (out-of-distribution)[3], or has been crafted to attack and "fool" the network (adversarial examples).

Recently, there has been substantial work on this topic using Bayesian deep learning [3, 6]. The parameters of a Bayesian Neural Network (BNN) are learned using variational training. At inference, multiple stochastic forward passes are performed to generate a distribution over the outputs, from which various measures of predictive uncertainty can be calculated [3]. Another class of methods, however, attempt to estimate uncertainty directly from a trained (non-Bayesian) DNN, and hence do not entail the computational overhead of multiple forward passes during inference. Some of these used posterior probabilities in the form of either softmax [5] or temperature-scaled softmax Liang et al. [8] to perform detection of OOD or misclassified samples. By contrast, Lee et al. [7] adopted a generative approach and proposed fitting class-conditional probability distributions to the features of a trained DNN. The confidence score was defined as the Mahalanobis distance with respect to the closest class conditional distribution. This approach yielded impressive results, outperforming [8] and [5].

Contribution: We present an approach for detecting OOD and adversarial samples in DNNs based on probabilistic modeling of the deep-features. Conceptually, our method extends the generative approach in [7] as we fit class-conditional distributions to the outputs of the various layers (deep-features). In [7], however, it is hypothesized that these distributions can be modeled as multivariate Gaussians with shared covariance across all classes (homoscedasticity assumption). We show that such an assumption is not valid in general; instead, we demonstrate that by removing this assumption and modeling the features with more general distribution types results in significantly improved detection of OOD and adversarial samples. Moreover, we demonstrate that the high-dimensional features actually reside in a low-dimensional subspace, and we use principal component analysis (PCA) as a principled way to capture this subspace, prior to the feature modeling.

4th workshop on Bayesian Deep Learning (NeurIPS 2019), Vancouver, Canada.

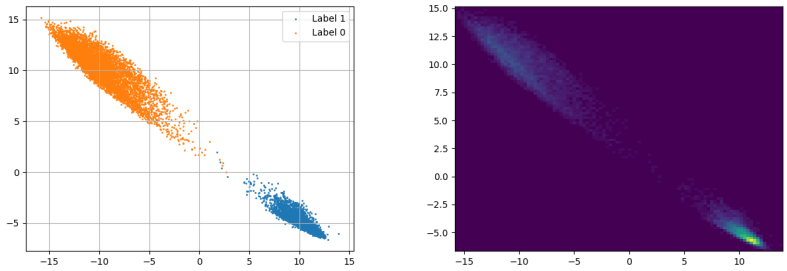

Figure 1: Scatterplot and the corresponding density histogram of the logits. It is clearly seen that the covariances of the two clusters are different.

2 Approach

Suppose we have a deep network trained to recognize samples from N classes, $\{C_k\}, k = 1, \ldots, N$. Let $f_i(\mathbf{x})$ denote the output at the i^{th} layer of the network, and n_i its dimension. Our approach consists of fitting class-conditional probability distributions, $p(f_i(\mathbf{x})|C_k)$, to the features of a DNN, once training is completed, thereby defining a generative model over the deep feature space. At test time, the log-likelihood scores of the features of a test sample are calculated with respect to these distributions and used as uncertainty estimates to discriminate in-distribution samples (which should have high likelihood) from OOD or adversarial samples (which should have low likelihood).

From the theory of linear discriminant analysis (LDA), it is known that in a generative classifier in which the underlying class-conditional distributions $p(f_i(\mathbf{x})|C_k)$ are Gaussians with tied covariance across all classes, the posterior distribution $p(C_k|f_i(\mathbf{x}))$ is equivalent to the softmax function with linear separation boundaries [1]. On the basis of this, Lee et al. [7] argued that the class-conditional densities of the penultimate layer of a deep-network could be well represented by multivariate Gaussian with shared covariance. As we demonstrate, though, this assumption does not hold true even in the simplest of networks. To test this, we constructed and trained a simple CNN architecture which we call MNET (shown in Figure 2) to classify only two digits ('0' and '1') from the MNIST dataset. A 2D density histogram of the features from the final layer (FC2) is shown in Figure 1. It is obvious even visually that the covariances of the two clusters are very different from each other.

In this work, therefore, we relax the assumption of tied covariance, and instead employ the following more general distributions: (a) Separate multivariate Gaussian distribution for each class without the assumption of a tied covariance matrix: this corresponds to the QDA (quadratic discriminant analysis) classifier (as opposed to the more restrictive LDA for tied-covariance Gaussian), and (b) Gaussian Mixture Model (GMM). Both of these are capable of representing larger classes of distributions. The parameters of these distributions are estimated using standard maximum-likelihood techniques.

Modeling within subspaces: According to the now well-known *manifold hypothesis*, real-world datasets exhibit low-dimensional structure despite being embedded in very high-dimensional input spaces [2, 9]. In the context of modeling distributions over high-dimensional features, this problem manifests itself as rank-deficiencies of data matrices, making it impossible to estimate the corresponding covariance matrices. This is a very real problem as will be shown in Table 4 in Section 3, which shows severe rank deficiencies in the higher-dimensional inner layers of the deep networks tested. To address this, we use PCA to reduce the dimensions of the feature vectors in order to model distributions in an appropriate lower-dimensional subspace.

3 Experiments and Results

Experimental setup and evaluation metrics: We use MNIST and CIFAR10 as the in-distribution datasets. We test each in-distribution dataset against two different out-of-distribution datasets, and the FGSM adversarial attack [4]. In all experiments, the parameters of the fitted densities are estimated from the training split of the in-distribution dataset, while performance metrics are calculated on the test split. During testing, the log-likelihood scores of the features generated by a test sample are

Table 1: AUROC scores (%) from three different density functions: GMM, Sep (Gaussian with separate covariance per class), Tied (Gaussian with tied covariance). Best values are shown in **bold**.

MNIST	FashionMNIST			EMNIST			FGSM, $\epsilon = 0.2$		
	GMM	Sep	Tied	GMM	Sep	Tied	GMM	Sep	Tied
Layer 0	**92.9**	91.8	92.1	**94.0**	93.2	91.9	**87.2**	85.9	84.4
Layer 1	92.9	**93.5**	75.3	96.2	**96.3**	93.4	**88.6**	85.7	66.8
Layer 2	97.0	**97.5**	89.0	96.7	**97.2**	93.1	**92.0**	92.0	62.6
CIFAR10 (Resnet)	SVHN			LSUN			FGSM, $\epsilon = 0.1$		
	GMM	Sep	Tied	GMM	Sep	Tied	GMM	Sep	Tied
Layer 0	**94.9**	93.4	92.9	**95.7**	95.1	94.8	**93.0**	92.6	93.0
Layer 1	**94.1**	92.3	91.6	**95.5**	94.7	94.1	**93.4**	93.2	93.3
Layer 2	90.2	90.1	**91.6**	87.8	**87.9**	78.1	**93.6**	93.6	93.1
CIFAR10 (Densenet)	SVHN			LSUN			FGSM, $\epsilon = 0.1$		
	GMM	Sep	Tied	GMM	Sep	Tied	GMM	Sep	Tied
Layer 0	76.7	75.2	**77.2**	69.5	**69.9**	64.8	85.7	**86.4**	83.6
Layer 1	**85.2**	84.7	73.3	94.8	95.2	**95.4**	**92.7**	92.5	90.3
Layer 2	**78.1**	78.1	51.1	**78.2**	78.2	74.8	**88.0**	88.0	80.5

calculated. These are then used to distinguish between in-distribution and out-of-distribution data, effectively creating a binary classifier. We characterize the performance of this classifier with the precision-recall (PR) curve and the receiver operating characteristics (ROC) curve.

For MNIST, we use the MNET architecture shown in Figure 2. For CIFAR10, we use two publicly available architectures: Resnet50 and Densenet-BC. We experimented on the final three layers of the networks listed above (in Densenet and Resnet, these are the outputs of the corresponding final 3 dense or residual blocks). The layers are labelled as 0, 1, and 2, (0 being the outermost layer).

Results: To see the performance on OOD samples, we calculate the AUROC scores as described earlier. The results are presented in Table 1. It is seen that the use of the more general distribution types typically results in improvements, often significant, in the AUROC scores over the baseline (tied-covariance) distribution. In Table 3, the average improvement (across all tested datasets) in the AUROC scores per layer are presented. It is seen that the extent of the improvement increases the further we are from the final output layer. This shows that the homoscedasticity assumption gets more violated as we move deeper into the network.

We also use the log-likelihood scores to perform classification instead of the softmax scores and measure the resulting classification accuracy. From the results summarized in Table 2, it is seen that the classification accuracy using the proposed method is comparable, if not slightly better, than the softmax-based accuracy, indicating that our scheme is as good as softmax for classification of in-distribution samples.

Computational aspects: We measure the time taken by the additional computations (PCA and log-likelihood calculations) required during inference only since the learning of class-conditional distributions is a one-time offline operation. We observe that these additional computations happen much faster than real-time rate. Specifically, for Gaussian models (tied or separate covariance), it takes on average an additional 1.32 ms to process each input frame (equivalent to about 757 frames/sec). For GMM models, the time increases to 2.75 ms/frame (363 fps). This is despite our current implementation being non-optimized Python code running on CPU. We expect significant speed-up with optimized implementations (e.g. in C/C++) using appropriate vectorization facilities available on modern processors.

4 Conclusions and Future Work

This paper presented a method for modeling the outputs of the various DNN layers (deep-features) with parametric probability distributions, with applications to adversarial and out-of-distribution sample detection. The methodology was theoretically motivated, and experimentally proven by

showing improvements to detection of OOD and adversarial image samples.As future work, we will investigate the use of more sophisticated techniques to model distributions on the manifold of features. We will also look into ways to combine log-likelihood scores across layers into a single reliable uncertainty metric.

Table 2: Classification accuracy

	MNIST			CIFAR10(Resnet)			CIFAR10(densenet)		
	GMM	Sep	Tied	GMM	Sep	Tied	GMM	Sep	Tied
Layer 0	98.9	98.6	98.6	**95.3**	95.2	**95.3**	**90.3**	90.0	89.8
Layer 1	98.2	98.6	98.6	95.0	**95.3**	**95.3**	89.1	88.7	89.7
Layer 2	86.0	97.4	98.3	90.8	90.8	92.2	88.9	88.9	89.9
Softmax	**98.99**			**95.3**			89.14		

Table 3: Average improvements in scores

	AUROC change	
	GMM	Sep
Layer 0	1.94	1.51
Layer 1	5.48	6.0
Layer 2	8.9	9.82

References

[1] Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

[2] Fefferman, C., Mitter, S., and Narayanan, H. (2016). Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049.

[3] Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059.

[4] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples.

[5] Hendrycks, D. and Gimpel, K. (2017). A baseline for detecting misclassified and out-of-distribution examples in neural networks.

[6] Kendall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584.

[7] Lee, K., Lee, K., Lee, H., and Shin, J. (2018). A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177.

[8] Liang, S., Li, Y., and Srikant, R. (2018). Enhancing the reliability of out-of-distribution image detection in neural networks.

[9] Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323.

Appendix

We present some additional results in this section. In particular, we report

- Rank of data matrices for layer 2 features showing severe rank-deficiencies.
- AUPR numbers in Table 5 to complement the AUROC numbers reported earlier. Here too, we observe the same trend in that the more general densities (Gaussian with separate covariance per class, and GMM) give superior results to the baseline distribution.

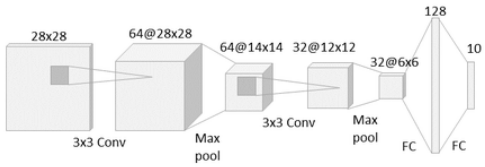

Figure 2: MNET architecture.

Table 4: Layer 2 feature dimension and data-matrix rank

	MNET	Resnet	Densenet
Dimension	1152	16384	1368
Rank	499	9365	206

Table 5: AUPR (%) scores from three different density functions: GMM, Sep (Gaussian with separate covariance per class), Tied (Gaussian with tied covariance). Best values are shown in **bold**.

MNIST	FashionMNIST			EMNIST			FGSM, $\epsilon = 0.2$		
	GMM	Sep	Tied	GMM	Sep	Tied	GMM	Sep	Tied
Layer 0	**86.4**	84.8	81.8	**66.5**	61.1	53.4	**96.0**	95.3	94.8
Layer 1	**84.5**	81.3	53.5	67.5	**72.8**	65.4	**96.4**	95.3	88.0
Layer 2	88.4	**90.9**	58.1	72.0	**77.7**	58.1	**97.7**	97.7	86.1
CIFAR10 (Resnet)	SVHN			LSUN			FGSM, $\epsilon = 0.1$		
	GMM	Sep	Tied	GMM	Sep	Tied	GMM	Sep	Tied
Layer 0	**84.3**	82.9	80.4	**96.3**	95.9	95.5	94.8	94.7	**94.9**
Layer 1	**85.9**	83.5	77.8	**96.3**	95.7	95.0	95.0	**95.1**	95.1
Layer 2	**62.7**	62.2	61.6	87.9	**88.0**	79.4	**92.6**	92.5	90.9
CIFAR10 (Densenet)	SVHN			LSUN			FGSM, $\epsilon = 0.1$		
	GMM	Sep	Tied	GMM	Sep	Tied	GMM	Sep	Tied
Layer 0	79.1	77.2	**79.6**	**29.5**	28.9	21.9	85.8	**86.1**	84.6
Layer 1	**86.8**	85.6	75.9	77.7	**80.4**	79.3	**90.6**	90.2	87.0
Layer 2	**80.4**	80.4	57.1	**37.2**	37.2	27.1	87.2	**87.2**	78.1