Mahalanobis distance based classifier for Out-of-Distribution samples detection

Tarek Ayed
Student
Royal Institute of Technology
KTH

Mohammed El Mendili
Student
Royal Institute of Technology
KTH

Abstract

In this study, we reproduce the main results of the paper **A simple unified framework for detecting out-of-distribution samples and adversarial attacks** [1] for Out-of-distribution detection in Neural Networks. Precisely, we fit a Class-conditional Gaussian distribution on the feature space of a pre-trained Neural Network and then define a Mahalanobis-based distance as a confidence score. We show that the original paper's main results reproduce fairly well. We also show that this method can also achieve high performance when applied to text data. Furthermore, we demonstrate how the base model's performance affects OOD detection. Our code is available on `https://github.com/MohammedElm/OOD-Detection-using-Mahalanobis-distance`

1 Introduction

Recent advances in Deep Learning have led to impressive performance in tasks such as object detection, speech recognition, image classification, etc. However, deploying these models in real world applications requires far more than a good test performance. Questions such as explainability, interpretability or uncertainty quickly become crucial when designing an automated decision model. In this study, we focus on the **uncertainty** topic in Neural Networks (NN). In fact, Deep Learning models often achieve very high performance when test data is similar to the training data. However, these networks tend to be overconfident on anomalous test samples [2]. Ideally, we want to be able to respond to the question **"How much is the model confident about its predictions?"**, and many works in literature have tried to solve this problem using various techniques. In Odin [3], Liang et al. have suggested the maximum of the posterior softmax distribution as a confidence score when input pre-processing and temperature scaling are performed. In [4], Vias et al. suggested an ensembling of multiple classifiers trained on different parts of the dataset with an entropy margin modification of the loss function.

In this project, we reimplemented and reproduced from scratch the results of the paper **"A simple unified framework for detecting out-of-distribution samples and adversarial attacks"** (Lee et al. [1]). Using the feature space generated by a pre-trained NN, the idea is to construct a Mahalanobis-distance based score to detect OOD and Adversarial samples. Unlike other approaches [3], the score proposed in [1] doesn't use the last softmax classifier results which is known to provide over-confident predictions when evaluated on abnormal samples [1].

Contributions: We reproduced the main results presented in [1]. Thus, we built from scratch the Mahalanobis-based score with Input pre-processing and feature ensembling as suggested in [1]. Also, we **re-trained** from scratch a ResNet34 neural network [3] on CIFAR-10 [5], SVNH [6] and CIFAR-100 [5] datasets and evaluated the performance of the Mahalanobis-based score on different

In-Distribution (ID) / Out-of-Distribution (OOD) setups. We achieve very comparable results to [1] (see table 1). We have also conducted an ablation study to understand the correlation between *validation accuracy of the NN* and the performance of the mahalanobis score OOD detector (figure 1). We found that the more well-trained a NN, the best is the performance of the score. We studied the contribution of each layer of the ResNet to the overall performance and concluded that in the case of ResNet, later layers usually yield better performance. Finally, we **extended** this score beyond Image classification tasks by applying it to text data using a pre-trained BERT$_{\text{BASE}}$ [7] model. In these experiments, we achieved good baseline performances. These experiments could be the ground for a more text-adapted confidence score. Future work could also explore how gradient-based input preprocessing could be implemented in a language model, such as BERT.

Mathematical Notation

Across this document, we denote $\mathbf{D}_{\text{in}} = (\mathbf{x_i}, \mathbf{y_i})_{i\in\{1,..,n\}}$ the training dataset (and the IN-Distribution dataset), $\mathbf{D}_{\text{out}} = (\hat{\mathbf{x}}_{\mathbf{i}})$ (possibly unlabelled) the OOD dataset, $\mathbf{w}$ the NN parameters and $f_{\mathbf{j},\mathbf{w}}(\mathbf{x_i})$ the output of the j-th layer of the NN (with L layers) for a given input x_i. We only considers classification problems and we denote C the number of classes.

2 Mahalanobis-based score

In [1], authors used the following simple, yet fundamental equivalence between LDA (Linear discriminant analysis) and the softmax classifier [1]:

$$\exists \Gamma, \forall k, \exists \mu_k, x|y = k \sim \mathcal{N}(\mu_k, \Gamma) \Leftrightarrow \forall k, \exists \beta_k, b_k, \mathbf{P}(y = k|x) = \frac{\exp(\beta_k x + b_k)}{\sum_{c\in C} \exp(\beta_c x + b_c)}$$

This equivalence links the softmax classifier (discriminative) and the class-conditional gaussian distribution (generative). Hence, fitting the latter on the feature space generated by the last layer of a NN is expected to yield comparative performance to the softmax classifier applied to this space (in [1], authors validated this conclusion empirically). Given a **well-trained** NN, the idea here is to use the features given by the NN's layers to assume a class-conditional multivariate Gaussian distribution (similar to Latent Discriminant analysis). More formally, let's denote $f_{\mathbf{i},\mathbf{w}}(x)$ the projection of x into the space generated by the i-th layer of the NN (**already trained** with w as a parameter vector) and y the corresponding label of x. We assume the following:

$$\forall k \in \{1, .., C\}, i \in \{1, .., L\}, \;\; (f_{\mathbf{i},\mathbf{w}}(X)|Y = k) \sim \mathcal{N}(\mu_{k,i}, \Sigma_i)$$

where

$$\mu_{k,i} = \frac{1}{N_k} \sum_{j\in\{1,..,n\}} \mathbf{1}(y_j = k) f_{\mathbf{i},\mathbf{w}}(x_j),$$

$$\Sigma_i = \frac{1}{n} \sum_{c\in\{1,..,C\}} \sum_{j\in\{1,..,n\}, y_j = c} (f_{\mathbf{i},\mathbf{w}}(x_j) - \mu_{c,i})^\top (f_{\mathbf{i},\mathbf{w}}(x_j) - \mu_{c,i})$$

and $N_k = \sum_{j\in\{1,..,n\}} \mathbf{1}(y_j = k)$.

Given these assumptions, we define, for each layer of the NN, the **Mahalanobis distance-based confidence score** as follows:

$$\forall i \in L, M_i(x) = \max_c - (f_{\mathbf{i},\mathbf{w}}(x) - \mu_c)^\top \Sigma_i^{-1} (f_{\mathbf{i},\mathbf{w}}(x) - \mu_c)$$

which corresponds to the log of the probability density (given training data) of the test sample considered.

After computing the scores for each layer, a logistic regression can be performed on these scores (as features) to detect whether a sample is OOD or not. This defines the final score as follows:

$$M(x) = \sum_i a_i M_i(x)$$

where a_i are the learnt parameters of the logistic regression model. Then, a threshold-based classifier is defined to separate in- and out-distributions.

Input pre-processing: Following [3], authors in [[1] also suggest adding a small noise ϵ to inputs. However, the noise is added in order to **increase the mahalanobis score** of the closest class gaussian distribution:

$$\hat{x} = x + \epsilon.\text{sign}(\nabla_{\mathbf{x}} M(x))$$

The idea is that such a perturbation will have a bigger impact on ID samples than on OOD ones, thus increasing detection performance. We verified in our experiments that adding such a noise makes the In-Distribution and the Out-Distribution samples more separable by the Mahalanobis score.

3 Experiments

Image data experiments

We started by training, **from scratch**, a ResNet-34 [3] model on three datasets: CIFAR-10, CIFAR-100 and SVHN. All three datasets are constituted of color images of size 32×32. We used exactly the same ResNet architecture as the paper [1], which is the one outlined in [3] for CIFAR-10 and CIFAR-100 datasets. Same training hyperparameters were also used: 200 epochs, SGD with 0.9 momentum and and a 0.1 learning for half the training, then 0.01 until 75% completed, and finally 0.001 until the end of the training. Resulting models yielded 91.8%, 67.2% and 96.0% validation accuracies on CIFAR-10, CIFAR-100 and SVHN respectively, which we found satisfactory for our experiments' purposes.

The first experiment we conducted was using last-layer features in the Mahalanobis detector. Like in [1], we average our ResNet's features along height and width, keeping a feature vector the size of the number of channels at the considered layer. This experiment was conducted with CIFAR-10 as ID and SVHN as OOD. We also measured performance with and without input pre-processing. It is worth noting that the only version of the Mahalanobis detector that does not require any supervision (training, fine-tuning) on OOD examples is the single-layer detector without input pre-processing.

Secondly, we measured the single-layer performances yielded by each of the ResNet's layers. These results are obtained while using input pre-processing, and also with CIFAR-10 as ID and SVHN as OOD.

Finally, we conducted the main experiment of [1]: multi-layer feature ensembling through a Logistic Regression. Weights of the logistic regression are trained with data from the considered pair of ID and OOD datasets. We sample 10000 examples from the ID/OOD testsets and train our logistic regression on half, and evaluate the performance on the other half. No hyperparameter tuning was done on the logistic regression. The value of the magnitude ϵ used is the one found to be best in the corresponding single-layer experiment. This was conducted on three pairs of ID/OOD datasets: CIFAR-10/SVHN, CIFAR-100/SVHN and SVHN/CIFAR-10. Results are compiled in table 1.

Hyper-parameters fine-tuning: The Mahalanobis detector has 2 main hyper-parameters: The amplitude ϵ of the added noise for the input pre-processing and which layer to consider when in single-layer mode. In order to fine-tune these parameters, we performed a grid search on a small portion of the validation set (1000 pairs from ID/OOD datasets). For the noise, we followed [1] and chose it from [0, 0.0005, 0.001, 0.0014, 0.002, 0.0024, 0.005, 0.01, 0.05, 0.1, 0.2].

Performance metrics: Following [1], we use the following threshold-free evaluation metrics for our detector:

- **AUROC:** Denotes the Area under the receiver operating characteristic curve. It is the plot of True Positive Tate against The False Positive Rate when a classifier's threshold is varied.
- **AUPR in(out)**: Denotes the Area under the precision-recall curve. *AUPR In (Out)* is when In(Out)-distribution samples are considered positives.
- **Accuracy:** Denotes the maximum possible (over all thresholds) probability of correct classification. We assume in our computations that $\mathbf{D}_{\text{in}}$ and $\mathbf{D}_{\text{out}}$ are equally probable: $\mathbf{acc} = 1 - \min_{\delta} \frac{1}{2}\Big(\mathbb{P}(M(x) \leq \delta | x \in \mathbf{D}_{\text{in}}) + \mathbb{P}(M(x) > \delta | x \in \mathbf{D}_{\text{out}})\Big)$
- **TNR AT 95% TPR:** Denotes the True negative rate (TNR) when true positive rate (TPR) is equal to 95%.

ID	OOD	ROCAUC	AUPR in	AUPR out	Accuracy	TNR at 95% TPR
CIFAR-10	SVHN	98.9	98.3	99.3	97.6	97.0
CIFAR-100	SVHN	99.4	98.9	99.5	97.8	98.0
SVHN	CIFAR-10	97.7	96.1	98.2	94.2	93.0

Table 1: Out-of-distribution detection performance using Mahalanobis distance, along with input pre-processing ($\epsilon = 0.005$) and feature ensembling with a logistic regression.

Text data experiments

In addition to these experiments on image data, we explored the use of the method presented in [1] for text data. In fact, this method relies heavily on high-level features computed by a pre-trained model. However, language models are essentially general-purpose feature computing models, which makes them good candidates for a Malanobis-distance based OOD detector.

We chose to conduct our experiments on the following pair of datasets: IMDB [8] as ID and Yelp Polarity [9] as OOD. The reason for this choice is that both are review datasets, and both are binary sentiment classification datasets, which makes them highly similar. Measured performance on this pair should thus be a good indicator of the Mahalanobis method for OOD detection in text data. As a base model, we chose the transformer language model BERT [7] in its smaller version BERT$_{\text{BASE}}$, which has been shown to provide meaningful features in a classification context [10].

In order to use the Mahalanobis method with BERT$_{\text{BASE}}$, a few minor adaptations were made. First, BERT is a tranformer language model, which means it operated at the level of words. However, as detailed in [7], a special token - [CLS] - is used internally in the model to compute a sequence feature vector. This vector is used when fine-tuning BERT for classification tasks and is present at each layer. In BERT$_{\text{BASE}}$, it has a 768 size. Therefore, we don't do any averaging on these feature vectors, as opposed to ResNet models. Another significant difference is that text inputs are seen as discrete integers and not a continuous variable. It is therefore not possible to differentiate the Mahalanobis score relative to the inputs, as outlined in the input preprocessing step of [1]. As a consequence, no input preprocessing is applied in these experiments.

Several experiments were conducted in this setup with a pre-trained BERT$_{\text{BASE}}$ model. First of all, we replicated the methods used with image data: last-layer based classifier and feature ensembling across all layers with a logistic regression. However, we observed that in this setup, the last layer is not the best performing one, as it usually was the case with image data. Therefore, we also reported the best-layer based detector's performance. Additionally, as the BERT model we initially used is not trained specifically on the datasets we use in our OOD detection task, we conducted these experiments again after fine-tuning the pre-trained BERT model on the downstream IMDB classification task, in order to see if dataset specificity helps or hurts performance. Our text data results are summarized in table 2.

Model used	Method	AUROC	AUPR in	AUPR out	Accuracy	TNR at 95% TPR
BERT$_{\text{BASE}}$ [7]	Last layer	82.9	84.6	80.3	76.1	29.6
	Best layer	**96.5**	**97.2**	**95.6**	**91.9**	**77.9**
	All layers	90.1	87.5	89.9	83.6	57.7
BERT$_{\text{BASE}}$ [7]	Last layer	73.0	74.3	68.7	71.4	10.9
fine-tuned with	Best layer	94.3	95.1	92.9	88.1	66.5
IMDB	All layers	82.6	81.2	81.1	76.3	35.1

Table 2: Out-of-distribution detection performance using Mahalanobis distance. IMDB [8] is the ID dataset and Yelp Polarity [9] is the OOD. "All layers" results correspond to the use of a logistic regression.

Ablation study and analysis of the method

Influence of the underlying NN used for feature generation As the results from our text data experiments (see table 2) tend to show, changes applied to the NN used in our Mahalanobis detector could translate into changed performance of the OOD detector. In fact after fine-tuning BERT$_{\text{BASE}}$, we saw an overall decrease in the OOD detector's performance.

In order to further explore the influence of the underlying model's performance on the performance of the OOD detector, we performed the following ablation study. During our training of ResNet-34 [3] on CIFAR-10, we saved snapshots of the model at different validation accuracies. We then compared the performance yielded by our Mahalanobis detector when using each of these snapshots. The results of this experiment are shown in figure 1. They suggest that better performance of the base model brings better performance of the OOD classifier. This is intuitively understandable as, when trained, a deep model such as ResNet-34 keeps improving the quality of its internal generated features. However, this finding is the opposite of what we observed with BERT (table 2) where fine-tuning, which improved classification accuracy from 50% to 80%, hurts the OOD detector's performance. This could be explained by the fact that the purpose of the non-fine-tuned version of BERT is general feature generation, and altering these features for better classification accuracy seems to decrease their ability to distinguish OOD samples from ID ones.

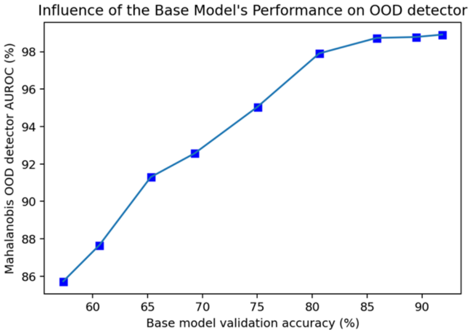

Figure 1: AUROC of the OOD detector (with input preprocessing at $\epsilon = 0.005$ and feature ensembling with a logistic regression) at different validation accuracies of our ResNet-34 model.

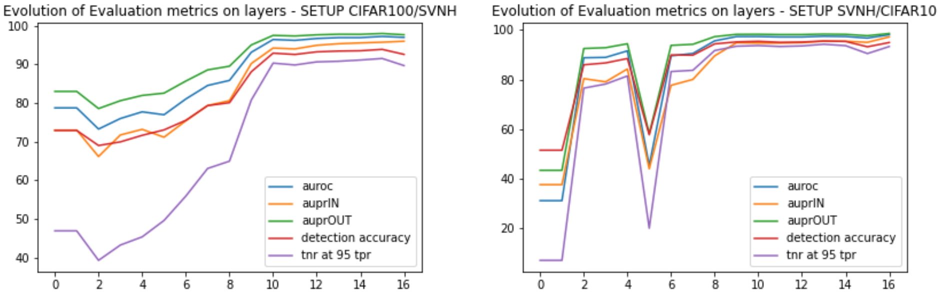

Figure 2: Performance measures of the OOD detector (with input preprocessing at $\epsilon = 0.005$) at different single layers of our ResNet-34 model.

Contribution of individual layers in OOD detection We have conducted several measurements/visualizations in order to understand the contribution of each individual layer of the models used in OOD detection. First, we measured individual layers performance when used in a single layer Mahalanobis detector (see figure 2). Similarly, we used the coefficients assigned by the logistic regression used in multi-layer detectors to each layer (after layer-normalizing) to compare each layer's importance (see figure 3)

It seems that for all ResNet-based detectors, later layers are most important and most powerful in OOD detection, which could signal better representations of the input space. It is also clear that in the SVHN/CIFAR-10 setup, the detector heavily relies on the last two layers (figure 3), which is probably part of why we have gotten worse performance in this setup compared to the two others (see our results in table 1).

Notably however, we had the opposite finding in our text data experiments. In these, the performance peaks at layers 2-3 and then decreases as we go deeper in the network. This could be explained by the fact that later layers in BERT are geared towards the auto-regressive pre-training objective, which is highly distinct from a classification task.

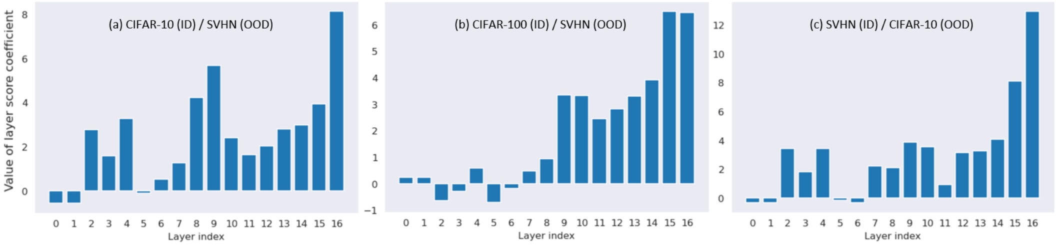

Figure 3: Relative importance of each individual layer in the multi-layer logistic regression.

4 Discussion

Similarity to the original paper's results The first goal of our work is to reproduce results from Lee et al. (2018) [1]. Our main results, shown in table 1 are similar overall to the corresponding results from [1]. A synthesis of the differences observed between the results of our experiments and the ones reported in [1] is given in table 3. When using SVHN as the OOD dataset, we achieve better results than [1]. However, when SVHN is ID, we significantly underperform the results from [1]. Overall, it seems that the paper's method reproduces relatively well. The differences when SVHN is ID can be attributed to the OOD detector's reliance on the underlying model. As shown in table 2, changing the base model often changes the OOD detector's performance, and a better classifier is not always a better feature generator, as shown by our BERT experiments.

ID	OOD	AUROC	Accuracy	TNR at 95% TPR	Average change
CIFAR-10	SVHN	-0.2	$+1.8$	$+0.6$	$+0.7$
CIFAR-100	SVHN	$+1.0$	$+4.1$	$+6.1$	$+3.7$
SVHN	CIFAR-10	-1.6	-2.7	-5.4	-3.2
	Average change	-0.3	$+1.1$	$+0.4$	$\mathbf{+0.4}$

Table 3: Difference between our results (see table 1) and the results from [1]

Generality of the Mahalanobis detector We gave also ourselves the objective of replicating the paper's experiment in the context of text data, in order to assess whether the Mahalanobis score method is generalizable. We successfully replicated the experiment, as shown in table 2. Although we found no baseline to compare these results to, we found them highly promising and similar to image data performance (see table 1).

OOD detection on text data can be very useful, for example when used on a ChatBot or virtual assistant, or on a text classifier like email filters. Being powerful on two widely different types of data (image and text), it is safe to say that the Mahalanobis score method from [1] is both a general and powerful method of OOD detection. Notably, text data results were obtained without any sort of input pre-processing, which makes inference a lot faster.

Important variables Auxiliary experiments, like our ablation study on the base model's performance and our analysis of single layer performance, allow to have a better understanding of the different factors that can affect the Mahalanobis detector's performance. One overarching finding is that this performance is highly dependent on the power and *relevance* of the underlying model's features. In the case of a ResNet classifier, this translates into better performance with later layers (see figure 2), and better performance with higher validation accuracy of the base model (see figure 1). It is also visible in the performance discrepancy between pre-trained BERT and its fine-tuned version (see table 2).

The importance of supervision Fundamentally, there are two types of OOD detectors: those that need exposure to OOD examples and those that do not. In the paper under study [1] and in our experiments, there are both types. In real-world applications, there is often not any OOD examples at hand. It is thus important to draw the line between the two types and assess the importance of supervision.

In the Mahalanobis method, supervision is needed for hyperparameter fine-tuning: noise amplitude when input pre-processing and layer selection for single-layer detectors, and to train the logistic regression when feature ensembling. Technically, this means that the only non-supervised detectors in our experiments are the last-layer BERT based models, the best of which has an AUROC of 82.9% and an accuracy of 76.1% (see table 2). However, the choice of which layer to use for a single-layer detector can be done without exposure to OOD examples by identifying a general rule of thumb of which layers perform best in general (the last few in the case of ResNet and layers 2-3 in the case of BERT). Having added this caveat, we can reasonably add the best-layer based detector to the non-supervised category, which achieves 96.5% AUROC and 91.9% accuracy. In the case of ResNet models, it could be argued that the value of ϵ can be fixed in advance without fine-tuning ($\epsilon = 0.005$ for example like in our multi-layer experiments), which makes the single-layer ResNet detectors also non-supervised (see figure 2). Therefore, it seems clear that the Mahalanobis score method detection does not suffer much from the lack of exposure to OOD samples and can yield high performance either way.

Next steps We have identified two investigation areas to further improve the Mahalanobis detector's performance. First, having observed that the pre-trained version of BERT outperforms the fine-tuned one, a natural next step would be to try this approach on image data, by using a general pre-trained ResNet instead of a classifier trained on the ID dataset.

Another unexplored avenue is the implementation of some sort of input pre-processing in the case of text data. This could be done, for example, by differentiating the Mahalanobis score relative to the embedding layer, rather than the input itself (which is not possible because of its discrete nature), and then perturb the input in the embedding space directly. Although conceptually simple, this idea is particularly difficult to implement on a pre-trained BERT, as one would need to customize the implementation of the BERT model itself, while keeping it compatible with the pre-trained version's code. Otherwise, one would need to retrain a BERT model from scratch.

References

[1] Kimin Lee et al. *A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks*. 2018. arXiv: 1807.03888 [stat.ML].

[2] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. *Deep Anomaly Detection with Outlier Exposure*. 2019. arXiv: 1812.04606 [cs.LG].

[3] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[4] Apoorv Vyas et al. "Out-of-Distribution Detection Using an Ensemble of Self Supervised Leave-out Classifiers". In: (Sept. 2018).

[5] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. "CIFAR (Canadian Institute for Advanced Research)". In: (). URL: http://www.cs.toronto.edu/~kriz/cifar.html.

[6] Yuval Netzer et al. "Reading Digits in Natural Images with Unsupervised Feature Learning". In: *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*. 2011. URL: http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

[7] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[8] Andrew L. Maas et al. "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, 2011, pp. 142–150. URL: http://www.aclweb.org/anthology/P11-1015.

[9] Xiang Zhang, Junbo Zhao, and Yann LeCun. "Character-Level Convolutional Networks for Text Classification". In: *arXiv:1509.01626 [cs]* (Sept. 2015). arXiv: 1509.01626 [cs].

[10] Chi Sun et al. *How to Fine-Tune BERT for Text Classification?* 2020. arXiv: 1905.05583 [cs.CL].